Confident Learning for Machines and Humans

by

Curtis George Northcutt

B.S., Vanderbilt University (2013)

S.M., Massachusetts Institute of Technology (2017)

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Author .....

Department of Electrical Engineering and Computer Science

May 20, 2021

Certified by .....

Isaac L. Chuang

Professor of Physics and Professor of Electrical Engineering and Computer Science, Senior Associate Dean of Digital Learning

Thesis Supervisor

Accepted by .....

Leslie A. Kolodziejski

Professor of Electrical Engineering and Computer Science

Chair, Department Committee on Graduate Students

Confident Learning for Machines and Humans

by
Curtis George Northcutt

Submitted to the Department of Electrical Engineering and Computer Science on May 20, 2021, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

The coupling of *machine* intelligence and *human* intelligence has the potential to empower humans with augmented capabilities (e.g., improving rhyme-density while writing song lyrics, enhancing empathy via emotion detection, and personalizing learning in online courses). Unfortunately, humans operate in an uncertain world – where the performance of even the most sophisticated *model-centric* artificially intelligent system often depends on its *data-centric* ability to deal with the uncertainty in the labels upon which it is trained.

To this end, we introduce *confident learning* whereby a machine (like humans) must learn with *noisy-labeled data*, directly quantify and identify label noise, and *unlearn* misconceptions by re-learning with confidence on cleaned data with erroneous labels removed. We achieve this by developing a principled theory and framework for confident learning with affordances for quantifying, identifying, and learning with label errors in data, and we open-source their implementations in the `cleanlab` Python package. Based on human verification of the label errors found using cleanlab: we estimate a 3.4% lower bound error rate of the *test set* labels of ten of the most commonly used machine learning datasets across audio, image, and text modalities; examine the noise prevalence needed to change machine benchmark rankings; and provide corrected test sets so that humans can benchmark machine performance with increased confidence.

We then build and evaluate three artificially intelligent systems that augment human capabilities in noisy, real-world settings. Namely: (1) assisted-turn-taking in multi-person conversations by combining noisy embodied audio and video signals from multiple synchronized perspectives, (2) assisted-generation of writing song lyrics by exploiting the inherent aleatoric uncertainty of language and semantics, and (3) assisted-human-learning in open online courses by depolarizing/diversifying comment rankings to mitigate the majority bias inherent in rankings based on upvotes. In each case, the artificially intelligent system's ability to overcome uncertainty is linked to its efficacy of augmenting human capabilities, and by extension, humans' confidence in their ability to perform the associated task.

Thesis Supervisor: Isaac L. Chuang
Title: Professor of Physics and Professor of Electrical Engineering and Computer Science, Senior Associate Dean of Digital Learning

Acknowledgments

To my family The greatest gift my family afforded me is the freedom to pursue my own interests. My family never suggested I pursue a PhD degree or my life manifesto. Instead, they afforded me the freedom to want them for myself. For this I am eternally grateful. My family and especially my father always encouraged me to do what interested me. There was never any expectation of me beyond that I would always do my best at anything I attempted.

I am grateful to my father for being so reliable throughout my life. I can count on one hand the number of times in the last three decades that I called my father and he did not answer the phone (or text me shortly after). Although my father's job was technically a mailman, I know that the *real* job he prioritized was to be a good father, and to that end, he succeeded immensely and deserves every award there is.

I have never taken a breath without the assurance of a mother who I know will always love me in every way she can. I have two incredible sisters, both who have provided for me and in many ways parented me. My eldest sister, Grace, served as a surrogate mother to me in my late teenage years – she clothed me, fed me, and forced me to figure things out on my own. She drove me to visit universities. She made sure I saw a movie or two. I am better off in life because I have Grace as a sister. My second-eldest sister Virginia provided limitless insights into emotional intelligence throughout my life, sharing tidbits of gold, such as, "Don't take yourself too seriously," and "Nothing good gets away," or as I prefer to put it, "If it got away, it was bad." I am better off in life because I have Virginia as a sister. My brother-in-law, Chad Gilpin, gave his spare time after work to go through every word of more than half of the pages of my thesis. He found dozens of grammatical issues and provided explanations to fix them. Chad is a remarkable friend.

My father delivered mail. My mom worked in call centers, reception jobs, Walmart, etc. My sister Virginia is an art therapist. And my sister Grace is a civil engineer and manager. I am proud of the diversity of work and labor performed by my family. The obstacles we faced growing up serve as a constant inspiration for me to do more with less.

To my advisor, Ike In my early years at MIT, Ike disassembled my dream-motivated, gut-based approach to science and reassembled it with a focus on asking good questions and leading with results. His guidance has been pivotal in shaping how I think as a scientist and shaping the framing of my results to reach a broader scientific audience. For this I am forever grateful. Below I share a few quotes of the advice given to me by Ike over the years. I hope they may prove valuable to others. At first, these quotes may appear disparate, but they are linked by a unified belief in humanity – a belief that humans can discover anything if they learn to be fascinated by the world around them and ask the right questions to understand it.

> *"What you may find most valuable about coming to MIT is the way we explore and formulate research questions here. Beyond just sharing interesting questions is the joy I get from teaching my students the art of finding good questions. The best questions open doors to new fields; the best question-askers become leaders of the fields."*
>
> - Isaac Chuang (2013)

> *"Think neither of the algorithm nor the application first. The result comes first - then you can choose how you want to frame it."*

- Isaac Chuang (2014)

"Choose your metrics for accomplishments carefully. It's not research versus award. The causality is that true (and sustained) accolades follow accomplishment. Go for the real thing, and not merely transient public perception. Curtis Northcutt - The Movie will come, but first, dig in and make yourself proud of what you've done. Earn the right to know you're one of the top 10 in the world, first for yourself, then the world will follow."

- Isaac Chuang (2015)

To my committee members I thank Suvrit Sra and Roz Picard for their spot-on introductions to colleagues and for their feedback on my thesis. I also thank Suvrit for his feedback and thoughts during my defense. Some of the the ideas he presented will shape the research questions I pursue next.

I thank Roz whose pioneering work in affective computing and emotion understanding helped inspire the humanitarian focus of my thesis. Had I not had the opportunity to work with Roz, my thesis may have just been "Confident Learning for Machines." I also thank Roz for her diligence in proof-reading and giving feedback on every page of my thesis. Roz gave multiple suggestions that strengthened the thesis. The support Roz showed me in my middle years at MIT while I was still finding my footing helped make MIT feel like home: I am very grateful for this.

To my co-authors A significant portion of the knowledge obtained during my doctoral studies arose from discussions with my co-authors who contributed

significantly to various chapters of this thesis. I am grateful to Lu Jiang (Ch. 2), Isaac Chuang (Ch. 2), Anish Athalye (Ch. 3), Jonas Mueller (Ch. 3), Cindy Zha (Ch. 5), Steven Lovegrove (Ch. 5), Richard Newcombe (Ch. 5), Nikola I. Nikolov (Ch. 6), Eric Malmi (Ch. 6), Loreto Parisi (Ch. 6), Kimberly Leon (Ch. 7), and Naichun Chen (Ch. 7).

To my friends and colleagues I rely daily on three people: Jonas Mueller, Anish Athalye, and Lisa Vo.

When I have a "big idea" in machine learning, I bounce it off Jonas. He is a loyal friend and a brilliant colleague with a strong understanding of the field and what has and has not already been done. Part of the reason I pursued confident learning with such vigor is because Jonas and I worked through a formative paper by Elkan and Noto (2008) together and discussed the strengths of the paper and where improvements could be made.

Twice during my time at MIT, I had chance encounters with Anish Athalye, an exquisite thinker who would become a good friend. Among many other things, Anish is good at uncovering where science might be broken/noisy in real-world applications. He has been a great support in discussing start-up ideas, altruism, and hiking. I am grateful for the numerous dataset discoveries we made together and hope there are more discoveries to come.

I am eternally grateful to Lisa Vo for the love she has shown to me. She has taught the language of intentionality and the art of dreaming. No matter my dreams, she encourages them, shares in them, and gives feedback. Together we ask big questions – how will we build things that empower other people, how will we create a sustainable impact, how will we provide more opportunities for those with less? To have Lisa as a partner is to be empowered. Her presence is a gift that gives every day.

Finally, I thank a number of other good friends, colleagues, and mentors who shaped my thinking over the past eight years: Martin McCormick, Cody Coleman, Bill Thies, Rich Caruana, Y-Lan Boureau, Martin Segado, Rich Rines, Guang Hao, and Ted Yoder, Tailin Wu, Andrew Ho, Regina Barzilay, Laurens van der Maaten, Jessy Lin, Niranjan Subrahmanya, and Ming Sun.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

List of Figures

List of Tables

Chapter 1

My Thesis

> *A good index of the expertness of a judge, is the relationship between his level of confidence and his level of accuracy.*
>
> - Stuart Oskamp (1965)

This introductory chapter situates every result in this thesis within a central claim: that quantifying uncertainty in labeled data empowers machines and humans to learn and perform tasks with confidence in noisy, real-world environments.

Section 1.1 starts the chapter by motivating the importance and challenges of confidence for learning in noisy, real-world environments. Section 1.2 states the central claim of this thesis with evidential support from each chapter. Section 1.3 outlines the salient contributions of this thesis in the context of prior work. This section goes on to discuss common types of label noise and methods for learning with noisy labels addressed in the field.

I then prepare for the chapters ahead, outlining the framework of notation and assumptions used by confident learning in Section 1.4 and mandating the structure of Chapters 2 - 7 in Section 1.5 along with a summary of the contents of each chapter.

This chapter concludes with my manifesto to *empower people* in Section 1.6. This manifesto is the motivational glue uniting every result in this thesis.

1.1 Why Confidence?

Consider a judge, who may be a human or a machine, who passes a sentence of conviction or acquittal based on the evidence presented. The judge evaluates evidence, often biased (i.e., noisily labeled) by both the prosecution and the defense, and based on her model of law and the data presented, the judge labels a human as "guilty" or "innocent." If the judge makes a false positive mistake, an innocent person may be sentenced to prison, or in extreme cases, to death. If the judge makes a false-negative mistake, an innocent person may be the next victim of an incorrectly acquitted criminal. This example illustrates a singular point: learning and making decisions *with confidence* in noisy environments is a critical human problem.

How can we create machines that *confidently* learn and make decisions despite only having access to noisily-labeled, real-world data?

To address the bold question above, this thesis develops a systematic theory and framework for *confident learning*, whereby a machine, like humans, learns in the context of *noisy-labeled data*, directly quantifies and identifies label noise, and *unlearns* misconceptions by re-learning with confidence on cleaned data. The salient theorem of confident learning provides realistic, sufficient conditions for exactly finding label errors in datasets. The salient algorithm of confident learning identifies which examples a model is confident are labeled corrected, providing cleaned data for (data-centric) learning.

Before we consider a motivating example, the following definition is needed:

Definition 1 (Self-Confidence)**.** *A model $\boldsymbol{\theta}$'s predicted probability that a new, unseen example $\boldsymbol{x}$ with given label i is correctly labeled. Formally, self-confidence is expressed as $\hat{p}(\tilde{y}{=}i; \boldsymbol{x}{\in}\boldsymbol{X}_{\tilde{y}=i}, \boldsymbol{\theta})$, where $\boldsymbol{X}_{\tilde{y}=i}$ is the set of examples belonging to class i.*

We can interpret Definition 1 in the context of our judge example. Consider a judge who studies from a corpus of 100 prior court cases with (sometimes erroneous) verdicts. The judge can only study from the guesses of previous judges: there is no ground truth. After studying the first 99 court cases, the judge ($\boldsymbol{\theta}$) covers the verdict of the last case with her hand, and reads the evidence ($\boldsymbol{x}$) of the case. The judge predicts, "I'm 20% confident the defendant is guilty, and 80% confident the defendant is innocent." The judge removes her hand, revealing that the given label for the verdict was "guilty" ($\tilde{y}$). Then $\hat{p}(\tilde{y}{=}\text{guilty}; \boldsymbol{x}{\in}\boldsymbol{X}_{\tilde{y}=\text{guilty}}, \boldsymbol{\theta}) = 0.2$.

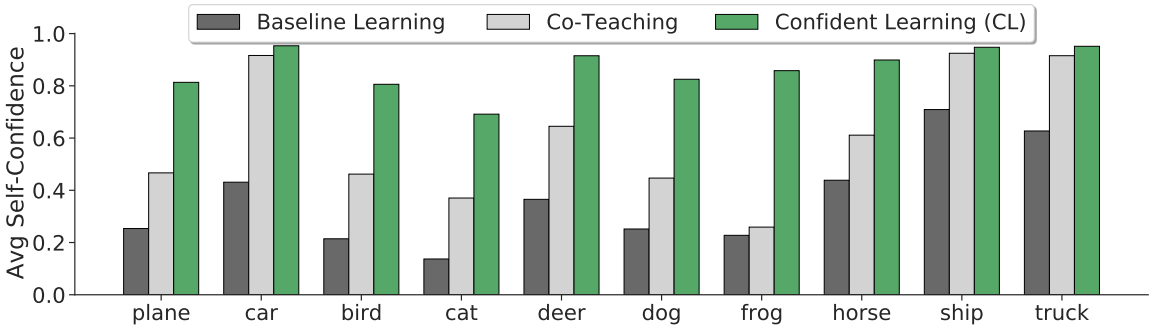


Figure 1-1: The average self-confidence on the clean (no noise added) CIFAR-10 image test set of three learning algorithms trained on the CIFAR-10 train set with 40% added label noise. For each class, the *data-centric* "Confident Learning" algorithm results in the highest average (test set) self-confidence and the "Baseline Learning" approach results in the lowest average (test set) self-confidence. "Co-Teaching" is a recent *model-centric* approach for learning with noisy labels that has been shown to be performant on CIFAR-10 (Han et al., 2018).

In Figure 1-1, I compare confident learning with two other machine learning algorithms for learning with noisy labels on the task of classifying whether an image

contains one of ten classes (e.g., plane, car, bird, etc.) on the CIFAR-10 image dataset, and examine the *average self-confidence* on unseen test data for the set of examples noisily-labeled "plane", and the set of examples noisily-labeled "car", and so forth (c.f., x-axis of Figure 1-1). Prior to training, I add 40% synthetic class-conditional label noise by randomly switching some labels of training examples to different classes non-uniformly based on a randomly generated transition matrix.

The details about each algorithm in Figure 1-1 are unimportant here (they are discussed in Chapter 2). Instead, the novelty is that Figure 1-1 shows that one can manipulate how confident a machine is in its decisions by changing how it learns with noisy data, instead of modifying/calibrating the machine's confidence directly, as is traditionally done (Guo et al., 2017).

At first glance, Figure 1-1 may seem like we are doing the right thing – the confident learning algorithm seems to make the machine more confident in its decisions, on average, for each task/class. Coming back to our judge example, this would mean the judge makes decisions for various verdicts with significantly increased self-confidence.

However, increasing self-confidence irrespective of performance is severely flawed: machines (and humans) can be *overconfident* or *underconfident* if their confidence is miscalibrated/differs from how they actually perform on a new example for the given task. Ideally, one's confidence/belief about their accuracy on a task should match one's actual accuracy when they perform the task in the future. This viewpoint is deeply rooted in our understanding of confidence from human psychology (Oskamp, 1962; Fischhoff et al., 1978; Koriat et al., 1980). In the words of Oskamp (1965):

> *"A good index of the expertness of a judge, is the relationship between his level of confidence and his level of accuracy. This measure shows, for instance, whether the judge is overconfident or underconfident in making*

his decisions. On this measure, which may be termed appropriateness of confidence, experienced judges have been found to be far superior to inexperienced ones." - Stuart Oskamp (1965)

Coming back to our judge example, the judge should be equally as self-confident in her verdict as she is accurate in her verdict, and ideally, also highly accurate. The confident learning (CL) algorithm in Figure 1-2 exhibits this behavior.

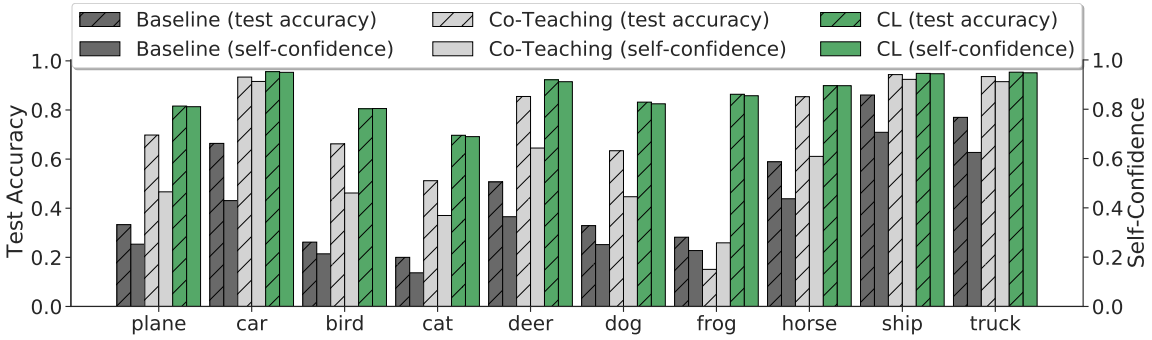


Figure 1-2: A replication of Figure 1-1 with the inclusion of the corresponding per-class test accuracy for each learning method. Confident learning (CL) has the highest test accuracy in each class with a well-calibrated average self-confidence that near-perfectly matches test performance.

Figure 1-2 includes the per-class test accuracy for each learning method with the corresponding average self-confidence. Although the co-teaching algorithm substantially improves the accuracy of the model, only the confident learning algorithm near-exactly matches self-confidence on the test set with accuracy on the test set, and is also the most accurate method for all classes. This example motivates the remaining chapters of this thesis, starting with the development of confident learning as a framework for uncertainty quantification to create machines that learn and make decisions confidently despite only having access to systematically noisily-labeled, real-world data (*Chapters 2 and 3*), followed by applications for

augmenting human capabilities in uncertain/noisy, real-world settings (*Chapters 5, 6, and 7*).

1.2 Thesis Statement

The central claim of this thesis is that quantifying uncertainty in labeled data empowers machines and humans to learn and perform tasks with confidence in noisy, real-world environments.

This claim necessitates a data-centric approach to the problem of learning with noisy labels which has traditionally been solved by introducing a new model or loss function. The reason for this paradigm shift from *model-centric* to *data-centric* learning stems from the observation that machine models produce stochastic/noisy predicted probabilities when trained in noisy, real-world environments. Model-centric methods rely on these noisy outputs to update their weights (e.g., via a weighted loss and gradient optimization step). In comparison, data-centric methods like confident learning remove or fix erroneous labels, avoiding this form of error propagation. A more extensive comparison of data-centric and model-centric methods for learning with noisy labels is discussed in the next section.

The central claim of this thesis is supported by algorithms and theorems for uncertainty quantification (*Chapter 2*) and experimental evidence for augmenting human capabilities, i.e. confidently ranking information (*Chapters 3 and 7*), conversational empathy (*Chapter 5*), and songwriting (*Chapter 6*). In Chapter 3, we overcome test set noise by *correcting the test data* to empower humans to benchmark machine models with increased confidence. In Chapter 5, we improve the performance of transcription and human turn-taking prediction applications by *combining multiple, synchronized, noisy data sources.* In Chapter 6, we support the

human creative process of song writing by modeling the task of selecting words in lyrics as a *text data denoising process*. In Chapter 7, we mitigate the majority bias inherent in online forum-based human learning by *diversifying the order of biased comment data*. Each chapter provides empirical evidence of using data-centric approaches to empower machines and humans to learn and perform tasks with confidence in noisy, real-world environments.

1.3 Confident Learning and Prior Work

This section starts with a description of confident learning and situates the subfield of confident learning within the field of machine learning. Next, I clarify how the salient contributions of confident learning in this thesis differ from prior work. Common types of label noise are then discussed, followed by a brief description of model-centric and data-centric methods for learning with noisy labels.

Confident learning (CL) is a principled framework of theory and algorithms for classification with noisy labels. CL provides affordances for (1) complete characterization of label noise in a dataset, (2) realistic sufficient conditions for exactly finding label errors in a dataset, (3) learning with noisy labels, and (4) dataset curation. Figure 1-3 provides a hierarchical contextualization of confident learning within machine learning. Notably, confident learning supports uncertainty in both the labels (aleatoric uncertainty) and the model outputs (epistemic uncertainty).

Contributions (in the context of prior work) There are are four salient contributions of confident learning (c.f., Chapters 2 and 3). First, confident learning is the first framework to estimate the joint distribution of noisy labels and true labels

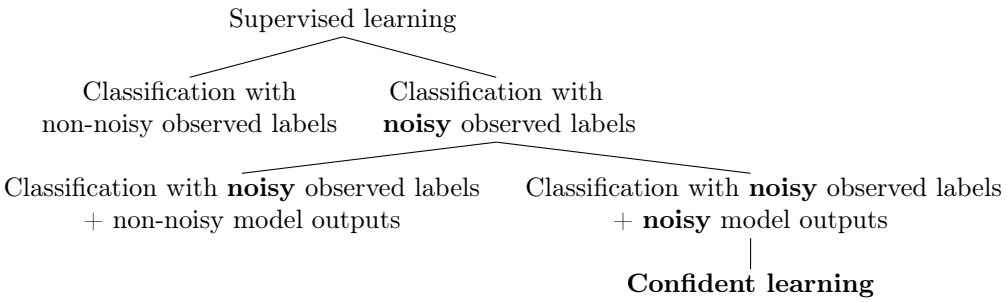


Figure 1-3: Confident learning in the context of supervised learning.

directly. Prior work focuses on estimating the conditionals (label flipping/transition rates) and marginals (prior distribution of latent, true labels) (Sukhbaatar et al., 2015; Goldberger and Ben-Reuven, 2017; Northcutt et al., 2017b; Scott, 2015), which can be directly computed from the joint distribution of noisy labels and true labels.

Second, confident learning is the first to provide sufficient conditions for exactly finding label errors with general/asymmetric class-conditional noise while allowing for stochastic/noisy model outputs for every example and class. Formative theoretical work has made significant strides in understanding learnability bounds and consistent estimation for machine learning with noisy labels (Angluin and Laird, 1988; Katz-Samuels et al., 2019; Natarajan et al., 2017, 2013; Liu and Tao, 2015; Ghosh et al., 2015), but little work has focused on the theory of the data/labels.

Third, confident learning is the first to quantify noise and find label errors at scale across ten popular machine learning test sets. Although several prior studies have considered label issues in datasets like ImageNet (Shankar et al., 2020; Beyer et al., 2020; Recht et al., 2019; Tsipras et al., 2020; Taori et al., 2020); we study the pervasive trends of label errors across many test sets, release all errors at `http://labelerrors.com`, and release tools to correct the test sets at `https://github.com/cgnorthcutt/label-errors`.

Finally, confident learning is the first work to estimate the noise prevalence needed to destabilize benchmark rankings of popular pre-trained models. Extensive prior work has verified linear trends under the distributional shift of test sets (i.e., ImageNet model rankings do not change much when a new test set is used) (Taori et al., 2020; Recht et al., 2019; Mania and Sra, 2021; Tsipras et al., 2020), but this is the first work to study the conditions (in particular, the prevalence of label noise) needed to destabilize benchmark rankings.

Types of label noise process assumptions An assumption about the noise process or the data is needed for uncertainty quantification methods to disambiguate label noise (aleatoric uncertainty) from model noise (epistemic uncertainty), otherwise the predicted probabilities of a model ($\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$) capture both uncertainties with no way to disambiguate them. This section examines three types of assumptions and explains which is used in this thesis and why. (A complete description of the notation and assumptions used in this thesis are defined in the next section (Section 1.4.) Common types of label noise studied in the literature include:

1. **uniform/symmetric class-conditional label noise**, which assumes the noise flipping rates are the same for all pairs of classes such that $p\left(\tilde{y}{=}i|y^*{=}j\right) = \epsilon, \forall i \neq j$. While a number of prior works assume this form of label noise (Goldberger and Ben-Reuven, 2017; Arazo et al., 2019; Huang et al., 2019a; Chen et al., 2019), it rarely occurs in real-world datasets (e.g., the probability a *dog* is more likely to be mislabeled as a *fox* than a *cow*).

2. **systematic/asymmetric class-conditional label noise**, which allows $p\left(\tilde{y}{=}i|y^*{=}j\right)$ to be any valid probability distribution. We assume this noise process in this thesis because it is the least assuming form of class-conditional

noise and it is common in related works (Wang et al., 2019; Natarajan et al., 2017; Lipton et al., 2018; Goldberger and Ben-Reuven, 2017; Sukhbaatar et al., 2015). We observe this form of label noise in real-world datasets (e.g., most label errors in ImageNet for the class "pig" have true label "wild boar"[1]).

3. **instance-dependent label noise**, which allows for noise of the form $p\left(\tilde{y}{=}i|y^*{=}j, \boldsymbol{x}\right)$ by making strong assumptions about the covariates of each data example $\boldsymbol{x}$ (Menon et al., 2018; Xia et al., 2020; Cheng et al., 2020; Berthon et al., 2020; Wang et al., 2021). It remains an open question whether these covariate assumptions can be modified to be broadly applicable for most real-world datasets (i.e., this form of label noise is out of scope for this thesis).

Model-centric and data-centric methods for learning with noisy labels For a complete overview of methods for learning with noisy labels, I recommend the surveys by Frénay and Verleysen (2014), Cordeiro and Carneiro (2020), and Song et al. (2021). Here, I cover a small subset of methods for learning with noisy labels and divide them into two categories: "model-centric methods" and "data-centric methods."

Model-centric methods *modify the loss* to learn with noisy labels still in the dataset. Examples of model-centric methods for learning with noisy labels include *Co-Teaching* (Han et al., 2018; Yu et al., 2019) and *MentorNet* (Jiang et al., 2018) which use the loss from one network to train another network. Another example is *SCE-loss* (Wang et al., 2019) which directly modifies the loss. A final example is *Importance Reweighting* (Liu and Tao, 2015; Patrini et al., 2017; Reed et al., 2015; Shu et al., 2019; Goldberger and Ben-Reuven, 2017) which down-weights the gradient update for presumably noisy data.

[1]An example of class-conditional label noise in a real-world dataset: `https://labelerrors.com/?dataset=ImageNet&label=pig`

Data-centric methods *modify the data*. These methods typically find label errors (directly or indirectly), then learn with noisy labels removed from the dataset by providing cleaned data for training. Data-centric methods have been shown to have robustness to stochastic/noisy model predicted probabilities (Yu et al., 2019; Pleiss et al., 2020; Li et al., 2020; Wei et al., 2020; Northcutt et al., 2017b, 2021b).

Model-centric methods often modify the loss using noisy predicted probabilities, propagating the error from these predicted probabilities into the model during optimization. Data-centric methods remove errors, avoiding the loss-modification step. Because confident learning is intended to augment human capabilities in real-world settings (Chapters 5, 6, and 7), we employ data-centric methods to learn with noisy labels robustly in real-world settings.

1.4 Framework: Notation and Assumptions

The framework described in this section applies to all chapters in this thesis, except for Chapters 4, 5, and 6 which focus on applications for augmented human capabilities and use a varied notation due to the peculiarities of each application.

In the context of multiclass data with possibly noisy labels, let $[m]$ denote $\{1, 2, ..., m\}$, the set of m unique class labels, and $\boldsymbol{X} \coloneqq (\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ denote the dataset of n examples $\boldsymbol{x} \in \mathbb{R}^d$ with associated observed noisy labels $\tilde{y} \in [m]$. $\boldsymbol{x}$ and $\tilde{y}$ are coupled in $\boldsymbol{X}$ to signify that *cleaning* removes data and label. While a number of relevant works address the setting where annotator labels are available (Sambasivan et al., 2021; Bouguelia et al., 2018; Tanno et al., 2019a,b; Khetan et al., 2018), this thesis addresses the general setting where no annotation information is available except the observed noisy labels.

Assumptions I assume there exists, for every example, a latent, true label y^*. Prior to observing $\tilde{y}$, a class-conditional classification noise process (Angluin and Laird, 1988) maps $y^* \rightarrow \tilde{y}$ such that every label in class $j \in [m]$ may be independently mislabeled as class $i \in [m]$ with probability $p(\tilde{y}{=}i|y^*{=}j)$. This assumption is reasonable and has been used in prior work (Goldberger and Ben-Reuven, 2017; Sukhbaatar et al., 2015).

Notation Notation is summarized in Table 1.1. The discrete random variable $\tilde{y}$ takes an observed, noisy label (potentially flipped to an incorrect class), and y^* takes a latent, uncorrupted label. The subset of examples in $\boldsymbol{X}$ with noisy class label i is denoted $\boldsymbol{X}_{\tilde{y}=i}$, *i.e.* $\boldsymbol{X}_{\tilde{y}=\text{cow}}$ is read, "examples with class label *cow*." The notation $p(\tilde{y}; \boldsymbol{x})$, as opposed to $p(\tilde{y}|\boldsymbol{x})$, expresses our assumption that input $\boldsymbol{x}$ is observed and error-free. I denote the discrete joint probability of the noisy and latent labels as $p(\tilde{y}, y^*)$, where conditionals $p(\tilde{y}|y^*)$ and $p(y^*|\tilde{y})$ denote probabilities of label flipping. I use $\hat{p}$ for predicted probabilities. In matrix notation, the $n \times m$ matrix of out-of-sample predicted probabilities is $\hat{\boldsymbol{P}}_{k,i} \coloneqq \hat{p}(\tilde{y} = i; \boldsymbol{x}_k, \boldsymbol{\theta})$, the prior of the latent labels is $\boldsymbol{Q}_{y^*} \coloneqq p(y^*{=}i)$; the $m \times m$ joint distribution matrix is $\boldsymbol{Q}_{\tilde{y},y^*} \coloneqq p(\tilde{y}{=}i, y^*{=}j)$; the $m \times m$ noise transition matrix (noisy channel) of flipping rates is $\boldsymbol{Q}_{\tilde{y}|y^*} \coloneqq p(\tilde{y}{=}i|y^*{=}j)$; and the $m \times m$ mixing matrix is $\boldsymbol{Q}_{y^*|\tilde{y}} \coloneqq p(y^*{=}i|\tilde{y}{=}j)$. At times, I abbreviate $\hat{p}(\tilde{y} = i; \boldsymbol{x}, \boldsymbol{\theta})$ as $\hat{p}_{\boldsymbol{x},\tilde{y}=i}$, where $\boldsymbol{\theta}$ denotes the model parameters. CL assumes no specific loss function associated with $\boldsymbol{\theta}$: the CL framework is model-agnostic.

Goal Our assumption of a class-conditional noise process implies the label noise transitions are data-independent, i.e., $p(\tilde{y}|y^*; \boldsymbol{x}) = p(\tilde{y}|y^*)$. To characterize class-conditional label uncertainty, one must estimate $p(\tilde{y}|y^*)$ and $p(y^*)$, the latent prior distribution of uncorrupted labels. Unlike prior works which estimate $p(\tilde{y}|y^*)$ and $p(y^*)$

Table 1.1: Notation used in confident learning.

Notation	Definition
m	The number of unique class labels
$[m]$	The set of m unique class labels
$\tilde{y}$	Discrete random variable $\tilde{y} \in [m]$ takes an observed, noisy label
y^*	Discrete random variable $y^* \in [m]$ takes the unknown, true, uncorrupted label
$\boldsymbol{X}$	The dataset $(\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ of n examples $\boldsymbol{x} \in \mathbb{R}^d$ with noisy labels
$\boldsymbol{x}_k$	The k^{th} training data example
$\tilde{y}_k$	The observed, noisy label corresponding to $\boldsymbol{x}_k$
y_k^*	The unknown, true label corresponding to $\boldsymbol{x}_k$
n	The cardinality of $\boldsymbol{X} \coloneqq (\boldsymbol{x}, \tilde{y})^n$, i.e. the number of examples in the dataset
$\boldsymbol{\theta}$	Model parameters
$\boldsymbol{X}_{\tilde{y}=i}$	Subset of examples in $\boldsymbol{X}$ with noisy label i, *i.e.* $\boldsymbol{X}_{\tilde{y}=\text{cat}}$ is "examples labeled cat"
$\boldsymbol{X}_{\tilde{y}=i,y^*=j}$	Subset of examples in $\boldsymbol{X}$ with noisy label i and true label j
$\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$	Estimate of subset of examples in $\boldsymbol{X}$ with noisy label i and true label j
$p(\tilde{y}=i, y^*=j)$	Discrete joint probability of noisy label i and true label j.
$p(\tilde{y}=i \vert y^*=j)$	Discrete conditional probability of true label flipping, called the noise rate
$p(y^*=j \vert \tilde{y}=i)$	Discrete conditional probability of noisy label flipping, called the inverse noise rate
$\hat{p}(\cdot)$	Estimated or predicted probability (may replace $p(\cdot)$ in any context)
$\boldsymbol{Q}_{y^*}$	The prior of the latent labels
$\hat{\boldsymbol{Q}}_{y^*}$	Estimate of the prior of the latent labels
$\boldsymbol{Q}_{\tilde{y},y^*}$	The $m \times m$ joint distribution matrix for $p(\tilde{y}, y^*)$
$\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$	Estimate of the $m \times m$ joint distribution matrix for $p(\tilde{y}, y^*)$
$\boldsymbol{Q}_{\tilde{y}\vert y^*}$	The $m \times m$ noise transition matrix (noisy channel) of flipping rates for $p(\tilde{y} \vert y^*)$
$\hat{\boldsymbol{Q}}_{\tilde{y}\vert y^*}$	Estimate of the $m \times m$ noise transition matrix of flipping rates for $p(\tilde{y} \vert y^*)$
$\boldsymbol{Q}_{y^*\vert\tilde{y}}$	The inverse noise matrix for $p(y^* \vert \tilde{y})$
$\hat{\boldsymbol{Q}}_{y^*\vert\tilde{y}}$	Estimate of the inverse noise matrix for $p(y^* \vert \tilde{y})$
$\hat{p}(\tilde{y} = i; \boldsymbol{x}, \boldsymbol{\theta})$	Predicted probability of label $\tilde{y} = i$ for example $\boldsymbol{x}$ and model parameters $\boldsymbol{\theta}$
$\hat{p}_{\boldsymbol{x},\tilde{y}=i}$	Shorthand abbreviation for predicted probability $\hat{p}(\tilde{y} = i; \boldsymbol{x}, \boldsymbol{\theta})$
$\hat{p}(\tilde{y}=i; \boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i}, \boldsymbol{\theta})$	The *self-confidence* of example $\boldsymbol{x}$ belonging to its given label $\tilde{y}=i$
$\hat{\boldsymbol{P}}_{k,i}$	$n \times m$ matrix of out-of-sample predicted probabilities $\hat{p}(\tilde{y} = i; \boldsymbol{x}_k, \boldsymbol{\theta})$
$\boldsymbol{C}_{\tilde{y},y^*}$	The *confident joint* $\boldsymbol{C}_{\tilde{y},y^*} \in \mathbb{N}_{\geq 0}^{m \times m}$, an unnormalized estimate of $\boldsymbol{Q}_{\tilde{y},y^*}$
$\boldsymbol{C}_{\text{confusion}}$	Confusion matrix of given labels $\tilde{y}_k$ and predictions $\arg\max_{i \in [m]} \hat{p}(\tilde{y}=i; \boldsymbol{x}_k, \boldsymbol{\theta})$
t_j	The expected (average) self-confidence for class j used as a threshold in $\boldsymbol{C}_{\tilde{y},y^*}$
$p^*(\tilde{y}=i \vert y^*=y_k^*)$	*Ideal* probability for some example $\boldsymbol{x}_k$, equivalent to noise rate $p^*(\tilde{y}=i \vert y^*=j)$
$p^*_{\boldsymbol{x},\tilde{y}=i}$	Shorthand abbreviation for ideal probability $p^*(\tilde{y}=i \vert y^*=y_k^*)$

independently, I estimate both jointly by directly estimating the joint distribution of label noise, $p(\tilde{y}, y^*)$. **Our goal** is to estimate every $p(\tilde{y}, y^*)$ as a matrix $\boldsymbol{Q}_{\tilde{y},y^*}$ and use $\boldsymbol{Q}_{\tilde{y},y^*}$ to find all mislabeled examples $\boldsymbol{x}$ in dataset $\boldsymbol{X}$ where $y^* \neq \tilde{y}$. This is hard because it requires disambiguation of model error (epistemic uncertainty) from

the intrinsic label noise (aleatoric uncertainty), while simultaneously estimating the joint distribution of label noise ($\boldsymbol{Q}_{\tilde{y},y^*}$) without prior knowledge of the latent noise transition matrix ($\boldsymbol{Q}_{\tilde{y}|y^*}$), the latent prior distribution of true labels ($\boldsymbol{Q}_{y^*}$), or any latent, true labels ($y*$).

Usage of the terms *Uncertainty* and *Confidence* In the context of this thesis, *uncertainty* implies uncertainty in the label space, not uncertainty in the data space (with the exception of Chapter 6), and *confidence* may imply either confidence in decision-making via out-of-sample predicted probability (formally) or confidence in learning via data-centric uncertainty quantification in the training data (informally).

Definition 2 (Sparsity)**.** *A statistic to quantify the characteristic shape of the label noise defined by fraction of zeros in the off-diagonals of* $\boldsymbol{Q}_{\tilde{y},y^*}$.

High sparsity quantifies non-uniformity of label noise, common to real-world datasets. For example, in ImageNet, *missile* may have high probability of being mislabeled as *projectile*, but near-zero probability of being mislabeled as most other classes, like *wool* or *wine*. Zero sparsity implies every noise rate in $\boldsymbol{Q}_{\tilde{y},y^*}$ is non-zero. A sparsity of 1 implies no label noise because the off-diagonals of $\boldsymbol{Q}_{\tilde{y},y^*}$, which encapsulate the class-conditional noise rates, must all be zero if sparsity $= 1$.

1.5 Thesis Organization

This thesis is organized into two parts: (1) "Confident Learning for Machines" and (2) "Confident Learning for Humans." If you are short on time, I recommend Chapter 8 which summarizes the questions addressed in this thesis in narrative form and concludes the thesis with open questions.

Each chapter in this thesis starts with an "Introduction," and ends with three sections: "Related Work," "Future Work," and "Chapter Contributions." The "Introduction" sections motivate the results of each chapter. The "Related Work" sections are useful for researchers looking to learn beyond the approaches discussed in the chapters. The "Future Work" sections are useful for researchers looking to extend the results in each chapter to new research. The "Chapter Contributions" provide a concise summary of the results in each chapter.

The contents of the remaining chapters of this thesis are as follows:

Chapter 2 sets the pace of the thesis by providing algorithms and proving theorems for exact label error finding and uncertainty quantification in realistic conditions, where error in the outputs of a machine learning model is tolerated for every training example and every class/task.

Chapter 3 applies the theory and algorithms from Chapter 2 to popular ML benchmark datasets across vision/audio/text modalities, identifying pervasive label errors in ten of the most-cited ML test sets. This chapter examines how test set errors can affect model benchmark rankings and provides key takeaways for ML practitioners who deploy machine learning models in noisy real-world settings.

Chapters 5, 6, and 7 shift the focus to augmenting human capabilities in their realistic, noisy environments. Each chapter addresses uncertainty because, unlike logic-based machine-only systems, humans operate in an uncertain world. As a simplified version of our judge example from before, to build a cheating detection system for massively open online courses (Northcutt et al., 2016) based on the principle that a person is "innocent until proven guilty", one must quantify false negatives (the uncertainty of cheating detection models) without knowing the ground truth, because, in practice, one can never know for certain that someone did not cheat in some way).

Chapter 5 combines multiple multi-modal, noisy, embodied, synchronized inputs

to augment human conversation in order to estimate turn-taking dynamics (to provide a heads-up when the person currently speaking is likely to stop). This is especially helpful for persons with autism or persons who differ greatly in background from the rest of the conversation participants – in other words, the results of this chapter serve to help such persons be more *confident* in conversations.

Chapter 6 uses denoising auto-encoders to augment human lyrical writing. Uniquely, this approach can generate lyrics for a given context (politics, romance, physics, etc.) by taking a context as input and then generating lyrics based on the keywords from that context.

Chapter 7 focuses on augmenting human learning in online forum discussions (e.g., edX online courses). The work deals with the majority bias that occurs in upvoting discussion forums (the majority will upvote more simply because they have more people to upvote), which results in effectively hiding the minority opinion. This can be detrimental to human learning if the majority opinion is incorrect and detrimental to empathy between people in different opinion spaces by reinforcing biases.

1.5.1 Publications contained in this thesis

This thesis is composed of results first derived in the following publications:

On Confident Learning for Machines: (*Chapters 2 and 3*)

- Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang (2021). Confident Learning: Estimating Uncertainty in Dataset Labels. In *Journal of Artificial Intelligence Research (JAIR)*.

- Curtis G. Northcutt, Anish Athalye, and Jonas Mueller (2021). Pervasive Label Errors in Test Sets Destabilize ML Benchmarks. In *ICLR 2021 Workshop on*

Weakly Supervised Learning.

On Confident Learning for Humans: (*Chapters 5, 6, and 7*)

- Curtis G. Northcutt, Shengxin Cindy Zha, Steven Lovegrove, and Richard Newcombe (2020). EgoCom: A Multi-person Multi-modal Egocentric Communications Dataset. In *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI).*
- Nikola I. Nikolov, Eric Malmi, Curtis G. Northcutt, and Loreto Parisi (2020). Conditional Rap Lyrics Generation with Denoising Autoencoders. In *International Conference on Natural Language Generation (INLG).*
- Curtis G. Northcutt, Kim Leon, and Naichun Chen (2017). Comment Ranking Diversification in Forum Discussions. In *Proceedings of the ACM Conference on Learning @ Scale (L@S).*

1.5.2 Additional related publications

The following publications contain early results in confident learning for (*machine*) binary classification and label error finding, and uncertainty quantification for cheating detection as a means to empower (*human*) learning in open-access online courses.

These papers were instrumental in motivating the need to address noisy labels when using machine learning to augment human capabilities:

On Confident Learning for Machines: (*not covered in this thesis*)

- Curtis G. Northcutt, Tailin Wu, and Isaac L. Chuang (2017). Learning with confident examples: Rank pruning for robust classification with noisy labels. In *Conference on Uncertainty in Artificial Intelligence (UAI).*

- Curtis G. Northcutt, Anish Athalye, and Jessy Lin (2020). Pervasive Label Errors in ML Benchmark Test Sets, Consequences, and Benefits. In *NeurIPS 2020 Workshop on Dataset Curation and Security.*

On Confident Learning for Humans: (*not covered in this thesis*)

- Curtis G. Northcutt, Andrew D. Ho, and Isaac L. Chuang (2016). Detecting and preventing "multiple account" cheating in massive open online courses. In *Computers & Education.*

- Henry Corrigan-Gibbs, Nakull Gupta, Curtis G. Northcutt, Edward Cutrell, and William Thies (2015). Deterring cheating in online environments. In *ACM Transactions on Computer-Human Interaction (TOCHI).*

1.6 Manifesto

My manifesto, motivating every result of this thesis, is to *empower humans*, to augment human intelligence with artificial intelligence, to empower people to be the best versions of themselves. For an engineer who needs to deploy the best-performing model, in Chapter 3 I apply confident learning to measure the benchmark rankings of models with confidence on cleaned test data. For a person who struggles with turn-taking dynamics in conversation (e.g., common in persons with autism), in Chapter 5 I create a new embodied conversation dataset which combines participants' noisy egocentric perspectives to provide an empathetic 5 seconds heads-up when its their turn to speak. For a musician writing song lyrics, in Chapter 6 I develop a system for automated lyric generation and rhyme enhancement for existing lyrics. And for a student taking an online course, in Chapter 7 I increase the diversity of

her learning experience by reducing the majority bias and polarization inherent in upvote-based learning forums.

Each of these examples relies on an artificially intelligent system to empower humans with augmented capabilities, but humans operate in a noisy, uncertain world – where the performance of even the most sophisticated (model-centric) artificially intelligent system often depends on its (data-centric) ability to deal with the uncertainty in the labels upon which it is trained. To this end, in Chapter 2, I develop confident learning, whereby machine models learn with confidence by providing cleaned data for training.

Confident learning unites these goals to empower both machines and humans to learn and perform tasks with confidence.

Part I

Confident Learning for Machines

Chapter 2

Confident Learning: Estimating Uncertainty in Dataset Labels

Information is the resolution of uncertainty.

- Claude Shannon (1948)

Learning exists in the context of data, yet notions of *confidence* typically focus on model predictions, not label quality. In this chapter, we introduce confident learning (CL), an alternative approach which focuses instead on label quality by characterizing and identifying label errors in datasets, based on the principles of pruning noisy data, counting with probabilistic thresholds to estimate noise, and ranking examples to train with confidence. Whereas numerous studies have developed these principles independently, in Section 2.2, we combine them, building on the assumption of a class-conditional noise process to directly estimate the joint distribution between noisy (given) labels and uncorrupted (unknown) labels. This results in a generalized CL which is provably consistent and experimentally performant. In Sections 2.3 and 2.4, we present sufficient conditions where CL exactly finds label errors, and in

Section 2.5, show CL performance exceeding seven recent competitive approaches for learning with noisy labels on the CIFAR dataset. Uniquely, the CL framework is *not* coupled to a specific data modality or model (e.g., in Section 2.5, we use CL to find several label errors in the presumed error-free MNIST dataset and improve sentiment classification on text data in Amazon Reviews). In Section 2.5, we also employ CL on ImageNet to quantify ontological class overlap (e.g., estimating 645 *missile* images are mislabeled as their parent class *projectile*), and moderately increase model accuracy (e.g., for ResNet) by cleaning data prior to training. These results are replicable using the open-source `cleanlab` release.

Attribution This chapter includes material previously published as (Northcutt et al., 2021b). Lu Jiang and Isaac Chuang contributed significantly to the material presented in this chapter.

Acknowledgements Aspects of the contents of this chapter were shaped by input from Jonas Mueller, who assisted with notation; Anish Athayle, who suggested starting the proof of claim 1 of Theorem 1 with the identity; Tailin Wu, who contributed significantly to Lemma 1; and Niranjan Subrahmanya, who provided feedback on baselines for confident learning.

2.1 Introduction

Advances in learning with noisy labels and weak supervision usually introduce a new model or loss function. Often this model-centric approach band-aids the real question: which data is mislabeled? Yet, large datasets with noisy labels have become increasingly common. Examples span prominent benchmark datasets like ImageNet

(Russakovsky et al., 2015) and MS-COCO (Lin et al., 2014) to human-centric datasets like electronic health records (Halpern et al., 2016) and educational data (Northcutt et al., 2016). The presence of noisy labels in these datasets introduces two problems. How can we identify examples with label errors and how can we learn well despite noisy labels, irrespective of the data modality or model employed? Here, we follow a data-centric approach to theoretically and experimentally investigate the premise that the key to learning with noisy labels lies in accurately and directly characterizing the uncertainty of label noise in the data.

A large body of work, which may be termed "confident learning," has arisen to address the uncertainty in dataset labels, from which two aspects stand out. First, Angluin and Laird's (1988) classification noise process (CNP) provides a starting assumption that label noise is class-conditional, depending only on the latent true class, not the data. While there are exceptions, this assumption is commonly used (Goldberger and Ben-Reuven, 2017; Sukhbaatar et al., 2015) because it is reasonable for many datasets. For example, in ImageNet, a *leopard* is more likely to be mislabeled *jaguar* than *bathtub*. Second, direct estimation of the joint distribution between noisy (given) labels and true (unknown) labels (see Fig. 2-1) can be pursued effectively based on three principled approaches used in many related studies: (a) **Prune**, to search for label errors, e.g. following the example of Chen et al. (2019); Patrini et al. (2017); Van Rooyen et al. (2015), using *soft-pruning* via loss-reweighting, to avoid the convergence pitfalls of iterative re-labeling – (b) **Count**, to train on clean data, avoiding error-propagation in learned model weights from reweighting the loss (Natarajan et al., 2017) with imperfect predicted probabilities, generalizing seminal work Forman (2005, 2008); Lipton et al. (2018) – and (c) **Rank** which examples to use during training, to allow learning with unnormalized probabilities or decision boundary distances, building on well-known robustness findings (Page et al., 1999)

and ideas of curriculum learning (Jiang et al., 2018).

No prior work has thoroughly analyzed the direct estimation of the joint distribution between noisy and uncorrupted labels. Here, we assemble these principled approaches to generalize confident learning (CL) for this purpose. Estimating the joint distribution is challenging as it requires disambiguation of epistemic uncertainty (model predicted probabilities) from aleatoric uncertainty (noisy labels) (Chowdhary and Dupuis, 2013), but useful because its marginals yield important statistics used in the literature, including latent noise transition rates (Sukhbaatar et al., 2015; Reed et al., 2015), latent prior of uncorrupted labels (Lawrence and Schölkopf, 2001; Graepel and Herbrich, 2001), and inverse noise rates (Katz-Samuels et al., 2019). While noise rates are useful for loss-reweighting (Natarajan et al., 2013), only the joint can directly estimate the number of label errors for each pair of true and noisy classes. Removal of these errors prior to training is an effective approach for learning with noisy labels (Chen et al., 2019). The joint is also useful to discover ontological issues in datasets for dataset curation, e.g. ImageNet includes two classes for the same *maillot* class (c.f. Table 2.5 in Sec. 2.5).

The generalized CL assembled in this thesis upon the principles of pruning, counting, and ranking, is a model-agnostic family of theories and algorithms for characterizing, finding, and learning with label errors. It uses predicted probabilities and noisy labels to count examples in the unnormalized *confident joint*, estimate the joint distribution, and prune noisy data, producing clean data as output.

This thesis makes two key contributions to prior work on finding, understanding, and learning with noisy labels. First, a proof is presented giving realistic sufficient conditions under which CL exactly finds label errors and exactly estimates the joint distribution of noisy and true labels. Second, experimental data are shared, showing

that this CL algorithm is empirically performant on three tasks (a) label noise estimation, (b) label error finding, and (c) learning with noisy labels, increasing ResNet accuracy on a cleaned-ImageNet and outperforming seven recent highly competitive methods for learning with noisy labels on the CIFAR dataset. The results presented are reproducible with the implementation of CL algorithms, open-sourced as the `cleanlab`[1] Python package.

These contributions are presented beginning with the formal problem specification and notation (Section 1.4), then defining the algorithmic methods employed for CL (Section 2.2) and theoretically bounding expected behavior under ideal and noisy conditions (Section 2.3). Experimental benchmarks on the CIFAR, ImageNet, WebVision, and MNIST datasets, cross-comparing CL performance with that from a wide range of highly competitive approaches, including *INCV* (Chen et al., 2019), *Mixup* (Zhang et al., 2018), *MentorNet* (Jiang et al., 2018), and *Co-Teaching* (Han et al., 2018), are then presented in Section 2.5. Related work (Section 2.6) and concluding observations (Section 2.8) wrap up the presentation. Extended proofs of the main theorems, algorithm details, and comprehensive performance comparison data are presented in the appendices.

2.2 Methods

Confident learning (CL) estimates the joint distribution between the (noisy) observed labels and the (true) latent labels. CL requires two inputs: (1) the out-of-sample predicted probabilities $\hat{\boldsymbol{P}}_{k,i}$ and (2) the vector of noisy labels $\tilde{y}_k$. The two inputs are linked via index k for all $\boldsymbol{x}_k \in \boldsymbol{X}$. None of the true labels y^* are available, except

[1]To foster future research in data cleaning and learning with noisy labels and to improve accessibility for newcomers, `cleanlab` is open-source and well-documented: https://github.com/cgnorthcutt/cleanlab/

when $\tilde{y} = y^*$, and we do not know when that is the case.

The out-of-sample predicted probabilities $\hat{\boldsymbol{P}}_{k,i}$ used as input to CL are computed beforehand (e.g. cross-validation) using a model $\boldsymbol{\theta}$, so how does $\boldsymbol{\theta}$ fit into the CL framework? Prior works typically learn with noisy labels by directly modifying the model or training loss function, restricting the class of models. Instead, CL decouples the model and data cleaning procedure by working with model outputs $\hat{\boldsymbol{P}}_{k,i}$, so that any model that produces a mapping $\boldsymbol{\theta} : \boldsymbol{x} \to \hat{p}(\tilde{y}{=}i; \boldsymbol{x}_k, \boldsymbol{\theta})$ can be used (e.g. neural nets with a softmax output, naive Bayes, logistic regression, etc.). However, $\boldsymbol{\theta}$ affects the predicted probabilities $\hat{p}(\tilde{y}{=}i; \boldsymbol{x}_k, \boldsymbol{\theta})$ which in turn affect the performance of CL. Hence, in Section 2.3, we examine sufficient conditions where CL finds label errors exactly, even when $\hat{p}(\tilde{y}{=}i; \boldsymbol{x}_k, \boldsymbol{\theta})$ is erroneous. Any model $\boldsymbol{\theta}$ may be used for final training on clean data provided by CL.

CL identifies noisy labels in existing datasets to improve learning with noisy labels. The main procedure (see Fig. 2-1) comprises three steps: (1) estimate $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ to characterize class-conditional label noise (Sec. 2.2.1), (2) filter out noisy examples (Sec. 2.2.2), and (3) train with errors removed, reweighting the examples by class weights $\frac{\hat{\boldsymbol{Q}}_{y^*}[i]}{\hat{\boldsymbol{Q}}_{\tilde{y},y^*}[i][i]}$ for each class $i \in [m]$. In this section, we define these three steps and discuss their expected outcomes.

2.2.1 Count: Quantify Uncertainty using the Confident Joint

To estimate the joint distribution of noisy labels $\tilde{y}$ and true labels, $\boldsymbol{Q}_{\tilde{y},y^*}$, we count examples that are likely to belong to another class and calibrate those counts so that they sum to the given count of noisy labels in each class, $|\boldsymbol{X}_{\tilde{y}=i}|$. Counts are captured in the *confident joint* $\boldsymbol{C}_{\tilde{y},y^*} \in {\mathbb{Z}_{\geq 0}}^{m \times m}$, a statistical data structure in CL to directly find label errors. Diagonal entries of $\boldsymbol{C}_{\tilde{y},y^*}$ count correct labels and non-diagonals

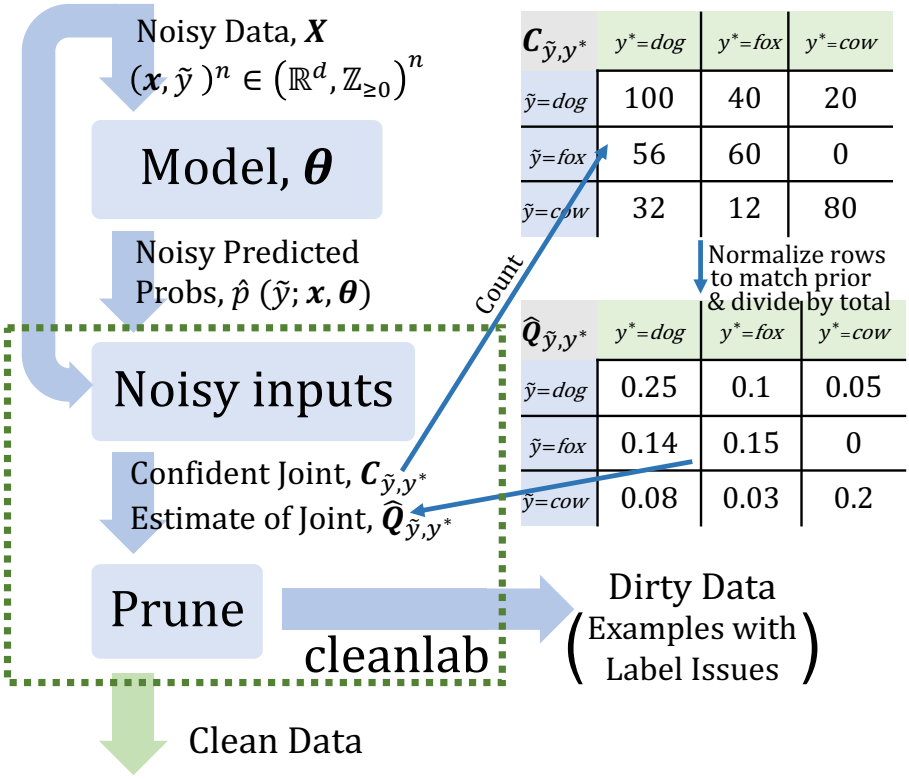


Figure 2-1: An example of the confident learning (CL) process. CL uses the confident joint, $\boldsymbol{C}_{\tilde{y},y^*}$, and $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$, an estimate of $\boldsymbol{Q}_{\tilde{y},y^*}$, the joint distribution of noisy observed labels $\tilde{y}$ and unknown true labels y^*, to find examples with label errors and produce clean data for training.

capture asymmetric label error counts. As an example, $C_{\tilde{y}=3,y^*=1}$=10 is read, "Ten examples are labeled *3* but should be labeled *1*."

In this section, we first introduce the *confident joint* $\boldsymbol{C}_{\tilde{y},y^*}$ to partition and count label errors. Second, we show how $\boldsymbol{C}_{\tilde{y},y^*}$ is used to estimate $\boldsymbol{Q}_{\tilde{y},y^*}$ and characterize label noise in a dataset $\boldsymbol{X}$. Finally, we provide a related baseline $\boldsymbol{C}_{\text{confusion}}$ and consider its assumptions and short-comings (e.g. class-imbalance) in comparison with $\boldsymbol{C}_{\tilde{y},y^*}$ and CL. CL overcomes these shortcomings using thresholding and collision handling to enable robustness to class imbalance and heterogeneity in predicted

probability distributions across classes.

The confident joint $\boldsymbol{C}_{\tilde{y},y^*}$ $\boldsymbol{C}_{\tilde{y},y^*}$ estimates $\boldsymbol{X}_{\tilde{y}=i,y^*=j}$, the set of examples with noisy label i that actually have true label j, by partitioning $\boldsymbol{X}$ into estimate bins $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$. When $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$, then $\boldsymbol{C}_{\tilde{y},y^*}$ exactly finds label errors (proof in Sec. 2.3). $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ (note the hat above $\hat{\boldsymbol{X}}$ to indicate $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ is an estimate of $\boldsymbol{X}_{\tilde{y}=i,y^*=j}$) is the set of examples $\boldsymbol{x}$ labeled $\tilde{y}{=}i$ with *large enough* $\hat{p}(\tilde{y}=j;\boldsymbol{x},\boldsymbol{\theta})$ to likely belong to class $y^*{=}j$, determined by a per-class threshold, t_j. Formally, the definition of the *confident joint* is

$$\boldsymbol{C}_{\tilde{y},y^*}[i][j] \coloneqq |\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}| \quad \text{where}$$
$$\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \coloneqq \left\{ \boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : \; \hat{p}(\tilde{y}=j;\boldsymbol{x},\boldsymbol{\theta}) \geq t_j, \;\; j = \underset{l \in [m]: \hat{p}(\tilde{y}=l;\boldsymbol{x},\boldsymbol{\theta}) \geq t_l}{\arg\max} \hat{p}(\tilde{y}=l;\boldsymbol{x},\boldsymbol{\theta}) \right\} \tag{2.1}$$

and the threshold t_j is the expected (average) self-confidence for each class

$$t_j = \frac{1}{|\boldsymbol{X}_{\tilde{y}=j}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \hat{p}(\tilde{y}=j;\boldsymbol{x},\boldsymbol{\theta}) \tag{2.2}$$

Unlike prior art, which estimates label errors under the assumption that the true labels are $\tilde{y}_k^* = \arg\max_{i \in [m]} \hat{p}(\tilde{y}{=}i;\boldsymbol{x}_k,\boldsymbol{\theta})$ (Chen et al., 2019), the thresholds in this formulation improve CL uncertainty quantification robustness to (1) heterogeneous class probability distributions and (2) class-imbalance. For example, if examples labeled i tend to have higher probabilities because the model is over-confident about class i, then t_i will be proportionally larger; if some other class j tends toward low probabilities, t_j will be smaller (in terms of stochastic model outputs/predicted

probabilities, irrespective of class priors). These thresholds allow us to guess y^* in spite of class-imbalance, unlike prior art which may guess over-confident classes for y^* because arg max is used (Guo et al., 2017). We examine "how good" the probabilities produced by model $\boldsymbol{\theta}$ need to be for this approach to work in Section 2.3.

To disentangle Eqn. 2.1, consider a simplified formulation:

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

The simplified formulation, however, introduces *label collisions* when an example $\boldsymbol{x}$ is confidently counted into more than one $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ bin. Collisions only occur along the y^* dimension of $\boldsymbol{C}_{\tilde{y},y^*}$ because $\tilde{y}$ is given. We handle collisions in the right-hand side of Eqn. 2.1 by selecting $\hat{y}^* \leftarrow \arg\max_{j\in[m]} \ \hat{p}(\tilde{y}=j;\boldsymbol{x},\boldsymbol{\theta})$ whenever $|\{k{\in}[m] : \hat{p}(\tilde{y}{=}k;\boldsymbol{x}{\in}\boldsymbol{X}_{\tilde{y}=i},\boldsymbol{\theta}) \geq t_k\}| \ > \ 1$ (collision). In practice with softmax, collisions sometimes occur for softmax outputs with higher temperature (more uniform probabilities), few collisions occur with lower temperature, and no collisions occur with a temperature of zero (one-hot prediction probabilities).

The definition of $\boldsymbol{C}_{\tilde{y},y^*}$ in Eqn. 2.1 has some nice properties in certain circumstances. First, if an example has low (near-uniform) predicted probabilities across classes, then it will not be counted for any class in $\boldsymbol{C}_{\tilde{y},y^*}$ so that $\boldsymbol{C}_{\tilde{y},y^*}$ may be robust to out of distribution examples from an alien class not in the dataset. Second, $\boldsymbol{C}_{\tilde{y},y^*}$ is intuitive: t_j embodies the intuition that examples with higher probability of belonging to class j than the expected probability of examples in class j probably belong to class j. Third, thresholding allows flexibility. For example, the 90^{th} percentile may be used in t_j instead of the mean to find errors with higher confidence. Despite the flexibility, we use the mean because we show (in Sec. 2.3) that this formulation exactly finds label errors in various settings, and we leave the

study of other formulations, like a percentile-based threshold, as future work.

Pseudo-code for the confident joint The confident joint is expressed succinctly in equation Eqn. 2.1 with the thresholds expressed in Eqn. 2.2. For clarity, we also reconstruct these equations into pseudo-code, as shown in Algorithm 1.

Algorithm 1 (**Confident Joint**) for class-conditional label noise characterization.

input $\hat{\boldsymbol{P}}$ an $n \times m$ matrix of out-of-sample predicted probabilities $\hat{\boldsymbol{P}}[i][j] \coloneqq \hat{p}(\tilde{y} = j; x, \boldsymbol{\theta})$
input $\tilde{\boldsymbol{y}} \in {\mathbb{N}_{\geq 0}}^n$, an $n \times 1$ array of noisy labels
procedure ConfidentJoint($\hat{\boldsymbol{P}}$, $\tilde{\boldsymbol{y}}$):
PART 1 (Compute thresholds)
for $j \leftarrow 1, m$ **do**
    for $i \leftarrow 1, n$ **do**
        $l \leftarrow$ new empty list []
        if $\tilde{\boldsymbol{y}}[i] = j$ **then**
            append $\hat{\boldsymbol{P}}[i][j]$ to l
    $\boldsymbol{t}[j] \leftarrow$ average(l) ▷ May use percentile instead of average for more confidence
PART 2 (Compute confident joint)
$\boldsymbol{C} \leftarrow m \times m$ matrix of zeros
for $i \leftarrow 1, n$ **do**
    $cnt \leftarrow 0$
    for $j \leftarrow 1, m$ **do**
        if $\hat{\boldsymbol{P}}[i][j] \geq \boldsymbol{t}[j]$ **then**
            $cnt \leftarrow cnt + 1$
            $y^* \leftarrow j$ ▷ guess of true label
    $\tilde{y} \leftarrow \tilde{\boldsymbol{y}}[i]$
    if $cnt > 1$ **then** ▷ if label collision
        $y^* \leftarrow \arg\max \hat{\boldsymbol{P}}[i]$
    if $cnt > 0$ **then**
        $\boldsymbol{C}[\tilde{y}][y^*] \leftarrow \boldsymbol{C}[\tilde{y}][y^*] + 1$
output $\boldsymbol{C}$, the $m \times m$ unnormalized counts matrix

Complexity Given predicted probabilities $\hat{\boldsymbol{P}}_{k,i}$ and noisy labels $\tilde{y}$, these require $\mathcal{O}(m^2 + nm)$ storage and arithmetic operations to compute $\boldsymbol{C}_{\tilde{y},y^*}$ for n training examples over m classes. As an example, given the predicted probabilities and the noisy labels, finding label errors in the ILSVRC ImageNet train set of 1.2 million RGB-color images of size 224x224 pixels using confident learning takes about 3 minutes on a 2018 i7 CPU using the `cleanlab` package.

Estimate the joint $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$. Given the confident joint $\boldsymbol{C}_{\tilde{y},y^*}$, we estimate $\boldsymbol{Q}_{\tilde{y},y^*}$ as

$$\hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j} = \frac{\frac{\boldsymbol{C}_{\tilde{y}=i,y^*=j}}{\sum_{j\in[m]} \boldsymbol{C}_{\tilde{y}=i,y^*=j}} \cdot |\boldsymbol{X}_{\tilde{y}=i}|}{\sum_{i\in[m],j\in[m]} \left(\frac{\boldsymbol{C}_{\tilde{y}=i,y^*=j}}{\sum_{j'\in[m]} \boldsymbol{C}_{\tilde{y}=i,y^*=j'}} \cdot |\boldsymbol{X}_{\tilde{y}=i}| \right)} \tag{2.3}$$

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 marginals. The denominator calibrates $\sum_{i,j} \hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j} = 1$ so that the distribution sums to 1.

Again, we reconstruct Equation 2.3 as pseudo-code, shown in Algorithm 2.

Algorithm 2 (**Joint**) calibrates the confident joint to estimate the latent joint distribution of noisy labels and true labels

input $\boldsymbol{C}_{\tilde{y},y^*}[i][j]$, $m \times m$ unnormalized counts
input $\tilde{\boldsymbol{y}}$ an $n \times 1$ array of noisy integer labels
procedure JointEstimation($\boldsymbol{C}$, $\tilde{\boldsymbol{y}}$):
$\tilde{\boldsymbol{C}}_{\tilde{y}=i,y^*=j} \leftarrow \frac{\boldsymbol{C}_{\tilde{y}=i,y^*=j}}{\sum_{j\in[m]} \boldsymbol{C}_{\tilde{y}=i,y^*=j}} \cdot |\boldsymbol{X}_{\tilde{y}=i}|$ ▷ calibrate marginals
$\hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j} \leftarrow \frac{\tilde{\boldsymbol{C}}_{\tilde{y}=i,y^*=j}}{\sum_{i\in[m],j\in[m]} \tilde{\boldsymbol{C}}_{\tilde{y}=i,y^*=j}}$ ▷ joint sums to 1
output $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ joint dist. matrix $\sim p(\tilde{y}, y^*)$

Label noise characterization Using the observed prior $\boldsymbol{Q}_{\tilde{y}=i} = |\boldsymbol{X}_i| \,/\, \sum_{i\in[m]} |\boldsymbol{X}_i|$ and marginals of $\boldsymbol{Q}_{\tilde{y},y^*}$, we estimate the latent prior as

$\hat{\boldsymbol{Q}}_{y^*=j} := \sum_i \hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j}, \forall j \in [m]$; the noise transition matrix (noisy channel) as $\hat{\boldsymbol{Q}}_{\tilde{y}=i|y^*=j} := \hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j} / \hat{\boldsymbol{Q}}_{y^*=j}, \forall i \in [m]$; and the mixing matrix (Katz-Samuels et al., 2019) as $\hat{\boldsymbol{Q}}_{y^*=j|\tilde{y}=i} := \hat{\boldsymbol{Q}}^{\top}_{\tilde{y}=j,y^*=i} / \boldsymbol{Q}_{\tilde{y}=i}, \forall i \in [m]$. As long as $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$, each of these estimators is similarly consistent (we prove this is the case under practical conditions in Sec. 2.3). Whereas prior approaches compute the noise transition matrices by directly averaging error-prone predicted probabilities (Reed et al., 2015; Goldberger and Ben-Reuven, 2017), CL is one step removed from the predicted probabilities by estimating noise rates based on counts from $\boldsymbol{C}_{\tilde{y},y^*}$. These counts are computed based on whether the predicted probability is greater than a threshold, relying only on the *relative ranking* of the predicted probability, not its exact value. This feature lends itself to the robustness of confident learning to imperfect probability estimation.

Baseline approach $\boldsymbol{C}_{\textbf{confusion}}$ To situate our understanding of $\boldsymbol{C}_{\tilde{y},y^*}$ performance in the context of prior work, we compare $\boldsymbol{C}_{\tilde{y},y^*}$ with $\boldsymbol{C}_{\text{confusion}}$, a baseline based on a single-iteration of the performant INCV method (Chen et al., 2019). $\boldsymbol{C}_{\text{confusion}}$ forms an $m \times m$ confusion matrix of counts $|\tilde{y}_k = i, y_k^* = j|$ across all examples $\boldsymbol{x}_k$, assuming that model predictions, trained from noisy labels, uncover the true labels, i.e. $\boldsymbol{C}_{\text{confusion}}$ simply assumes $y_k^* = \arg\max_{i \in [m]} \hat{p}(\tilde{y}=i; \boldsymbol{x}_k, \boldsymbol{\theta})$. This baseline approach performs reasonably empirically (Sec. 2.5) and is a consistent estimator for noiseless predicted probabilities (Thm. 1), but it fails when the distributions of probabilities are not similar for each class (Thm. 2) (e.g., class-imbalance, or when predicted probabilities are overconfident (Guo et al., 2017)).

Comparison of $\boldsymbol{C}_{\tilde{y},y^*}$ (confident joint) with $\boldsymbol{C}_{\textbf{confusion}}$ (baseline) To overcome the sensitivity of $\boldsymbol{C}_{\text{confusion}}$ to class-imbalance and distribution heterogeneity, the *confident joint*, $\boldsymbol{C}_{\tilde{y},y^*}$, uses per-class thresholding (Richard and Lippmann, 1991;

Elkan, 2001) as a form of calibration (Hendrycks and Gimpel, 2017). Moreover, we prove that unlike $\boldsymbol{C}_{\text{confusion}}$, the confident joint (Eqn. 2.1) exactly finds label errors and consistently estimates $\boldsymbol{Q}_{\tilde{y},y^*}$ in more realistic settings with noisy predicted probabilities (see Sec. 2.3, Thm. 2).

2.2.2 Rank and Prune: Data Cleaning

Following the estimation of $\boldsymbol{C}_{\tilde{y},y^*}$ and $\boldsymbol{Q}_{\tilde{y},y^*}$ (Section 2.2.1), any rank and prune approach can be used to clean data. This *modularity* property allows CL to find label errors using interpretable and explainable ranking methods, whereas prior works typically couple estimation of the noise transition matrix with training loss (Goldberger and Ben-Reuven, 2017) or couple the label confidence of each example with the training loss using loss reweighting (Natarajan et al., 2013; Jiang et al., 2018). In this chapter, we investigate and evaluate five rank and prune methods for finding label errors, grouped into two approaches. We provide a theoretical analysis for Method 2: $\boldsymbol{C}_{\tilde{y},y^*}$ in Sec. 2.3 and evaluate all methods empirically in Sec. 2.5.

Approach 1: Use off-diagonals of $\boldsymbol{C}_{\tilde{y},y^*}$ to estimate $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ We directly use the sets of examples counted in the off-diagonals of $\boldsymbol{C}_{\tilde{y},y^*}$ to estimate label errors.

CL baseline 1: $\boldsymbol{C}_{confusion}$. Estimate label errors as the Boolean vector $\tilde{y}_k \neq \arg\max_{j\in[m]} \hat{p}(\tilde{y}=j;\boldsymbol{x}_k,\boldsymbol{\theta})$, for all $\boldsymbol{x}_k \in \boldsymbol{X}$, where *true* implies label error and *false* implies clean data. This is identical to using the off-diagonals of $\boldsymbol{C}_{\text{confusion}}$ and similar to a single iteration of INCV (Chen et al., 2019).

CL method 2: $\boldsymbol{C}_{\tilde{y},y^*}$. Estimate label errors as $\{\boldsymbol{x} \in \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} : i \neq j\}$ from the off-diagonals of $\boldsymbol{C}_{\tilde{y},y^*}$.

Approach 2: Use $n \cdot \hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ to estimate $|\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}|$, prune by probability ranking These approaches calculate $n \cdot \hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ to estimate $|\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}|$, the count of label errors in each partition. They either sum over the y^* dimension of $|\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}|$ to estimate and remove the number of errors in each class (prune by class) or prune for every off-diagonal partition (*prune by noise rate*). The choice of which examples to remove is made by ranking the examples based on predicted probabilities.

CL method 3: Prune by Class (PBC). For each class $i \in [m]$, select the $n \cdot \sum_{j \in [m]: j \neq i} \left(\hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j}[i] \right)$ examples with lowest self-confidence $\hat{p}(\tilde{y} = i; \boldsymbol{x} \in \boldsymbol{X}_i)$.

CL method 4: Prune by Noise Rate (PBNR). For each off-diagonal entry in $\hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j}, i \neq j$, select the $n \cdot \hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j}$ examples $\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i}$ with max margin $\hat{p}_{\boldsymbol{x},\tilde{y}=j} - \hat{p}_{\boldsymbol{x},\tilde{y}=i}$. This margin is adapted from Wei et al.'s (2018) normalized margin.

CL method 5: C+NR. Combine the previous two methods via element-wise '*and*', i.e. set intersection. Prune an example if both methods PBC and PBNR prune that example.

Learning with Noisy Labels To train with errors removed, we account for missing data by reweighting the loss by $\frac{1}{\hat{p}(\tilde{y}=i|y^*=i)} = \frac{\hat{\boldsymbol{Q}}_{y^*}[i]}{\hat{\boldsymbol{Q}}_{\tilde{y},y^*}[i][i]}$ for each class $i \in [m]$, where dividing by $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}[i][i]$ normalizes out the count of clean training data and $\hat{\boldsymbol{Q}}_{y^*}[i]$ re-normalizes to the latent number of examples in class i. CL finds errors but does not prescribe a specific training procedure using the clean data. Theoretically, CL requires no hyper-parameters to find label errors. In practice, cross-validation might introduce a hyper-parameter: k-fold. Here, $k = 4$ is fixed in the experiments using cross-validation.

Which CL method to use? Five methods are presented to clean data. By default we use CL: $\boldsymbol{C}_{\tilde{y},y^*}$ because it matches the conditions of Thm. 2 exactly and is

experimentally performant (see Table 2.3). Once label errors are found, we observe ordering label errors 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) works well.

2.3 Theorems

In this section, we examine sufficient conditions when (1) the confident joint exactly finds label errors and (2) $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ is a consistent estimator for $\boldsymbol{Q}_{\tilde{y},y^*}$. We first analyze CL for noiseless $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$, then evaluate more realistic conditions, culminating in Theorem 2 where we prove (1) and (2) with noise in the predicted probabilities for every example. Proofs are provided in Sec. 2.4. As a notation reminder, $\hat{p}_{\boldsymbol{x},\tilde{y}=i}$ is shorthand for $\hat{p}(\tilde{y}{=}i; \boldsymbol{x}, \boldsymbol{\theta})$.

In the statement of each theorem, we use $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$, i.e. *approximately equals*, to account for precision error of using discrete count-based $\boldsymbol{C}_{\tilde{y},y^*}$ to estimate real-valued $\boldsymbol{Q}_{\tilde{y},y^*}$. For example, if a noise rate is 0.39, but the dataset has only 5 examples in that class, the nearest possible estimate by removing errors is $2/5 = 0.4 \approxeq 0.39$. So, $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ is technically a *consistent estimator* for $\boldsymbol{Q}_{\tilde{y},y^*}$ only because of discretization error, otherwise all equalities are exact. Throughout, we assume $\boldsymbol{X}$ includes at least one example from every class.

2.3.1 Noiseless Predicted Probabilities

We start with the *ideal* condition and a non-obvious lemma that yields a closed-form expression for threshold t_i when $\hat{p}_{\boldsymbol{x},\tilde{y}=i}$ is ideal. Without some condition on $\hat{p}_{\boldsymbol{x},\tilde{y}=i}$, one cannot disambiguate label noise from model noise.

Condition 1 (Ideal)**.** *The predicted probabilities* $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$ *for a model* θ *are* ideal

if $\forall \boldsymbol{x}_k \in \boldsymbol{X}_{y^*=j}, i \in [m], j \in [m]$, *we have that* $\hat{p}(\tilde{y}=i; \boldsymbol{x}_k \in \boldsymbol{X}_{y^*=j}, \boldsymbol{\theta}) = p^*(\tilde{y}=i|y^*=y_k^*) = p^*(\tilde{y}=i|y^*=j)$. The final equality follows from the class-conditional noise process assumption. The *ideal* condition implies error-free predicted probabilities: they match the noise rates corresponding to the y^* label of $\boldsymbol{x}$. We use $p^*_{\boldsymbol{x},\tilde{y}=i}$ as a shorthand.

Lemma 1 (Ideal Thresholds)**.** *For a noisy dataset* $\boldsymbol{X} \coloneqq (\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ *and model* $\boldsymbol{\theta}$*, if* $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$ *is* ideal*, then* $\forall i \in [m], t_i = \sum_{j \in [m]} p(\tilde{y} = i|y^*=j) p(y^*=j|\tilde{y} = i)$.

This form of the threshold is intuitively reasonable: the contributions to the sum when $i = j$ represent the probabilities of correct labeling, whereas when $i \neq j$, the terms give the probabilities of mislabeling $p(\tilde{y} = i|y^* = j)$, weighted by the probability $p(y^* = j|\tilde{y} = i)$ that the mislabeling is corrected. Using Lemma 1 under the ideal condition, we prove in Theorem 1 that confident learning exactly finds label errors and that $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ is a consistent estimator for $\boldsymbol{Q}_{\tilde{y},y^*}$ when each diagonal entry of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its row and column. The proof hinges on the fact that the construction of $\boldsymbol{C}_{\tilde{y},y^*}$ eliminates collisions.

Theorem 1 (Exact Label Errors)**.** *For a noisy dataset,* $\boldsymbol{X} \coloneqq (\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ *and model* $\boldsymbol{\theta}{:}\boldsymbol{x} \to \hat{p}(\tilde{y})$*, if* $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$ *is* ideal *and each diagonal entry of* $\boldsymbol{Q}_{\tilde{y}|y^*}$ *maximizes its row and column, then* $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ *and* $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$ *(consistent estimator for* $\boldsymbol{Q}_{\tilde{y},y^*}$*).*

While Theorem 1 is a reasonable sanity check, observe that $y^* \leftarrow \arg\max_j \hat{p}(\tilde{y}=i|\tilde{y}^*=i; \boldsymbol{x})$, used by $\boldsymbol{C}_{\text{confusion}}$, trivially satisfies Theorem 1 if the diagonal of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its row and column. We highlight this because $\boldsymbol{C}_{\text{confusion}}$ is the variant of CL most-related to prior work (e.g., Chen et al. (2019)). We next consider relaxed conditions *motivated by real-world settings* (e.g., Jiang et al. (2020a)) where $\boldsymbol{C}_{\tilde{y},y^*}$ exactly finds label errors ($\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$) and

consistently estimates the joint distribution of noisy and true labels ($\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$), but $\boldsymbol{C}_{\text{confusion}}$ does not.

2.3.2 Noisy Predicted Probabilities

Motivated by the importance of addressing class imbalance and heterogeneous class probability distributions, we consider linear combinations of noise per-class. Here, we index $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ by j to match the comparison $\hat{p}(\tilde{y}{=}j;\boldsymbol{x},\boldsymbol{\theta}) \geq t_j$ from the construction of $\boldsymbol{C}_{\tilde{y},y^*}$ (see Eqn. 2.1).

Condition 2 (Per-Class Diffracted)**.** *$\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ is **per-class diffracted** if there exist linear combinations of class-conditional errors in the predicted probabilities s.t. $\hat{p}_{\boldsymbol{x},\tilde{y}=j} = \epsilon_j^{(1)} p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)}$ where $\epsilon_j^{(1)}, \epsilon_j^{(2)} \in \mathbb{R}$ and ϵ_j can be any distribution.* This relaxes the *ideal* condition with noise that is relevant for neural networks, which are known to be class-conditionally overly confident (Guo et al., 2017).

Corollary 1.1 (Per-Class Robustness)**.** *For a noisy dataset, $\boldsymbol{X} \coloneqq (\boldsymbol{x},\tilde{y})^n{\in}(\mathbb{R}^d,[m])^n$ and model $\boldsymbol{\theta}{:}\boldsymbol{x}{\to}\hat{p}(\tilde{y})$, if $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ is **per-class diffracted** without label collisions and each diagonal entry of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its row, then $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ and $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$.*

Cor. 1.1 shows us that $\boldsymbol{C}_{\tilde{y},y^*}$ in confident learning (which counts $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$) is robust to any linear combination of per-class error in probabilities. This is not the case for $\boldsymbol{C}_{\text{confusion}}$ because Cor. 1.1 no longer requires that the diagonal of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximize its column as before in Theorem 1. For intuition, consider an extreme case of per-class diffraction where the probabilities of only one class are all dramatically increased. Then $\boldsymbol{C}_{\text{confusion}}$, which relies on $\tilde{y}_k^* \leftarrow \arg\max_{i\in[m]} \hat{p}(\tilde{y}{=}i|y^*{=}j;\boldsymbol{x}_k)$, will count only that one class for all y^* such that all entries in the $\boldsymbol{C}_{\text{confusion}}$ will be zero

except for one column, i.e. $\boldsymbol{C}_{\text{confusion}}$ cannot count entries in any other column, so $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \neq \boldsymbol{X}_{\tilde{y}=i,y^*=j}$. In comparison, for $\boldsymbol{C}_{\tilde{y},y^*}$, the increased probabilities of the one class would be subtracted by the class-threshold, re-normalizing the columns of the matrix, such that, $\boldsymbol{C}_{\tilde{y},y^*}$ satisfies Cor. 1.1 using thresholds for robustness to distributional shift and class-imbalance.

Cor. 1.1 only allows for m alterations in the probabilities and there are only m^2 unique probabilities under the ideal condition, whereas in real-world conditions, an error-prone model could potentially output $n \times m$ unique probabilities. Next, in Theorem 2, we examine a reasonable sufficient condition where CL is robust to erroneous probabilities for every example and class.

Condition 3 (Per-Example Diffracted)**.** *$\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ is **per-example diffracted** if $\forall j \in [m], \forall \boldsymbol{x} \in \boldsymbol{X}$, we have error as $\hat{p}_{\boldsymbol{x},\tilde{y}=j} = p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j}$ where*

$$\epsilon_{\boldsymbol{x},\tilde{y}=j} \sim \begin{cases} \mathcal{U}(\epsilon_j + t_j - p^*_{\boldsymbol{x},\tilde{y}=j}\,,\ \epsilon_j - t_j + p^*_{\boldsymbol{x},\tilde{y}=j}] & p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j \\ \mathcal{U}[\epsilon_j - t_j + p^*_{\boldsymbol{x},\tilde{y}=j}\,,\ \epsilon_j + t_j - p^*_{\boldsymbol{x},\tilde{y}=j}) & p^*_{\boldsymbol{x},\tilde{y}=j} < t_j \end{cases} \tag{2.4}$$

where $\epsilon_j = \mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}}\left[\epsilon_{\boldsymbol{x},\tilde{y}=j}\right]$ and $\mathcal{U}$ denotes a uniform distribution (we discuss a more general case in the Appendix).

Theorem 2 (Per-Example Robustness)**.** *For a noisy dataset, $\boldsymbol{X} \coloneqq (\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ and model $\boldsymbol{\theta}{:}\boldsymbol{x} \to \hat{p}(\tilde{y})$, if $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ is **per-example diffracted** without label collisions and each diagonal entry of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its row, then $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \approxeq \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ and $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$.*

In Theorem 2, we observe that if each example's predicted probability resides within the residual range of the ideal probability and the threshold, then CL exactly identifies the label errors and consistently estimates $\boldsymbol{Q}_{\tilde{y},y^*}$. Intuitively, if $\hat{p}_{\boldsymbol{x},\tilde{y}=j} \geq t_j$ whenever

$p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j$, and $\hat{p}_{\boldsymbol{x},\tilde{y}=j} < t_j$ whenever $p^*_{\boldsymbol{x},\tilde{y}=j} < t_j$, then regardless of error in $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$, CL exactly finds label errors. As an example, consider an image $\boldsymbol{x}_k$ that is mislabeled as *fox*, but is actually a *dog* where $t_{fox} = 0.6$, $p^*(\tilde{y}{=}fox; \boldsymbol{x} \in \boldsymbol{X}_{y^*=dog}, \boldsymbol{\theta}) = 0.2$, $t_{dog} = 0.8$, and $p^*(\tilde{y}{=}dog; \boldsymbol{x} \in \boldsymbol{X}_{y^*=dog}, \boldsymbol{\theta}) = 0.9$. Then as long as $-0.4 \leq \epsilon_{\boldsymbol{x},fox} < 0.4$ and $-0.1 < \epsilon_{\boldsymbol{x},dog} \leq 0.1$, CL will surmise $y^*_k = dog$, not fox, even though $\tilde{y}_k = fox$ is given. We empirically substantiate this theoretical result in Section 2.5.2.

Theorem 2 addresses the *epistemic* uncertainty of latent label noise, via the statistic, $\boldsymbol{Q}_{\tilde{y},y^*}$, while accounting for the *aleatoric* uncertainty of inherently erroneous predicted probabilities.

2.4 Proofs

In this section, we restate the main theorems for confident learning and provide their proofs.

Lemma 1 (Ideal Thresholds)**.** *For a noisy dataset* $\boldsymbol{X} \coloneqq (\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ *and model* $\boldsymbol{\theta}$*, if* $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$ *is* ideal*, then* $\forall i{\in}[m], t_i = \sum_{j\in[m]} p(\tilde{y} = i|y^*{=}j)p(y^*{=}j|\tilde{y} = i)$*.*

Proof. We use t_i to denote the thresholds used to partition $\boldsymbol{X}$ into m bins, each estimating one of $\boldsymbol{X}_{y^*}$. By definition,

$$\forall i{\in}[m], t_i = \mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}_{\tilde{y}=i}} \hat{p}(\tilde{y} = i; \boldsymbol{x}, \boldsymbol{\theta})$$

For any t_i, we show the following.

$$t_i = \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i}} \sum_{j \in [m]} \hat{p}(\tilde{y}{=}i | y^*{=}j; \boldsymbol{x}, \boldsymbol{\theta}) \hat{p}(y^*{=}j; \boldsymbol{x}, \boldsymbol{\theta}) \quad \triangleright \text{Bayes Rule}$$

$$t_i = \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i}} \sum_{j \in [m]} \hat{p}(\tilde{y}{=}i | y^*{=}j) \hat{p}(y^*{=}j; \boldsymbol{x}, \boldsymbol{\theta}) \quad \triangleright \text{Class-conditional Noise Process (CNP)}$$

$$t_i = \sum_{j \in [m]} \hat{p}(\tilde{y}{=}i | y^*{=}j) \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i}} \hat{p}(y^*{=}j; \boldsymbol{x}, \boldsymbol{\theta})$$

$$t_i = \sum_{j \in [m]} p(\tilde{y} = i | y^* = j) p(y^* = j | \tilde{y} = i) \quad \triangleright \text{Ideal Condition}$$

This form of the threshold is intuitively reasonable: the contributions to the sum when $i = j$ represents the probabilities of correct labeling, whereas when $i \neq j$, the terms give the probabilities of mislabeling $p(\tilde{y} = i | y^* = j)$, weighted by the probability $p(y^* = j | \tilde{y} = i)$ that the mislabeling is corrected. □

Theorem 1 (Exact Label Errors)**.** *For a noisy dataset,* $\boldsymbol{X} \coloneqq (\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ *and model* $\boldsymbol{\theta} : \boldsymbol{x} \to \hat{p}(\tilde{y})$*, if* $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$ *is* ideal *and each diagonal entry of* $\boldsymbol{Q}_{\tilde{y}|y^*}$ *maximizes its row and column, then* $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ *and* $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$ *(consistent estimator for* $\boldsymbol{Q}_{\tilde{y},y^*}$*).*

Proof. Alg. 1 defines the construction of the confident joint. We consider Case 1: when there are collisions (trivial by the construction of Alg. 1) and case 2: when there are no collisions (harder).

Case 1 (collisions):

When a collision occurs, by the construction of the confident joint (Eqn. 2.1), a given example $\boldsymbol{x}_k$ gets assigned bijectively into bin

$$\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y},y^*}[\tilde{y}_k][\arg\max_{i \in [m]} \hat{p}(\tilde{y} = i; \boldsymbol{x}, \boldsymbol{\theta})]$$

Because we have that $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$ is ideal, we can rewrite this as

$$\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y},y^*}[\tilde{y}_k][\arg\max_{i \in [m]} \hat{p}(\tilde{y} = i | y^* = y_k^*; \boldsymbol{x})]$$

And because by assumption each diagonal entry in $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its column, we have

$$\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y},y^*}[\tilde{y}_k][y_k^*]$$

Thus, any example $\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ having a collision will be exactly assigned to $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$.

Case 2 (no collisions):

We want to show that $\forall i \in [m], j \in [m], \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$.
We can partition $\boldsymbol{X}_{\tilde{y}=i}$ as

$$\boldsymbol{X}_{\tilde{y}=i} = \boldsymbol{X}_{\tilde{y}=i,y^*=j} \cup \boldsymbol{X}_{\tilde{y}=i,y^* \neq j}$$

We prove $\forall i \in [m], j \in [m], \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ by proving two claims:

Claim 1: $\boldsymbol{X}_{\tilde{y}=i,y^*=j} \subseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$

Claim 2: $\boldsymbol{X}_{\tilde{y}=i,y^* \neq j} \nsubseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$

We do not need to show $\boldsymbol{X}_{\tilde{y} \neq i,y^*=j} \nsubseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ and $\boldsymbol{X}_{\tilde{y} \neq i,y^* \neq j} \nsubseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ because the noisy labels $\tilde{y}$ are given, thus the confident joint (Eqn. 2.1) will never place them in the wrong bin of $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$. Thus, claim 1 and claim 2 suffice to show that $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$.

***Proof (Claim 1) of Case 2*:** Inspecting Eqn. (2.1) and Alg (1), by the construction of $\boldsymbol{C}_{\tilde{y},y^*}$, we have that $\forall \boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i}$,

$\hat{p}(\tilde{y}=j|y^*{=}j;\boldsymbol{x},\boldsymbol{\theta}) \geq t_j \longrightarrow \boldsymbol{X}_{\tilde{y}=i,y^*=j} \subseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$. When the left-hand side is true, all examples with noisy label i and hidden, true label j are counted in $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$.

Thus, it suffices to prove:

$$\forall \boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i}, \hat{p}(\tilde{y}=j|y^*{=}j;\boldsymbol{x},\boldsymbol{\theta}) \geq t_j \tag{2.5}$$

Because the predicted probabilities satisfy the ideal condition, $\hat{p}(\tilde{y}=j|y^*{=}j,\boldsymbol{x}) = p(\tilde{y}=j|y^*{=}j), \forall \boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i}$. Note the change from predicted probability, $\hat{p}$, to an exact probability, p. Thus by the ideal condition, the inequality in (2.5) can be written as $p(\tilde{y}=j|y^*{=}j) \geq t_j$, which we prove below:

$$\begin{aligned}
p(\tilde{y}=j|y^*{=}j) &\geq p(\tilde{y}=j|y^*{=}j)\cdot 1 && \triangleright \text{Identity} \\
&\geq p(\tilde{y}=j|y^*{=}j)\cdot \sum_{i\in[m]} p(y^*{=}i|\tilde{y}{=}j) && \\
&\geq \sum_{i\in[m]} p(\tilde{y}=j|y^*{=}j)\cdot p(y^*{=}i|\tilde{y}{=}j) && \triangleright \text{move product into sum} \\
&\geq \sum_{i\in[m]} p(\tilde{y}=j|y^*{=}i)\cdot p(y^*{=}i|\tilde{y}{=}j) && \triangleright \text{diagonal entry maximizes row} \\
&\geq t_j && \triangleright \text{Lemma 1, ideal condition}
\end{aligned}$$

***Proof (Claim 2) of Case 2*:** We prove $\boldsymbol{X}_{\tilde{y}=i,y^*\neq j} \nsubseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ by contradiction. Assume there exists some example $\boldsymbol{x}_k \in \boldsymbol{X}_{\tilde{y}=i,y^*=z}$ for $z \neq j$ such that $\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$. By claim 1, we have that $\boldsymbol{X}_{\tilde{y}=i,y^*=j} \subseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$, therefore, $\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=z}$.

Thus, for some example $\boldsymbol{x}_k$, we have that $\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ and also $\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=z}$.

However, this is a collision, and when a collision occurs the confident joint will break the tie with arg max. Because each diagonal entry of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its row

and column, $\boldsymbol{x}_k$ will always be binned into $\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y},y^*}[\tilde{y}_k][y_k^*]$ (the assignment from Claim 1).

This theorem also states $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$. This follows directly from the fact that $\forall i \in [m], j \in [m], \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$, i.e. the confident joint *exactly counts* the partitions $\boldsymbol{X}_{\tilde{y}=i,y^*=j}$ for all pairs $(i,j) \in [m] \times [m]$, thus $\boldsymbol{C}_{\tilde{y},y^*} = n\boldsymbol{Q}_{\tilde{y},y^*}$ and $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$. Omitting discretization error, the confident joint $\boldsymbol{C}_{\tilde{y},y^*}$, when normalized to $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$, is an exact estimator for $\boldsymbol{Q}_{\tilde{y},y^*}$. For example, if the noise rate is 0.39, but the dataset has only 5 examples in that class, the best possible estimate by removing errors is $2/5 = 0.4 \approxeq 0.39$.

□

Corollary 1.0 (Exact Estimation)**.** *For a noisy dataset,* $(\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ *and* $\boldsymbol{\theta}{:}\boldsymbol{x} \to \hat{p}(\tilde{y})$*, if* $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$ *is ideal and each diagonal entry of* $\boldsymbol{Q}_{\tilde{y}|y^*}$ *maximizes its row and column, and if* $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$*, then* $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$*.*

Proof. The result follows directly from Thm. 1. Because the confident joint *exactly counts* the partitions $\boldsymbol{X}_{\tilde{y}=i,y^*=j}$ for all pairs $(i,j) \in [m] \times [m]$ by Thm. 1, $\boldsymbol{C}_{\tilde{y},y^*} = n\boldsymbol{Q}_{\tilde{y},y^*}$, omitting discretization rounding errors. □

In the main text, Thm. 1 includes Corollary 1.0 for brevity. We have separated out Corollary 1.0 here to make apparent that the primary contribution of Thm. 1 is to prove $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$, from which the result of Corollary 1.0, namely that $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$ naturally follows, omitting discretization rounding errors.

Corollary 1.1 (Per-Class Robustness)**.** *For a noisy dataset,* $\boldsymbol{X} \coloneqq (\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ *and model* $\boldsymbol{\theta}{:}\boldsymbol{x} \to \hat{p}(\tilde{y})$*, if* $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ *is* ***per-class diffracted*** *without label collisions and each diagonal entry of* $\boldsymbol{Q}_{\tilde{y}|y^*}$ *maximizes its row, then* $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ *and* $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$*.*

Proof. Re-stating the meaning of **per-class diffracted**, we wish to show that if $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$ is diffracted with class-conditional noise s.t. $\forall j \in [m], \hat{p}(\tilde{y} = j; \boldsymbol{x}, \boldsymbol{\theta}) = \epsilon_j^{(1)} \cdot p^*(\tilde{y} = j | y^* = y_k^*) + \epsilon_j^{(2)}$ where $\epsilon_j^{(1)} \in \mathcal{R}, \epsilon_j^{(2)} \in \mathcal{R}$ (for any distribution) without label collisions and each diagonal entry of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its row, then $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ and $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$.

First note that linear combinations of real-valued $\epsilon_j^{(1)}$ and $\epsilon_j^{(2)}$ with the probabilities of class j for each example may result in some examples having $\hat{p}_{\boldsymbol{x},\tilde{y}=j} = \epsilon_j^{(1)} p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)} > 1$ or $\hat{p}_{\boldsymbol{x},\tilde{y}=j} = \epsilon_j^{(1)} p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)} < 0$. The proof makes no assumption about the validity of the model outputs and therefore holds when this occurs. Furthermore, confident learning does not require valid probabilities when finding label errors because confident learning depends on the *rank* principle, i.e., the rankings of the probabilities, not the values of the probabilities.

When there are no label collisions, the bins created by the confident joint are:

$$\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \coloneqq \{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : \hat{p}(\tilde{y} = j; \boldsymbol{x}, \boldsymbol{\theta}) \geq t_j\} \tag{2.6}$$

where

$$t_j = \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \hat{p}_{\boldsymbol{x},\tilde{y}=j}$$

WLOG: we re-formulate the error $\epsilon_j^{(1)} p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)}$ as $\epsilon_j^{(1)}(p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)})$.

Now, for diffracted (non-ideal) probabilities, we rearrange how the threshold t_j

changes for a given $\epsilon_j^{(1)}, \epsilon_j^{(2)}$:

$$
\begin{aligned}
t_j^{\epsilon_j} &= \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \epsilon_j^{(1)}(p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)}) \\
t_j^{\epsilon_j} &= \epsilon_j^{(1)} \left(\mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} p^*_{\boldsymbol{x},\tilde{y}=j} + \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \epsilon_j^{(2)} \right) \\
t_j^{\epsilon_j} &= \epsilon_j^{(1)} \left(t_j^* + \epsilon_j^{(2)} \cdot \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} 1 \right) \\
t_j^{\epsilon_j} &= \epsilon_j^{(1)}(t_j^* + \epsilon_j^{(2)})
\end{aligned}
$$

Thus, for per-class diffracted (non-ideal) probabilities, Eqn. (2.6) becomes

$$
\begin{aligned}
\hat{\boldsymbol{X}}^{\epsilon_j}_{\tilde{y}=i,y^*=j} &= \{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : \epsilon_j^{(1)}(p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)}) \geq \epsilon_j^{(1)}(t_j^* + \epsilon_j^{(2)})\} \\
&= \{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j^*\} \\
&= \boldsymbol{X}_{\tilde{y}=i,y^*=j} \qquad \triangleright \text{ by Thm. (1)}
\end{aligned}
$$

In the second to last step, we see that the formulation of the label errors is the formulation of $\boldsymbol{C}_{\tilde{y},y^*}$ for *ideal* probabilities, which we proved yields exact label errors and consistent estimation of $\boldsymbol{Q}_{\tilde{y},y^*}$ in Thm. 1. This concludes the proof. Note that we eliminate the need for the assumption that each diagonal entry of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its column because this assumption is only used in the proof of Thm. 1 when collisions occur, but here we only consider the case when there are no collisions.

□

Theorem 2 (Per-Example Robustness)**.** *For a noisy dataset,* $\boldsymbol{X} \coloneqq (\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ *and model* $\boldsymbol{\theta}{:}\boldsymbol{x} \to \hat{p}(\tilde{y})$*, if* $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ *is* ***per-example diffracted*** *without label collisions and each diagonal entry of* $\boldsymbol{Q}_{\tilde{y}|y^*}$ *maximizes its row, then* $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \approxeq \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ *and* $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$.

Proof. We consider the nontrivial real-world setting when a learning model $\boldsymbol{\theta}{:}\boldsymbol{x}{\to}\hat{p}(\tilde{y})$ outputs erroneous, non-ideal predicted probabilities with an error term added for every example, across every class, such that $\forall \boldsymbol{x} \in \boldsymbol{X}, \forall j \in [m],\ \hat{p}_{\boldsymbol{x},\tilde{y}=j} = p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j}$. As a notation reminder $p^*_{\boldsymbol{x},\tilde{y}=j}$ is shorthand for the ideal probabilities $p^*(\tilde{y}=j|y^*=y^*_k) + \epsilon_{\boldsymbol{x},\tilde{y}=j}$, and $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ is shorthand for the predicted probabilities $\hat{p}(\tilde{y}=j;\boldsymbol{x},\boldsymbol{\theta})$.

The predicted probability error $\epsilon_{\boldsymbol{x},\tilde{y}=j}$ is distributed uniformly with no other constraints. We use $\epsilon_j \in \mathcal{R}$ to represent the mean of $\epsilon_{\boldsymbol{x},\tilde{y}=j}$ per class, i.e. $\epsilon_j = \mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}}\, \epsilon_{\boldsymbol{x},\tilde{y}=j}$, which can be seen by looking at the form of the uniform distribution in Eqn. (2.4). If we wanted, we could add the constraint that $\epsilon_j = 0, \forall j \in [m]$, which would simplify the theorem and the proof, but this result is not as general. Instead, we prove exact label error finding and joint estimation without this constraint.

We re-iterate the form of the error in Eqn. (2.4) here ($\mathcal{U}$ denotes a uniform distribution):

$$\epsilon_{\boldsymbol{x},\tilde{y}=j} \sim \begin{cases} \mathcal{U}(\epsilon_j + t_j - p^*_{\boldsymbol{x},\tilde{y}=j}\,,\, \epsilon_j - t_j + p^*_{\boldsymbol{x},\tilde{y}=j}] & p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j \\ \mathcal{U}[\epsilon_j - t_j + p^*_{\boldsymbol{x},\tilde{y}=j}\,,\, \epsilon_j + t_j - p^*_{\boldsymbol{x},\tilde{y}=j}) & p^*_{\boldsymbol{x},\tilde{y}=j} < t_j \end{cases}$$

When there are no label collisions, the bins created by the confident joint are:

$$\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \coloneqq \{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : \hat{p}_{\boldsymbol{x},\tilde{y}=j} \geq t_j\} \tag{2.7}$$

where

$$t_j = \frac{1}{|X_{\tilde{y}=j}|} \sum_{\boldsymbol{x}\in\boldsymbol{X}_{\tilde{y}=j}} \hat{p}_{\boldsymbol{x},\tilde{y}=j}$$

Rewriting the threshold t_j to include the error terms $\epsilon_{\boldsymbol{x},\tilde{y}=j}$ and ϵ_j, we have

$$
\begin{aligned}
t_j^{\epsilon_j} &= \frac{1}{|X_{\tilde{y}=j}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} \\
t_j^{\epsilon_j} &= \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} p^*_{\boldsymbol{x},\tilde{y}=j} + \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \epsilon_{\boldsymbol{x},\tilde{y}=j} \\
&= t_j + \epsilon_j
\end{aligned}
$$

where the last step uses the fact that $\epsilon_{\boldsymbol{x},\tilde{y}=j}$ is uniformly distributed over $\boldsymbol{x} \in \boldsymbol{X}$ and $n \to \infty$ so that $\mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \epsilon_{\boldsymbol{x},\tilde{y}=j} = \mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}} \epsilon_{\boldsymbol{x},\tilde{y}=j} = \epsilon_j$. We now complete the proof by showing that

$$
p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} \geq t_j + \epsilon_j \iff p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j
$$

If this statement is true then the subsets created by the confident joint in Eqn. 2.7 are unaltered and therefore $\hat{\boldsymbol{X}}^{\epsilon_{\boldsymbol{x},\tilde{y}=j}}_{\tilde{y}=i,y^*=j} = \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \overset{Thm.\ 1}{=} \boldsymbol{X}_{\tilde{y}=i,y^*=j}$, where $\hat{\boldsymbol{X}}^{\epsilon_{\boldsymbol{x},\tilde{y}=j}}_{\tilde{y}=i,y^*=j}$ denotes the confident joint subsets for $\epsilon_{\boldsymbol{x},\tilde{y}=j}$ predicted probabilities.

Now we complete the proof. From the distribution for $\epsilon_{\boldsymbol{x},\tilde{y}=j}$ (Eqn. 2.4), we have that

$$
\begin{aligned}
p^*_{\boldsymbol{x},\tilde{y}=j} < t_j &\implies \epsilon_{\boldsymbol{x},\tilde{y}=j} < \epsilon_j + t_j - p^*_{\boldsymbol{x},\tilde{y}=j} \\
p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j &\implies \epsilon_{\boldsymbol{x},\tilde{y}=j} \geq \epsilon_j + t_j - p^*_{\boldsymbol{x},\tilde{y}=j}
\end{aligned}
$$

Re-arranging

$$
\begin{aligned}
p^*_{\boldsymbol{x},\tilde{y}=j} < t_j &\implies p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} < t_j + \epsilon_j \\
p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j &\implies p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} \geq t_j + \epsilon_j
\end{aligned}
$$

Using the contrapositive, we have

$$p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} \geq t_j + \epsilon_j \implies p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j$$
$$p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j \implies p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} \geq t_j + \epsilon_j$$

Combining, we have

$$p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} \geq t_j + \epsilon_j \iff p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j$$

Therefore,

$$\hat{\boldsymbol{X}}^{\epsilon_{\boldsymbol{x}},\tilde{y}=j}_{\tilde{y}=i,y^*=j} \overset{Thm.\ 1}{=} \boldsymbol{X}_{\tilde{y}=i,y^*=j}$$

The last line follows from the fact that we have reduced $\hat{\boldsymbol{X}}^{\epsilon_{\boldsymbol{x}},\tilde{y}=j}_{\tilde{y}=i,y^*=j}$ to counting the same condition ($p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j$) as the confident joint counts under ideal probabilities in Thm (1). Thus, we maintain exact finding of label errors and exact estimation (Corollary 1.1) holds under no label collisions. The proof applies for finite datasets because we ignore discretization error; however, for equality, the proof requires the assumption $n \to \infty$, which is used in this step: $\mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \epsilon_{\boldsymbol{x},\tilde{y}=j} \overset{n\to\infty}{=} \mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}} \epsilon_{\boldsymbol{x},\tilde{y}=j} = \epsilon_j$. Thus, we use approximately equals in the statement of the theorem.

Note that while we use a uniform distribution in Eqn. 2.4, any bounded symmetric distribution with mode $\epsilon_j = \mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}} \epsilon_{\boldsymbol{x},j}$ is sufficient. Observe that the bounds of the distribution are non-vacuous (they do not collapse to a single value e_j) because $t_j \neq p^*_{\boldsymbol{x},\tilde{y}=j}$ by Lemma 1.

□

2.5 Experiments

This section empirically validates CL on CIFAR (Krizhevsky and Hinton, 2009) and ImageNet (Russakovsky et al., 2015) benchmarks. Sec. 2.5.1 presents CL performance on noisy examples in CIFAR where true labels are presumed known. Sec. 2.5.2 shows real-world label errors found in the original, unperturbed MNIST, ImageNet, WebVision, and Amazon Reviews datasets, and shows performance advantages using cleaned data provided by CL to train ImageNet. Unless otherwise specified, we compute out-of-sample predicted probabilities $\hat{\boldsymbol{P}}_{k,j}$ using four-fold cross-validation and ResNet architectures.

2.5.1 Asymmetric Label Noise on CIFAR-10 dataset

We evaluate CL on three criteria: (a) joint estimation (Fig. 2-2), (b) accuracy finding label errors (Table 2.3), and (c) accuracy learning with noisy labels (Table 2.1).

Noise Generation Following prior work by Sukhbaatar et al. (2015); Goldberger and Ben-Reuven (2017), we verify CL performance on the commonly used asymmetric label noise, where the labels of error-free/clean data are randomly flipped, for its resemblance to real-world noise. We generate noisy data from clean data by randomly switching some labels of training examples to different classes non-uniformly according to a randomly generated $\boldsymbol{Q}_{\tilde{y}|y^*}$ noise transition matrix. We generate $\boldsymbol{Q}_{\tilde{y}|y^*}$ matrices with different traces to run experiments for different noise levels. The noise matrices used in our experiments are in the Appendix in Fig. A-2. We generate noise in the CIFAR-10 training dataset across varying *sparsities*, the fraction of off-diagonal elements in $\boldsymbol{Q}_{\tilde{y},y^*}$ that are zero, and the percent of incorrect labels (noise). We evaluate all models on the unaltered test set.

Baselines and our method In Table 2.1, we compare CL performance versus seven recent highly competitive approaches and a vanilla baseline for multiclass learning with noisy labels on CIFAR-10, including *INCV* (Chen et al., 2019) which finds clean data with multiple iterations of cross-validation and then trains on the clean set, *SCE-loss* (symmetric cross entropy) (Wang et al., 2019) which adds a reverse cross entropy term for loss-correction, *Mixup* (Zhang et al., 2018) which linearly combines examples and labels to augment data, *MentorNet* (Jiang et al., 2018) which uses curriculum learning to avoid noisy data in training, *Co-Teaching* (Han et al., 2018) which trains two models in tandem to learn from clean data, *S-Model* (Goldberger and Ben-Reuven, 2017) which uses an extra softmax layer to model noise during training, and *Reed* (Reed et al., 2015) which uses loss-reweighting; and a *Baseline* model that denotes a vanilla training with the noisy labels.

Training settings All models are trained using ResNet-50 with the common setting: learning rate 0.1 for epoch [0,150), 0.01 for epoch [150,250), 0.001 for epoch [250,350); momentum 0.9; and weight decay 0.0001, except *INCV*, *SCE-loss*, and *Co-Teaching* which are trained using their official GitHub code. Settings are copied from the kuangliu/pytorch-cifar GitHub open-source code and were not tuned by hand. We report the highest score across hyper-parameters $\alpha \in \{1, 2, 4, 8\}$ for *Mixup* and $p \in \{0.7, 0.8, 0.9\}$ for *MentorNet*. For fair comparison with *Co-Teaching*, *INCV*, and *MentorNet*, we also train using the *co-teaching* approach with forget rate $= 0.5 \times$ [noise fraction], and we report the max accuracy of the two trained models for each method. We observe that dropping the last partial batch of each epoch during training improves stability by avoiding weight updates (in some cases from a single noisy example). Exactly the same noisy labels are used for training all models for each column of Table 2.1. For our method, we fix its hyper-parameter, *i.e.*

Table 2.1: Test accuracy (%) of confident learning versus recent methods for learning with noisy labels in CIFAR-10. Scores reported for CL methods are averaged over ten trials with standard deviations shown in Table 2.2. CL methods estimate label errors, remove them, then train on the cleaned data. Whereas other methods decrease in performance from low sparsity (e.g., 0.0) to high sparsity (e.g., 0.6), CL methods are robust across sparsity, as indicated by comparing the two column-wise red highlighted cells.

Noise	20%				40%				70%			
Sparsity	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.6
CL: $\boldsymbol{C}_{\text{confusion}}$	89.6	89.4	90.2	89.9	83.9	83.9	83.2	84.2	31.5	39.3	33.7	30.6
CL: PBC	90.5	90.1	90.6	90.7	84.8	85.5	85.3	86.2	33.7	40.7	35.1	31.4
CL: $\boldsymbol{C}_{\tilde{y},y^*}$	**91.1**	**90.9**	**91.1**	**91.3**	86.7	86.7	86.6	86.9	32.4	**41.8**	34.4	34.5
CL: C+NR	90.8	90.7	91.0	91.1	**87.1**	**86.9**	**86.7**	**87.2**	**41.1**	41.7	39.0	32.9
CL: PBNR	90.7	90.5	90.9	90.9	**87.1**	86.8	86.6	**87.2**	41.0	**41.8**	**39.1**	**36.4**
INCV (Chen et al., 2019)	87.8	88.6	89.6	89.2	84.4	76.6	85.4	73.6	28.3	25.3	34.8	29.7
Mixup (Zhang et al., 2018)	85.6	86.8	87.0	84.3	76.1	75.4	68.6	59.8	32.2	31.3	32.3	26.9
SCE-loss (Wang et al., 2019)	87.2	87.5	88.8	84.4	76.3	74.1	64.9	58.3	33.0	28.7	30.9	24.0
MentorNet (Jiang et al., 2018)	84.9	85.1	83.2	83.4	64.4	64.2	62.4	61.5	30.0	31.6	29.3	27.9
Co-Teaching (Han et al., 2018)	81.2	81.3	81.4	80.6	62.9	61.6	60.9	58.1	30.5	30.2	27.7	26.0
S-Model (Goldberger et al., 2017)	80.0	80.0	79.7	79.1	58.6	61.2	59.1	57.5	28.4	28.5	27.9	27.3
Reed (Reed et al., 2015)	78.1	78.9	80.8	79.3	60.5	60.4	61.2	58.6	29.0	29.4	29.1	26.8
Baseline	78.4	79.2	79.0	78.2	60.2	60.8	59.6	57.3	27.0	29.7	28.2	26.8

the number of folds in cross-validation across different noise levels, and do not tune it on the validation set.

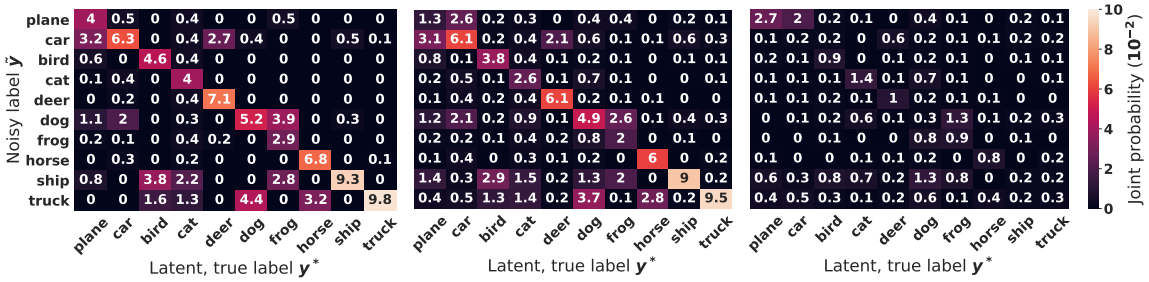
For each CL method, sparsity, and noise setting, we report the mean accuracy in Table 2.1, averaged over ten trials, by varying the random seed and initial weights of the neural network for training. Standard deviations are reported in Table 2.2 to improve readability. For each column in Table 2.1, the corresponding standard deviations in Table 2.2 are significantly less than the performance difference between CL methods and baseline methods. Notably, all standard deviations are significantly ($\sim$10x) less than the mean performance difference between the top-performing CL method and baseline methods for each setting, averaged over random weight initialization.

Table 2.2: Standard deviations (% units) associated with the mean score (over ten trials) for scores reported for CL methods in Table 2.1. Each trial uses a different random seed and network weight initialization. No standard deviation exceeds 2%.

Noise	20%				40%				70%			
Sparsity	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.6
CL: $\boldsymbol{C}_{\text{confusion}}$	0.07	0.10	0.17	0.08	0.19	0.22	0.23	0.20	0.93	0.24	0.13	0.26
CL: PBC	0.14	0.12	0.11	0.10	0.15	0.17	0.16	0.10	0.12	0.22	0.11	0.30
CL: $\boldsymbol{C}_{\tilde{y},y^*}$	0.17	0.09	0.17	0.11	0.10	0.20	0.09	0.13	1.02	0.15	0.18	1.63
CL: C+NR	0.09	0.10	0.08	0.08	0.11	0.14	0.16	0.10	0.42	0.33	0.26	1.90
CL: PBNR	0.15	0.09	0.09	0.10	0.18	0.10	0.15	0.12	0.26	0.28	0.24	1.43

Standard deviations are only reported for CL methods because of difficulty reproducing consistent results for some of the other methods (we discuss memory issues preventing consistent results for one of the methods in Appendix A.2).

In Fig. 2-2, we visualize the quality of CL joint estimation in a challenging high-noise (40%), high-sparsity (60%) regime on CIFAR. Subfigure (2-2a) demonstrates high sparsity in the latent true joint $\boldsymbol{Q}_{\tilde{y},y^*}$, with over half the noise in just six noise rates. Yet, as can be seen in Subfigures (2-2b) and (2-2c), CL still estimates over 80%



(a) True $\boldsymbol{Q}_{\tilde{y},y^*}$ (unknown to CL) (b) CL estimated $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ (c) Absolute diff. $|\boldsymbol{Q}_{\tilde{y},y^*} - \hat{\boldsymbol{Q}}_{\tilde{y},y^*}|$

Figure 2-2: Our estimation of the joint distribution of noisy labels and true labels for CIFAR with 40% label noise and 60% sparsity. Observe the similarity (RSME = .004) between (a) and (b) and the low absolute error in every entry in (c). Probabilities are scaled up by 100.

of the entries of $\boldsymbol{Q}_{\tilde{y},y^*}$ within an absolute difference of .005.

These results in Fig. 2-2 empirically substantiate the theoretical bounds of Thm. 2 in settings beyond those covered in Section 2.3 (which focused on conditions for *exact* estimation). Here, the error in the predicted probabilities exceeds the conditions of Thm. 2, resulting in imperfect estimation, yet the overall absolute difference (Subfigure 2-2c) remains low.

The most egregious mistake made by CL in Fig. 2-2 is shown in the *plane* class of $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ in Subfigure 2-2b where the model had difficulty disambiguating planes and cars (shown by $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}[plane][car] = 2.6$) because of the large noise added to the latent $\boldsymbol{Q}_{\tilde{y},y^*}[car][plane] = 3.2$. The resulting $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ has a diagonal element ($\hat{\boldsymbol{Q}}_{\tilde{y},y^*}[plane][plane] = 1.3$) which does not maximize its row. This is reasonable because the max-diagonal condition only applies for exact estimation (c.f., Thm. 2), which we do not achieve as shown in Subfigure 2-2c. Further, the max-diagonal condition only applies to the latent, true $\boldsymbol{Q}_{\tilde{y},y^*}$, not the CL estimated $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$. These observations emphasize that confident learning does not impose the max-diagonal condition while estimating $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$, extending its usage to real-world settings where the conditions of the theory in Section 2.3 may no longer hold.

We also evaluate CL's accuracy in finding label errors. In Table 2.3, we compare five variants of CL methods across noise and sparsity and report their precision, recall, and F1 in recovering the true label. The results show that CL is able to find the label errors with high recall and reasonable F1.

Robustness to Sparsity Table 2.1 reports CIFAR test accuracy for learning with noisy labels across noise amount and sparsity, where the first five rows report our CL approaches. As shown, CL consistently performs well compared to prior art across all noise and sparsity settings. We observe significant improvement in high-noise and/or

Table 2.3: Mean accuracy, F1, precision, and recall measures of CL methods for finding label errors in CIFAR-10, averaged over ten trials.

Measure	Accuracy (%) ± Std. Dev. (%)				F1 (%)				Precision (%)				Recall (%)			
Noise	20%		40%		20%		40%		20%		40%		20%		40%	
Sparsity	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6
CL: $\boldsymbol{C}_{\text{confusion}}$	84±0.07	85±0.09	85±0.24	81±0.21	71	72	84	79	56	58	74	70	**98**	**97**	**97**	**90**
CL: $\boldsymbol{C}_{\tilde{y},y^*}$	89±0.15	**90**±0.10	86±0.15	**84**±0.12	75	78	84	**80**	**67**	**70**	78	77	86	88	91	84
CL: PBC	88±0.22	88±0.11	86±0.17	82±0.13	76	76	84	79	64	65	76	74	96	93	94	85
CL: PBNR	89±0.11	**90**±0.08	**88**±0.12	**84**±0.11	77	**79**	**85**	**80**	65	68	**82**	**79**	93	94	88	82
CL: C+NR	**90**±0.21	**90**±0.10	87±0.23	83±0.14	**78**	78	84	78	**67**	69	**82**	**79**	93	90	87	78

high-sparsity regimes. The simplest CL method $CL : \boldsymbol{C}_{\text{confusion}}$ performs similarly to *INCV* and comparably to prior art with best performance by $\boldsymbol{C}_{\tilde{y},y^*}$ across all noise and sparsity settings. The results validate the benefit of directly modeling the joint noise distribution and show that our method is competitive compared to highly competitive, robust learning methods.

Fig. 2-3 shows the absolute difference of the true joint $\boldsymbol{Q}_{\tilde{y},y^*}$ and the joint distribution estimated using confident learning $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ on CIFAR-10, for 20%, 40%, and 70% label noise, 20%, 40%, and 60% sparsity, for all pairs of classes in the joint distribution of label noise. Observe that in moderate noise regimes between 20% and 40% noise, CL accurately estimates nearly every entry in the joint distribution of label noise. The high accuracy demonstrated in Fig. 2-3 supports our theoretical finding that CL exactly estimates the joint distribution of labels in conditions allowing for noise in every predicted probability, for every example (c.f., Thm. 2).

To understand why CL performs well, we evaluate CL joint estimation across noise and sparsity with RMSE in Table 2.4 and estimated $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ in Fig. 2-3. For the 20% and 40% noise settings, on average, CL achieves an RMSE of .004 relative to the true joint $\boldsymbol{Q}_{\tilde{y},y^*}$ across all sparsities. The simplest CL variant, $\boldsymbol{C}_{\text{confusion}}$ normalized

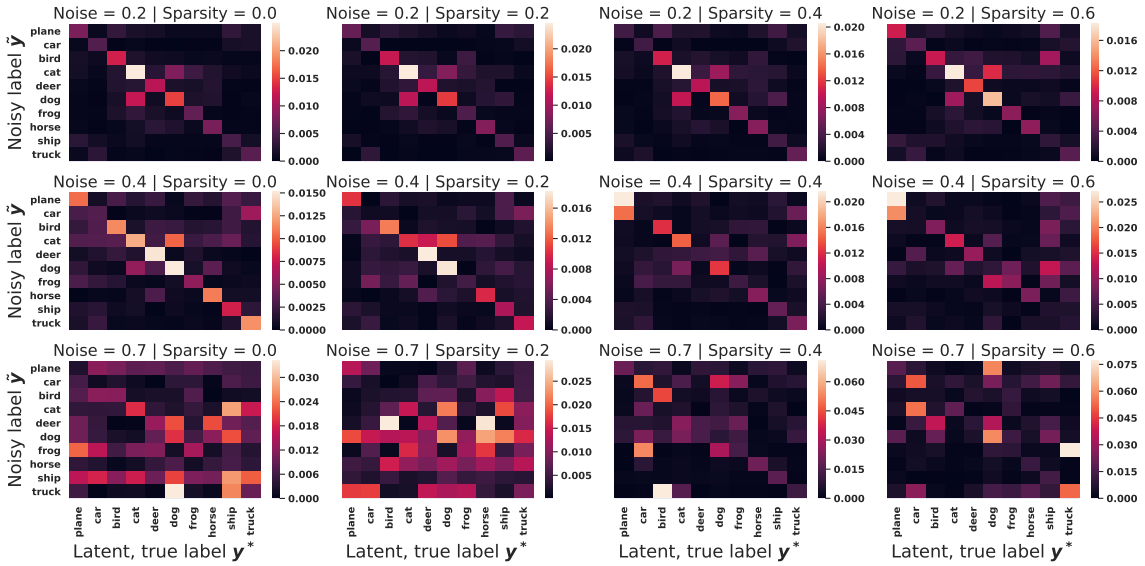


Figure 2-3: Absolute difference of the true joint $\boldsymbol{Q}_{\tilde{y},y^*}$ and the joint distribution estimated using confident learning $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ on CIFAR-10, for 20%, 40%, and 70% label noise, 20%, 40%, and 60% sparsity, for all pairs of classes in the joint distribution of label noise.

via Eqn. (2.3) to obtain $\hat{\boldsymbol{Q}}_{\text{confusion}}$, achieves a slightly worse RMSE of .006.

In Table 2.4, we estimate the $\boldsymbol{Q}_{\tilde{y},y^*}$ using the confusion-matrix $\boldsymbol{C}_{\text{confusion}}$ approach normalized via Eqn. (2.3) and compare this $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$, estimated by normalizing the CL approach with the confident joint $\boldsymbol{C}_{\tilde{y},y^*}$, for various amounts of noise and sparsity in $\boldsymbol{Q}_{\tilde{y},y^*}$. Table 2.4 shows improvement using $\boldsymbol{C}_{\tilde{y},y^*}$ over $\boldsymbol{C}_{\text{confusion}}$, low RMSE scores, and robustness to sparsity in moderate-noise regimes.

In Table A.1 in the Appendices, we report the training time required to achieve the accuracies reported in Table 2.1 for INCV and confident learning. As shown in Table A.1, INCV training time exceeded 20 hours. In comparison, CL takes less than three hours on the same machine: an hour for cross-validation, less than a minute to find errors, and an hour to re-train.

Table 2.4: RMSE error of $\boldsymbol{Q}_{\tilde{y},y^*}$ estimation on CIFAR-10 using $\boldsymbol{C}_{\tilde{y},y^*}$ to estimate $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ compared with using the baseline approach $\boldsymbol{C}_{\text{confusion}}$ to estimate $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$.

Noise	0.2				0.4				0.7			
Sparsity	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.6
$\|\hat{\boldsymbol{Q}}_{\tilde{y},y^*} - \boldsymbol{Q}_{\tilde{y},y^*}\|_2$	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	**0.005**	0.011	**0.010**	0.015	**0.017**
$\|\hat{\boldsymbol{Q}}_{confusion} - \boldsymbol{Q}_{\tilde{y},y^*}\|_2$	0.006	0.006	0.005	0.005	0.005	0.005	0.005	0.007	0.011	0.011	0.015	0.019

2.5.2 Real-world Label Errors in the ImageNet Train Dataset

Russakovsky et al. (2015) suggest label errors exist in ImageNet due to human error, but to our knowledge few attempts have been made to find label errors in the ILSVRC 2012 training set, characterize them, or re-train without them. Here, we consider each application. We use ResNet18 and ResNet50 architectures with standard settings: 0.1 initial learning rate, 90 training epochs, and 0.9 momentum.

Table 2.5: Ten largest non-diagonal entries in the confident joint $\boldsymbol{C}_{\tilde{y},y^*}$ for ImageNet train set used for ontological issue discovery. A duplicated class detected by CL is highlighted in red.

$\boldsymbol{C}_{\tilde{y},y^*}$	$\tilde{y}$ name	y^* name	$\tilde{y}$ nid	y^* nid	$\boldsymbol{C}_{\text{confusion}}$	$\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$
645	projectile	missile	n04008634	n03773504	494	0.00050
539	tub	bathtub	n04493381	n02808440	400	0.00042
476	breastplate	cuirass	n02895154	n03146219	398	0.00037
437	green_lizard	chameleon	n01693334	n01682714	369	0.00034
435	chameleon	green_lizard	n01682714	n01693334	362	0.00034
433	missile	projectile	n03773504	n04008634	362	0.00034
417	maillot	maillot	n03710637	n03710721	338	0.00033
416	horned_viper	sidewinder	n01753488	n01756291	336	0.00033
410	corn	ear	n12144580	n13133613	333	0.00032
407	keyboard	space_bar	n04505470	n04264628	293	0.00032

Ontological discovery for dataset curation Because ImageNet is an one-hot class dataset, the classes are required to be mutually exclusive. Using ImageNet

as a case study, we observe auto-discovery of ontological issues at the class level in Table 2.5, operationalized by listing the 10 largest non-diagonal entries in $\boldsymbol{C}_{\tilde{y},y^*}$. For example, the class *maillot* appears twice, the existence of *is-a* relationships like *bathtub is a tub*, misnomers like *projectile* and *missile*, and unanticipated issues caused by words with multiple definitions like *corn* and *ear*. We include the baseline $\boldsymbol{C}_{\text{confusion}}$ to show that while $\boldsymbol{C}_{\text{confusion}}$ finds fewer label errors than $\boldsymbol{C}_{\tilde{y},y^*}$, they rank ontological issues similarly.

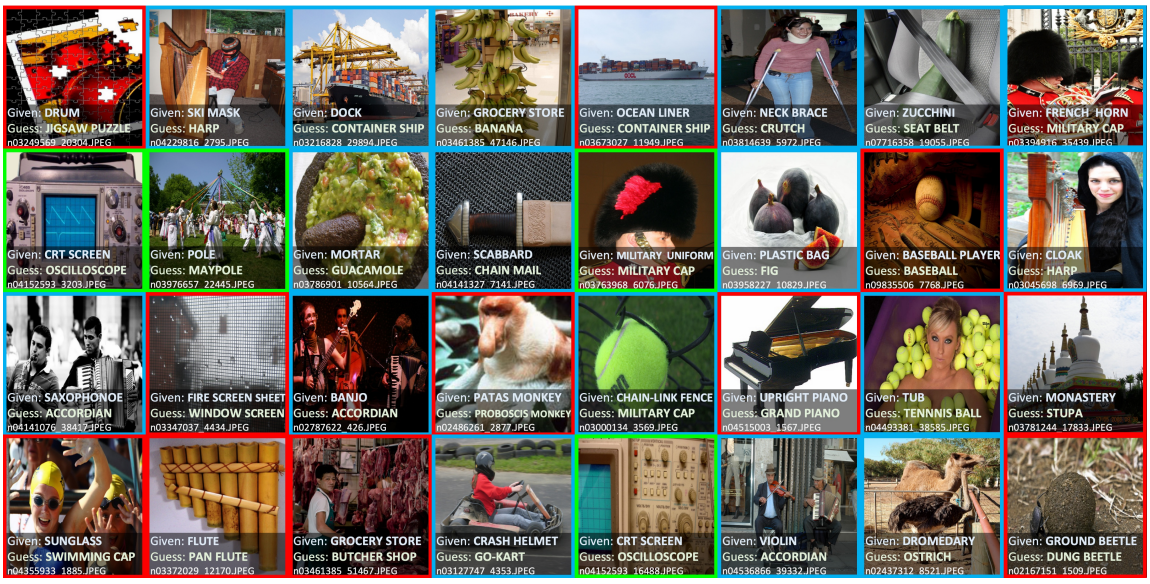
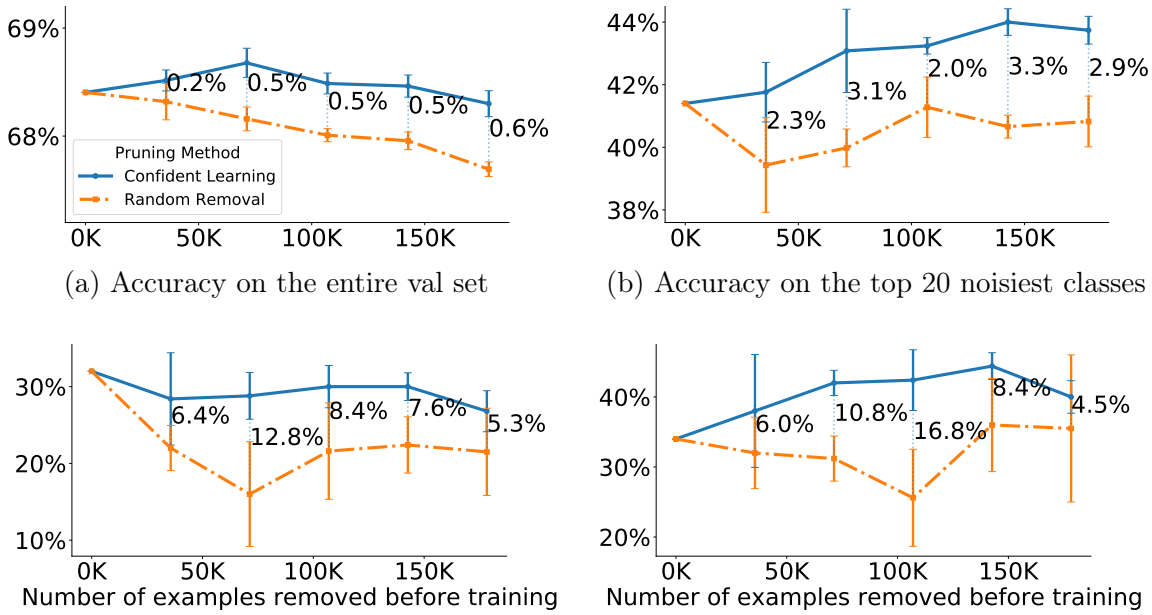


Figure 2-4: Top 32 (ordered automatically by normalized margin) identified label issues in the 2012 ILSVRC ImageNet train set using CL: PBNR. Errors are boxed in red. Ontological issues are boxed in green. Multi-label images are boxed in blue. (The top-left image is an edge case that could also reasonably be labeled as "multi-label" although it does not actually contain a real drum, only a partial image of one.)

Finding label issues Fig. 2-4 depicts the top 16 label issues found using CL: PBNR with ResNet50 ordered by the normalized margin. We use the term *issue* versus *error* because examples found by CL consist of a mixture of multi-label images,



(a) Accuracy on the entire val set

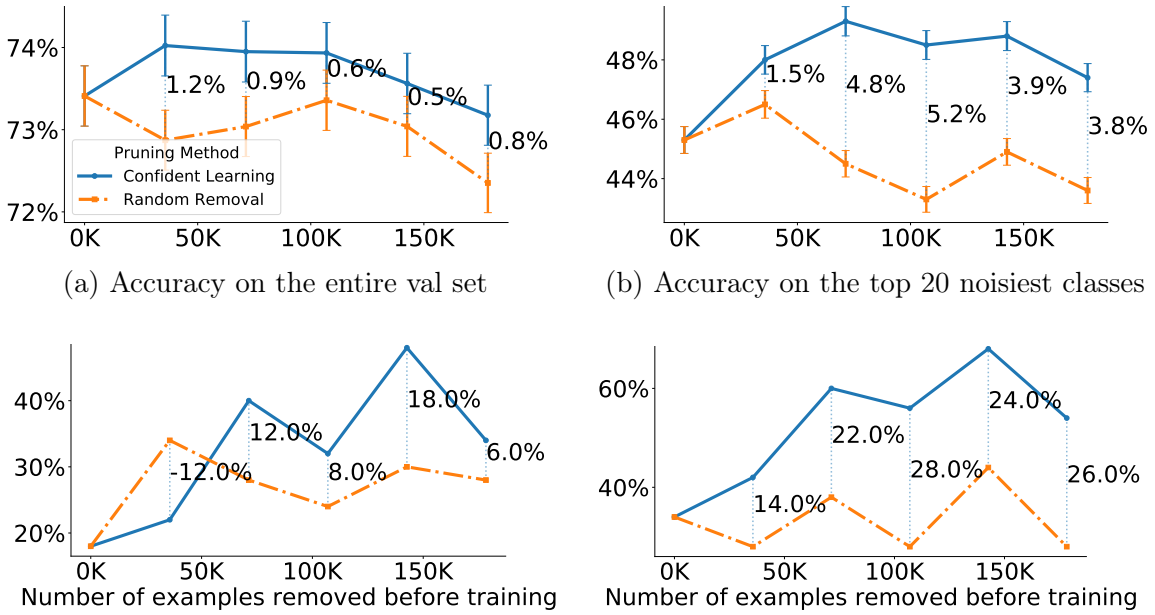
(b) Accuracy on the top 20 noisiest classes

(c) Accuracy on the noisiest class: foxhound

(d) Accuracy on erroneous class: maillot

Figure 2-5: ResNet-18 Validation Accuracy on ImageNet (ILSVRC2012) when 20%, 40%, ..., 100% of the label issues found using confident learning are removed prior to training (blue, solid line), compared with random examples removed prior to training (orange, dash-dotted line). Each subplot is read from left-to-right as incrementally more CL-identified issues are removed prior to training (shown by the x-axis). The translucent black dotted vertical bars measure the improvement when removing examples with CL vs random examples. Each point in all subfigures represents an independent training of ResNet-18 from scratch. Each point on the graph depicts the average accuracy of 5 trials (varying random seeding and weight initialization). The capped, colored vertical bars depict the standard deviation.

ontological issues, and actual label errors. Examples of each are indicated by colored borders in the figure. To evaluate CL in the absence of true labels, we conducted a small-scale human validation on a random sample of 500 errors (as identified using CL: PBNR) and found 58% were either multi-label, ontological issues, or errors. ImageNet data are often presumed error-free, yet ours is the first attempt to identify label errors automatically in ImageNet training images.



(a) Accuracy on the entire val set

(b) Accuracy on the top 20 noisiest classes

(c) Accuracy on the noisiest class: foxhound

(d) Accuracy on erroneous class: maillot

Figure 2-6: Replication of the experiments in Fig. 2-5 with ResNet-50. Each point in each subfigure depicts the accuracy of a single trial (due to computational limitations). The x-axis of each plot denotes the number of examples removed. Error bars, shown by the colored vertical lines, are estimated via Clopper-Pearson intervals for subfigures (a) and (b). For additional information, see the caption of Fig. 2-5.

Training ResNet on ImageNet with label issues removed By providing cleaned data for training, we explore how CL can be used to achieve similar or better validation accuracy on ImageNet when trained with less data. To understand the performance differences, we train ResNet-18 (Fig. 2-5) on progressively less data, removing 20%, 40%,..., 100% of ImageNet train set label issues identified by CL and training from scratch each time. Fig. 2-5 depicts the top-1 validation accuracy when training with cleaned data from CL versus removing uniformly random examples on each of (a) the entire ILSVRC validation set, (b) the 20 (noisiest) classes with the smallest diagonal in $\boldsymbol{C}_{\tilde{y},y^*}$, (c) the foxhound class, which has the smallest diagonal

in $\boldsymbol{C}_{\tilde{y},y^*}$, and (d) the maillot class, a known erroneous class, duplicated accidentally in ImageNet, as previously published (Hoffman et al., 2015), and verified (c.f. line 7 in Table 2.5). For readability, we plot the best performing CL method at each point and provide the individual performance of each CL method in the Appendix (see Fig. A-1). For the case of a single class, as shown in Fig. 2-5(c) and 2-5(d), we show the recall using the model's top-1 prediction, hence the comparatively larger variance in classification accuracy reported compared to (a) and (b). We observed that CL outperforms the random removal baseline in nearly all experiments, and improves on the no-data-removal baseline accuracy, depicted by the left-most point in the subfigures, on average over the five trials for the 1,000 and 20 class settings, as shown in Fig. 2-5(a) and 2-5(b). To verify the result is not model-specific, we repeat each experiment for a single trial with ResNet-50 (Fig. 2-6) and find that CL similarly outperforms the random removal baseline.

Although accuracy increases in the noisiest classes as incrementally more label issues found by CL are removed (Subfigures 2-5b, 2-5c and 2-5d), the overall accuracy (Subfigure 2-5a) starts to decrease beyond removal of 40% of the label issues found by CL – likely because the significant reduction in available training data reduces accuracy on the less-noisy classes.

These results suggest that CL can reduce the size of a real-world noisy training dataset by 10% while still moderately improving the validation accuracy (Subfigures 2-5a and 2-5b) and significantly improving the validation accuracy on the erroneous maillot class (Subfigures 2-5d and 2-6d). While we find CL methods may improve the standard ImageNet training on clean training data by filtering out a subset of training examples, the significance of this result lies not in the magnitude of improvement, but as a warrant of exploration in the use of cleaning methods when training with ImageNet, which is typically assumed to have correct labels. Whereas many of the

label issues in ImageNet are due to multi-labeled examples (Yun et al., 2021), next we consider a dataset with disjoint classes.

2.5.3 Amazon Reviews Dataset: CL using logistic regression on noisy text data

The Amazon Reviews dataset is a corpus of textual reviews labeled with 1-star to 5-star ratings from Amazon customers used to benchmark sentiment analysis models (He and McAuley, 2016). We study the 5-core (9.9 GB) variant of the dataset – the subset of data in which all users and items have at least 5 reviews. 2-star and 4-star reviews are removed due to ambiguity with 1-star and 5-star reviews, respectively. Left in the dataset, 2-star and 4-star reviews could inflate error counts, making CL appear to be more effective than it is.

This subsection serves three goals. First, we use a logistic regression classifier, as opposed to a deep-learning model, for our experiments in this section to evaluate CL for non-deep-learning methods. Second, we seek to understand how CL may improve learning with noise in the label space of text data, but not noise in the text data itself (e.g. typos). Towards this goal, we consider non-empty reviews with more "helpful" up-votes than down-votes. The resulting dataset consists of approximately ten million reviews. Finally, Thm. 2 shows that CL is robust to class-imbalance, but datasets like ImageNet and CIFAR-10 are balanced by construction: the Amazon Reviews dataset, however, is naturally and extremely imbalanced. The distribution of given labels (i.e., the noisy prior), is: 9% 1-star reviews:, 12% 3-star reviews, and 79% 5-star reviews. We seek to understand if CL can find label errors and improve performance in learning with noisy labels in this class-imbalanced setting.

Table 2.6: Top 20 CL-identified label issues in the Amazon Reviews text dataset using CL: C+NR, ordered by normalized margin. A logistic regression classifier trained on fastText embeddings is used to obtain out-of-sample predicted probabilities. Most errors are reasonable, with the exception of sarcastic reviews, which are poorly modeled by the bag-of-words model.

Review	Given Label	CL Guess
A very good addition to kindle. Cleans and scans. Very easy TO USE	⋆	⋆⋆⋆⋆⋆
Buy it and enjoy a great story.	⋆⋆⋆	⋆⋆⋆⋆⋆
Works great! I highly recommend it to everyone that enjoys singing hymns! Love it! Love it! Love it! :) .	⋆⋆⋆	⋆⋆⋆⋆⋆
Awesome it was better than all the other my weirder school books. I love it! The best book ever.Awesome	⋆	⋆⋆⋆⋆⋆
I gave this 5 stars under duress. I would rather give it 3 stars. it plays fine but it is a little boring so far.	⋆⋆⋆⋆⋆	⋆⋆⋆
only six words: don't waist your money on this	⋆⋆⋆⋆⋆	⋆
I love it so much at first I though it would be boring but turns out its fun for all ages get it	⋆	⋆⋆⋆⋆⋆
Excellent read, could not put it down! Keep up the great works ms. Brown. Cannot wait to download the next one.	⋆	⋆⋆⋆⋆⋆
This is one of the easiest to use games I have ever played. It is adaptable and fun. I love it.	⋆	⋆⋆⋆⋆⋆
So this is what today's music has become?	⋆	⋆⋆⋆⋆⋆
Sarah and Charlie, what a wonderful story. I loved this book and look forward to reading more of this series.	⋆⋆⋆	⋆⋆⋆⋆⋆
I've had this for over a year and it works very well. I am very happy with this purchase.	⋆	⋆⋆⋆⋆⋆
this show is insane and I love it. I will be ordering more seasons of it.	⋆⋆⋆	⋆⋆⋆⋆⋆
Just what the world needs, more generic r&b.	⋆	⋆⋆⋆⋆⋆
I did like the Making Of This Is movie it okay it not the best okay it not great .	⋆	⋆⋆⋆
Tough game. But of course it has the very best sound track ever!	⋆	⋆⋆⋆⋆⋆
unexpected kid on the way thanks to this shit	⋆	⋆⋆⋆⋆⋆
The kids are fascinated by it, Plus my wife loves it.. I love it I love it we love it	⋆⋆⋆	⋆⋆⋆⋆⋆
Loved this book! A great story and insight into the time period and life during those times. Highly recommend this book	⋆⋆⋆	⋆⋆⋆⋆⋆
Great reading I could not put it down. Highly recommend reading this book. You will not be disappointed. Must read.	⋆⋆⋆	⋆⋆⋆⋆⋆

Training settings To demonstrate that non-deep-learning methods can be effective in finding label issues under the CL framework, we use a multinomial logistic regression classifier for both finding label errors and learning with noisy labels. The built-in SGD optimizer in the open-sourced fastText library (Joulin et al., 2017) is used with settings: initial learning rate = 0.1, embedding dimension = 100, and n-gram = 3). Out-of-sample predicted probabilities are obtained via 5-fold cross-validation. For input during training, a review is represented as the mean of pre-trained, tri-gram, word-level fastText embeddings (Bojanowski et al., 2017).

Finding label issues Table 2.6 shows examples of label issues in the Amazon Reviews dataset found automatically using the CL: C+NR variant of confident learning. As an example, the first row of the table is labeled a 1-star review in the original dataset, but confident learning guesses it should be labeled a 5-star review. We observe qualitatively that most label issues identified by CL in this context are

Table 2.7: Ablation study (varying train set size, test split, and epochs) comparing test accuracy (%) of CL methods versus a standard training baseline for classifying noisy, real-world Amazon reviews text data as either 1-star, 3-stars, or 5-stars. A simple multinomial logistic regression classifier is used. Mean top-1 accuracy and standard deviations are reported over five trials. The number of estimated label errors CL methods removed prior to training is shown in the "Pruned" column. Baseline training begins to overfit to noise with additional epochs trained, whereas CL test accuracy continues to increase *(cf. N=1000K, Epochs: 50)*.

Test	Train set size	$N = 1000K$				$N = 500K$		
		Epochs: 5	Epochs: 20	Epochs: 50	Pruned	Epochs: 5	Epochs: 20	Pruned
10th	CL: $\boldsymbol{C}_{\text{confusion}}$	85.2±0.06	89.2±0.02	90.0±0.02	291K	86.6±0.03	86.6±0.03	259K
	CL: C+NR	86.3±0.04	**89.8±0.01**	**90.2±0.01**	250K	**87.5±0.05**	**87.5±0.03**	244K
	CL: $\boldsymbol{C}_{\tilde{y},y^*}$	**86.4±0.01**	**89.8±0.02**	90.1±0.02	246K	**87.5±0.02**	**87.5±0.02**	243K
	CL: PBC	86.2±0.03	89.7±0.01	**90.2±0.01**	257K	87.4±0.03	87.4±0.03	247K
	CL: PBNR	86.2±0.07	89.7±0.01	**90.2±0.01**	257K	87.4±0.05	87.4±0.05	247K
	Baseline	83.9±0.11	86.3±0.06	84.4±0.04	0K	82.7±0.07	82.8±0.07	0K
11th	CL: $\boldsymbol{C}_{\text{confusion}}$	85.3±0.05	89.3±0.01	90.0±0.0	294K	86.6±0.04	86.6±0.06	261K
	CL: C+NR	**86.4±0.06**	**89.8±0.01**	90.2±0.01	252K	**87.5±0.04**	**87.5±0.03**	247K
	CL: $\boldsymbol{C}_{\tilde{y},y^*}$	86.3±0.05	**89.8±0.01**	90.1±0.02	249K	**87.5±0.03**	**87.5±0.02**	246K
	CL: PBC	86.2±0.03	**89.8±0.01**	**90.3±0.0**	260K	87.4±0.03	87.4±0.05	250K
	CL: PBNR	86.2±0.06	**89.8±0.01**	90.2±0.02	260K	87.4±0.05	87.4±0.03	249K
	Baseline	83.9±0.0	86.3±0.05	84.4±0.12	0K	82.7±0.04	82.7±0.09	0K

reasonable except for sarcastic reviews, which appear to be poorly modeled by the bag-of-words approach.

Learning with noisy labels / weak supervision We compare the CL methods, which prune errors from the train set and subsequently provide clean data for training, versus a standard training baseline (denoted *Baseline* in Table 2.7), which trains on the original, uncleaned train dataset. The same training settings used to find label errors (see Subsection 2.5.3) are used to obtain all scores reported in Table 2.7 for all methods. For a fair comparison, all mean accuracies in Table 2.7 are reported on the same held-out test set, created by splitting the Amazon reviews dataset into a train set and test set such that every tenth example is placed in a test set and the remaining data is available for training (the Amazon Reviews 5-core dataset provides

no explicit train set and test set).

The Amazon Reviews dataset is naturally noisy, but the fraction of noise in the dataset is estimated to be less than 4% (Northcutt et al., 2021a), which makes studying the benefits of providing clean data for training challenging. To increase the percentage of noisy labels without adding synthetic noise, we subsample 1 million training examples from the train set by combining the label issues identified by all five CL methods from the original training data (244K examples) and a uniformly random subsample (766k examples) of the remaining "cleaner" training data. This process increases the percentage of label noise to 24% (estimated) in the train set and, importantly, does *not* increase the percentage of noisy labels in the test set – large amounts of test set label noise have been shown to severely impact benchmark rankings (Northcutt et al., 2021a).

To mitigate the bias induced by the choice of train set size, test set split, and the number of epochs trained, we conduct an ablation study shown in Table 2.7. For the train set size, we repeat each experiment with train set sizes of 1-million examples and $500,000$ examples. For the test set split, we repeat all experiments by removing every *eleventh* example (instead of tenth) in our train/test split (c.f. the first column in Table 2.7), minimizing the overlap (9%) between the two test sets. For each number of epochs trained, we repeat each experiment with 5, 20, and 50 epochs. We omit ($N = 500K$, Epochs: 50) because no learning occurs after 5 epochs.

Every score reported in Table 2.7 is the mean and standard deviation of five trials: each trial varies the randomly selected subset of training data and the initial weights of the logistic regression model used for training.

The results in Table 2.7 reveal three notable observations. First, all CL methods outperform the baseline method by a significant margin in all cases. Second, CL methods outperform the baseline method even with nearly half of the training data

pruned (Table 2.7, cf. N=500K). Finally, for the train set size $N = 1000K$, baseline training begins to overfit to noise with additional epochs trained, whereas CL test accuracy continues to increase *(cf. N=1000K, Epochs: 50)*, suggesting CL robustness to overfitting to noise during training. The results in Table 2.7 suggest CL's efficacy for noisy supervision with logistic regression in the context of text data.

2.5.4 Real-world Label Errors in Other Datasets

We use CL to find label errors in the purported "error-free" MNIST dataset consisting of preprocessed black-and-white handwritten digits, and also in the noisy-labeled WebVision dataset (Li et al., 2017a) consisting of color images collected from online image repositories where the search query is used as the noisy label.

Figure 2-7: Label errors in the original, unperturbed MNIST train dataset identified using CL: PBNR. These are the top 24 errors found by CL, ordered left-right, top-down by increasing self-confidence, denoted *conf* in teal. The predicted $\arg\max \hat{p}(\tilde{y} = k; x, \boldsymbol{\theta})$ label is in green. Overt errors are in red. This dataset is assumed "error-free" in tens of thousands of studies.

To our surprise, the original, unperturbed MNIST dataset, which is predominately assumed error-free, contains blatant label errors, highlighted by the red boxes in

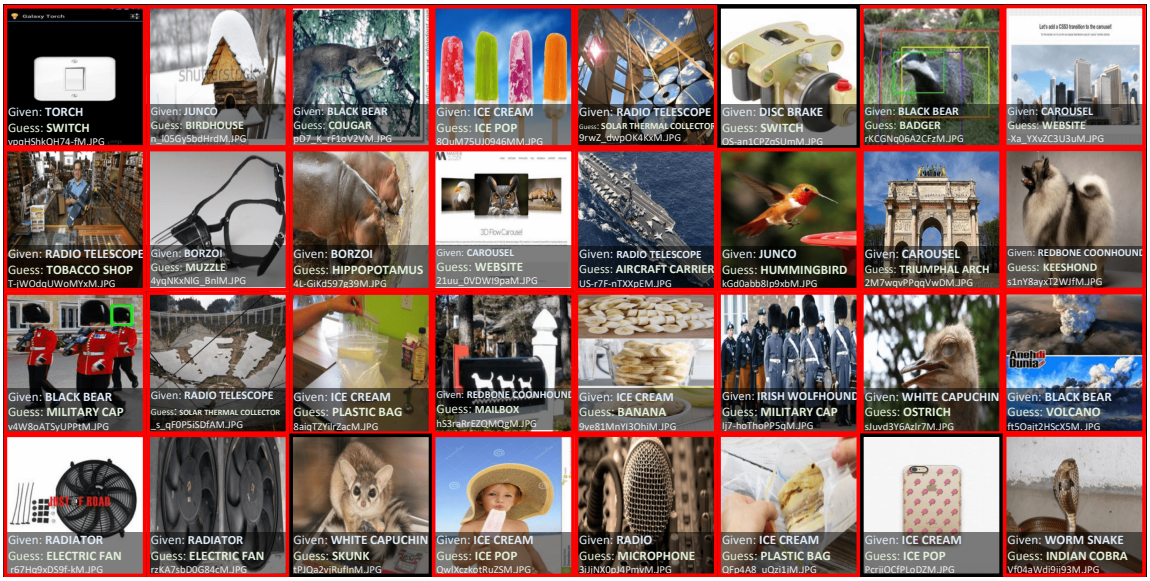


Figure 2-8: Top 32 identified label issues in the WebVision train set using CL: $\boldsymbol{C}_{\tilde{y},y^*}$. Out-of-sample predicted probabilities are obtained using a model pre-trained on ImageNet, avoiding training entirely. Errors are boxed in red. Ambiguous cases or mistakes are boxed in black. Label errors are ordered automatically by normalized margin.

Fig. 2-7. To find label errors in MNIST, we pre-trained a simple 2-layer CNN for 50 epochs, then used cross-validation to obtain $\hat{\boldsymbol{P}}_{k,i}$, the out-of-sample predicted probabilities for the train set. CL: PBNR was used to identify the errors. The top 24 label errors, ordered by self-confidence, are shown in Fig. 2-7. For verification, the indices of the train label errors are shown in grey.

To find label errors in WebVision, we used a pre-trained model to obtain $\hat{\boldsymbol{P}}_{k,i}$, observing two practical advantages of CL: (1) a pre-trained model can be used to obtain $\hat{\boldsymbol{P}}_{k,i}$ out-of-sample instead of cross-validation, and (2) this makes CL fast. For example, finding label errors in WebVision, with over a million images and 1,000 classes, took three minutes on a laptop using a pre-trained ResNext model that had never seen the noisy WebVision train set before. We used the CL: $\boldsymbol{C}_{\tilde{y},y^*}$ method

to find the label errors and ordered errors by normalized margins. Examples of WebVision label errors found by CL are shown in Fig. 2-8.

2.5.5 Failure Modes of Confident Learning

Confident learning can fail to exactly estimate $\boldsymbol{X}_{\tilde{y}=i,y^*=j}$ (and therefore $\boldsymbol{Q}_{\tilde{y},y^*}$) if the conditions in Thm. 2 are not met. This occurs for some example $\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ when either:

- $\hat{p}(\tilde{y}{=}j; \boldsymbol{x}, \boldsymbol{\theta}) < t_j \longrightarrow \boldsymbol{x} \notin \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$, or
- $\hat{p}(\tilde{y}{=}k; \boldsymbol{x}, \boldsymbol{\theta}) \geq t_k \longrightarrow \boldsymbol{x} \in \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=k}$, for some $k \neq j$

We can rewrite these two cases in terms of the per-example diffracted condition using our abbreviated notation, $\hat{p}_{\boldsymbol{x},\tilde{y}=j} = p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j}$ such that $\hat{p}(\tilde{y}{=}j; \boldsymbol{x}, \boldsymbol{\theta}) < t_j$ becomes $p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} < t_j \longrightarrow \epsilon_{\boldsymbol{x},\tilde{y}=j} < t_j - p^*_{\boldsymbol{x},\tilde{y}=j}$. Expressing the two failure cases in terms of error, we have:

- $\epsilon_{\boldsymbol{x},\tilde{y}=j} < t_j - p^*_{\boldsymbol{x},\tilde{y}=j} \longrightarrow \boldsymbol{x} \notin \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$, or
- $\epsilon_{\boldsymbol{x},\tilde{y}=k} \geq t_k - p^*_{\boldsymbol{x},\tilde{y}=k} \longrightarrow \boldsymbol{x} \in \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=k}$, for some $k \neq j$

When either case occurs, $\exists (i,j) \in [m]{\times}[m]$, s.t. $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \neq \boldsymbol{X}_{\tilde{y}=i,y^*=j}$.

Figure 2-9 shows examples from various datasets in `https://labelerrors.com` (discussed in Chapter 3) where CL potentially finds a label error incorrectly. Each example presents unique challenges. The sewing machine in Subfigure 2-9(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 2-9(b) is from the perspective of the pilot, looking out of the front cockpit window. In each example,

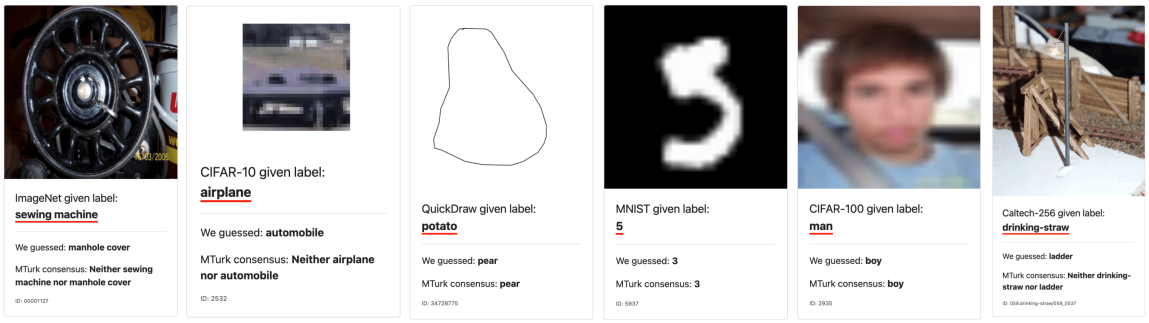


Figure 2-9: Difficult examples from various datasets in https://labelerrors.com (discussed in Chapter 3) where confident learning potentially finds a label error incorrectly. Example (a) is a cropped image of part of an antiquated sewing machine; (b) is a viewpoint from inside an airplane, looking out 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.

the error (deviation from the ideal probability) of the predicted probability for the example and class exceeds the threshold margin allowed for by the per-example diffracted condition used in Thm. 2.

2.6 Related Work

We first discuss prior work on confident learning and then review how CL relates to noise estimation and robust learning.

Confident learning Our results build on a large body of work termed "confident learning". Elkan (2001) and Forman (2005) pioneered counting approaches to estimate false positive and false negative rates for binary classification. We extend counting principles to the multi-class setting. To increase robustness against epistemic error

in predicted probabilities and class imbalance, Elkan and Noto (2008) introduced thresholding, but their approach required uncorrupted positive labels. CL generalizes the use of thresholds to multi-class noisy labels. CL also reweights the loss during training to adjust priors for the data removed. This choice builds on formative works (Natarajan et al., 2013; Van Rooyen et al., 2015) which used loss reweighting to prove equivalent empirical risk minimization for learning with noisy labels. More recently, Han et al. (2019) proposed an empirical deep self-supervised learning approach to avoid loss reweighting by using embedding layers of a neural network. In comparison, CL is non-iterative and theoretically grounded. Lipton et al. (2018) estimate label noise using approaches based on confusion matrices and cross-validation. However, unlike CL, the former assumes a less general label shift in the prior of noisy labels instead of class-conditional noise. Huang et al. (2019a) demonstrate the empirical efficacy of first finding label errors, then training on clean data, but the study evaluates only uniform (symmetric) and pair label noise – CL augments these empirical findings with theoretical justification for the broader class of asymmetric and class-conditional label noise.

Theory: a model-free, data-free approach Theoretical analysis with noisy labels often assumes a restricted class of models or data to disambiguate model noise from label noise. For example, Shen and Sanghavi (2019) provide theoretical guarantees for learning with noisy labels in a more general setting than CL that includes adversarial examples and noisy data, but they limit their findings to generalized linear models. CL theory is model and dataset agnostic, instead restricting the magnitude of example-level noise. In a formative related approach, Xu et al. (2019) prove that using the loss function $-\log\left(|\det(\boldsymbol{Q}_{\tilde{y},y^*})|\right)$ enables noise-robust training for any model and dataset, further justified by performant

empirical results. Similar to confident learning, their approach hinges on the use of $\boldsymbol{Q}_{\tilde{y},y^*}$; however, they require that $\boldsymbol{Q}_{\tilde{y}|y^*}$ is invertible and estimate $\boldsymbol{Q}_{\tilde{y},y^*}$ using $\boldsymbol{C}_{\text{confusion}}$, which is sensitive to class-imbalance and heterogeneous class probability distributions (see Sec. 2.2.1). In Sec. 2.3, we show sufficient conditions in Thm. 2 where $\boldsymbol{C}_{\tilde{y},y^*}$ exactly finds label errors, regardless of each class's probability distribution.

Uncertainty quantification and label noise estimation A number of formative works developed solutions to estimate noise rates using convergence criterion (Scott, 2015), positive-unlabeled learning (Elkan and Noto, 2008), and predicted probability ratios (Northcutt et al., 2017b), but are limited to binary classification. Others prove equivalent empirical risk for *binary* learning with noisy labels (Natarajan et al., 2013; Liu and Tao, 2015; Sugiyama et al., 2012) assuming noise rates are known, which is rarely true in practice. Unlike these binary approaches, CL estimates label uncertainty in the multiclass setting, where prior work often falls into five categories: (1) theoretical contributions (Katz-Samuels et al., 2019), (2) loss modification for label noise robustness (Patrini et al., 2016, 2017; Sukhbaatar et al., 2015; Van Rooyen et al., 2015), (3) deep learning and model-specific approaches (Sukhbaatar et al., 2015; Patrini et al., 2016; Jindal et al., 2016), (4) crowd-sourced labels via multiple workers (Zhang et al., 2017b; Dawid and Skene, 1979; Ratner et al., 2016), (5) factorization, distillation (Li et al., 2017b), and imputation (Amjad et al., 2018) methods, among other (Sáez et al., 2014). Unlike these approaches, CL provides a consistent estimator for exact estimation of the joint distribution of noisy and true labels directly, under practical conditions.

Label-noise robust learning Beyond the above noise estimation approaches, extensive studies have investigated training models on noisy datasets, e.g. (Beigman and Klebanov, 2009; Brodley and Friedl, 1999). Noise-robust learning is important for deep learning because modern neural networks trained on noisy labels generalize poorly on clean validation data (Zhang et al., 2017a). A notable recent trend in noise robust learning is benchmarking with symmetric label noise in which labels are uniformly flipped, (e.g., Goldberger and Ben-Reuven (2017); Arazo et al. (2019)). However, noise in real-world datasets is highly non-uniform and often sparse. For example, in ImageNet (Russakovsky et al., 2015), *missile* is likely to be mislabeled as *projectile*, but *missile* has a near-zero probability of being mislabeled as most other classes like *wool*, *ox*, or *wine*. To approximate real-world noise, an increasing number of studies examined asymmetric noise using. Examples include loss or label correction (Patrini et al., 2017; Reed et al., 2015; Goldberger and Ben-Reuven, 2017), per-example loss reweighting (Jiang et al., 2020a, 2018; Shu et al., 2019), Co-Teaching (Han et al., 2018), semi-supervised learning (Hendrycks et al., 2018; Li et al., 2017b; Vahdat, 2017), symmetric cross entropy (Wang et al., 2019), and semi-supervised learning (Li et al., 2020), among others. These approaches work by introducing novel new models or insightful modifications to the loss function during training. CL takes a loss-agnostic approach, instead focusing on generating clean data for training by directly estimating the joint distribution of noisy and true labels.

Comparison of the INCV Method and Confident Learning The INCV algorithm (Chen et al., 2019) and confident learning both estimate clean data, use cross-validation, and use aspects of confusion matrices to deal with label errors in ML workflows. Due to these similarities, we discuss four key differences between confident learning and INCV.

First, INCV errors are found using an iterative version of the $\boldsymbol{C}_{\text{confusion}}$ confident learning baseline: any example with a different given label than its argmax prediction is considered a label error. This approach, while effective (see Table 2.1), fails to properly count errors for class imbalance or when a model is more confident (larger or smaller probabilities on average) for certain classes than others, as discussed in Section 2.3. To account for this class-level bias in predicted probabilities and enable robustness, confident learning uses theoretically-supported (see Section 2.3) thresholds (Elkan, 2001; Richard and Lippmann, 1991) while estimating the confident joint. Second, a major contribution of CL is finding the label errors in the presumed error-free benchmarks such as ImageNet and MNIST, whereas INCV emphasizes empirical results for learning with noisy labels. Third, in each INCV training iteration, 2-fold cross-validation is performed. The iterative nature of INCV makes training slow (see Appendix Table A.1) and uses fewer data during training. Unlike INCV, confident learning is not iterative. In confident learning, cross-validated probabilities are computed only once beforehand, from which the joint distribution of noisy and true labels is directly estimated. This statistic is then used to identify clean data for a single re-training. We demonstrate that this approach is experimentally performant without iteration (see Table 2.1). Finally, confident learning is modular. CL approaches for training, finding label errors, and ordering label errors for removal are independent. In INCV, the procedure is iterative, and all three steps are tied together in a single looping process. A single iteration of INCV equates to the $\boldsymbol{C}_{\text{confusion}}$ baseline benchmarked in this chapter.

2.7 Future Work

Confident learning is a sub-field of machine learning, where the nature of learning a classifier resembles supervised learning, but the nature of uncertainty quantification of unknown true labels looks more like unsupervised, semi-supervised, and self-supervised approaches. Because confident learning intersects these fields, it opens several directions for future work, including but not limited to: assimilation of CL label error finding with pseudo-labeling and/or curriculum learning to *dynamically* provide clean data during training; learning with noise instead of removing noise, i.e., instead of exact prediction, allow for "close enough" prediction based on inherent ontological overlap in classes (e.g., predicting missile instead of projectile is "close enough"); and further exploration of iterative and/or regression-based extensions of CL methods. Some more direct future directions that extend the results presented in this chapter include: validation of CL methods on more datasets, such as the OpenML Benchmark (Feurer et al., 2019); the multi-modal Egocentric Communications (EgoCom) benchmark (Northcutt et al., 2020); and the realistic noisy label benchmark CNWL (Jiang et al., 2020a). Other future directions include evaluation of CL methods using other non-neural network models, such as random forests and XGBoost; examination of other threshold function formulations; and examination of label errors in test sets and they affect machine learning benchmarks at scale (see Chapter 3).

2.8 Chapter Contributions

Following the principles of confident learning, we developed a novel approach to estimate the joint distribution of noisy labels and true labels and explicated theoretical

and experimental insights into the benefits of doing so. We demonstrated accurate uncertainty quantification in high noise and sparsity regimes across multiple datasets, data modalities, and model architectures. We empirically evaluated three criteria: (1) uncertainty quantification via estimation of the joint distribution of noisy labels and true labels, (2) finding label errors, and (3) learning with noisy labels on CIFAR-10. We found that CL methods outperform recent prior art across all three.

These findings emphasize the practical nature of confident learning, identifying numerous pre-existing label issues in ImageNet, Amazon Reviews, MNIST, and other datasets, and improving the performance of learning models like deep neural networks by training on cleaned datasets. Confident learning motivates the need for further understanding of dataset uncertainty estimation, methods to clean training and test sets, and approaches to identify ontological and label issues for dataset curation.

This thesis makes two key contributions to prior work on finding, understanding, and learning with noisy labels. First, a proof is presented giving realistic sufficient conditions under which CL exactly finds label errors and exactly estimates the joint distribution of noisy and true labels. Second, experimental results suggest that confident learning is empirically performant, outperforming seven recent highly competitive methods for learning with noisy labels on the CIFAR dataset. The results presented are reproducible with the implementation of CL algorithms, open-sourced as the `cleanlab`[2] Python package.

These contributions are presented beginning with the formal problem specification and notation (Section 1.4), then defining the algorithmic methods employed for CL (Section 2.2), and theoretically bounding expected behavior under ideal and

[2]To foster future research in data cleaning and learning with noisy labels and to improve accessibility for newcomers, `cleanlab` is open-source and well-documented: https://github.com/cgnorthcutt/cleanlab/

noisy conditions (Section 2.3). Experimental benchmarks on the CIFAR, ImageNet, WebVision, and MNIST datasets, cross-comparing CL performance with that from a wide range of highly competitive approaches, including *INCV* (Chen et al., 2019), *Mixup* (Zhang et al., 2018), *MentorNet* (Jiang et al., 2018), and *Co-Teaching* (Han et al., 2018), are then presented in Section 2.5. Related work (Section 2.6) and concluding observations (Section 2.8) wrap up the presentation. Extended proofs of the main theorems, algorithm details, and comprehensive performance comparison data are presented in the appendices.

The contributions of this chapter include:

1. Proved realistic sufficient conditions under which CL exactly finds label errors and exactly estimates the joint distribution of noisy and true labels.

2. Verified the generality and efficacy of CL in several commonly-used machine learning datasets. We show CL performance exceeds seven recent competitive approaches for learning with noisy labels on the CIFAR dataset, find several label errors in the presumed error-free MNIST dataset, and improve sentiment classification on text data in Amazon Reviews. We also employ CL on ImageNet to quantify ontological class overlap (e.g., estimating 645 *missile* images are mislabeled as their parent class *projectile*) and moderately increase model accuracy (e.g., for ResNet) by cleaning data prior to training.

3. Released the `cleanlab` as a standard package (supporting Posix/Linux, Windows, and MacOS/Unix systems) for machine learning with noisy labels. `cleanlab` provides a standard platform for seasoned researchers in data-centric machine learning with noisy labels and is well-documented to promote accessibility for new researchers. All results in this chapter are reproducible with `cleanlab`.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Errors in Test Sets Destabilize Machine Learning Benchmarks

> *"What I'm finding is that for a lot of problems, it's useful to shift our mindset toward not just improving the code, but in a more systematic way, improving the data."*
>
> - Andrew Ng (2021)

In this chapter, 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: in Section 3.3, 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, in Section 3.4, via crowdsourcing (54% of the algorithmically-flagged candidates are indeed erroneously labeled). Traditionally,

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

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, in Section 3.5 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%.

Attribution This chapter includes material previously published as (Northcutt et al., 2021a). Anish Athalye and Jonas Mueller contributed significantly to the material presented in this chapter. This work was supported in part by funding from the MIT-IBM Watson AI Lab.

Acknowledgements Aspects of the contents of this chapter were shaped by input from Romain Futrzynski, who assisted with notation and feedback; Jessy Lin, who contributed significantly to aspects of an earlier version of this work (e.g., finding errors in Caltech-256), and Lu Jiang and Isaac Chuang, who contributed to the underlying framework, confident learning, used to identify the label errors.

3.1 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 (Sambasivan et al., 2021). Even with controls for error correction (Kremer et al., 2018; Zhang et al., 2017b), 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., 2019b; 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., 2021b; 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. 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 and Carneiro, 2020), this chapter 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., 2021b), 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 3-1 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 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.

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

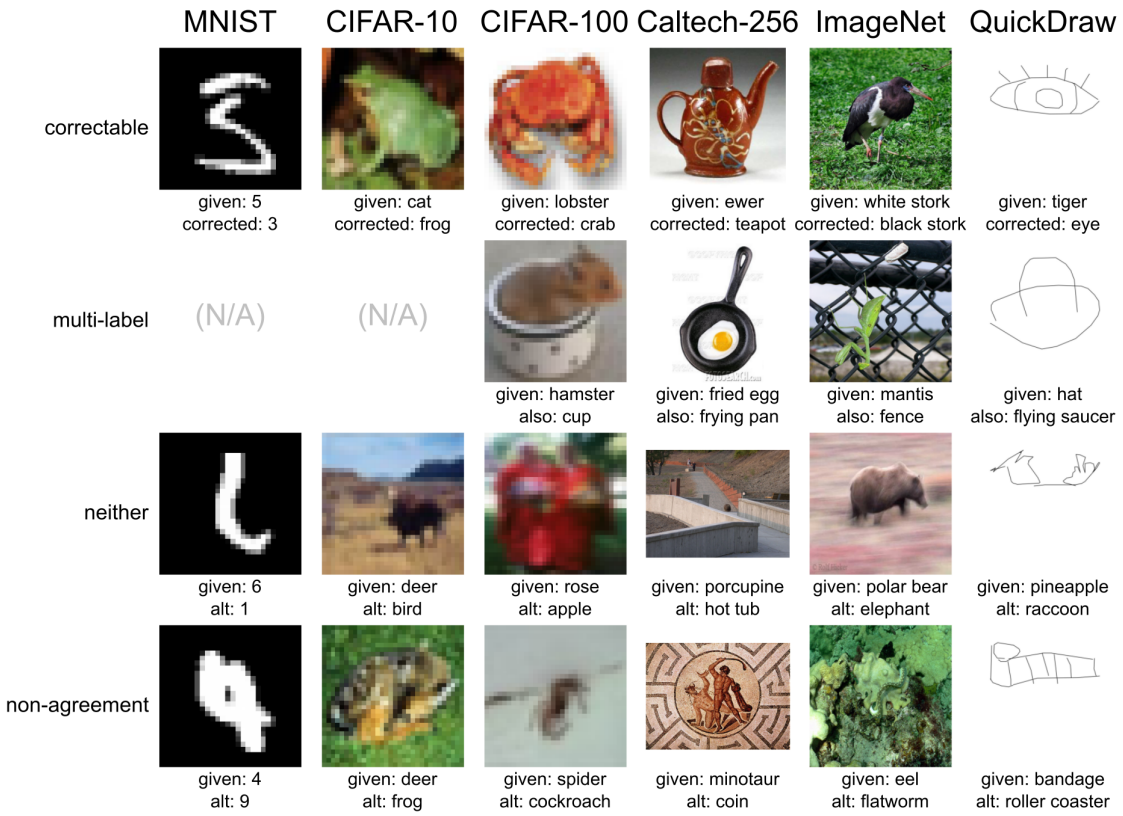


Figure 3-1: An example label error from each category (Sec. 3.4) for image datasets. The figure shows given labels, human-validated corrected labels, also the second label for multi-class data points, and CL-guessed alternatives. A browser for all label errors across all 10 datasets is available at `https://labelerrors.com`. Errors from text and audio datasets are also included on the website.

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 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.

3.2 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 (List of Datasets for Machine Learning Research, 2018), 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.

3.2.1 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. Details, e.g., the number of examples in each dataset, are listed in Table 3.1.

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 and Hinton, 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 and 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."

3.3 Identifying Label Errors

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., 2021b), 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., 2021b) 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 and Laird, 1988), depending only on the latent true class, not the data. This assumption is commonly used (Goldberger

and 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 3.1 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 described in detail in Chapter 2.

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 3.1.

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

Table 3.1: 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.

3.4 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 3-2 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

Table 3.2: 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	-	-	-	-

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 3.2 summarizes the results, and Figure 3-1 shows examples of validated label errors from image datasets.

3.5 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.

Definition 3 (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.*

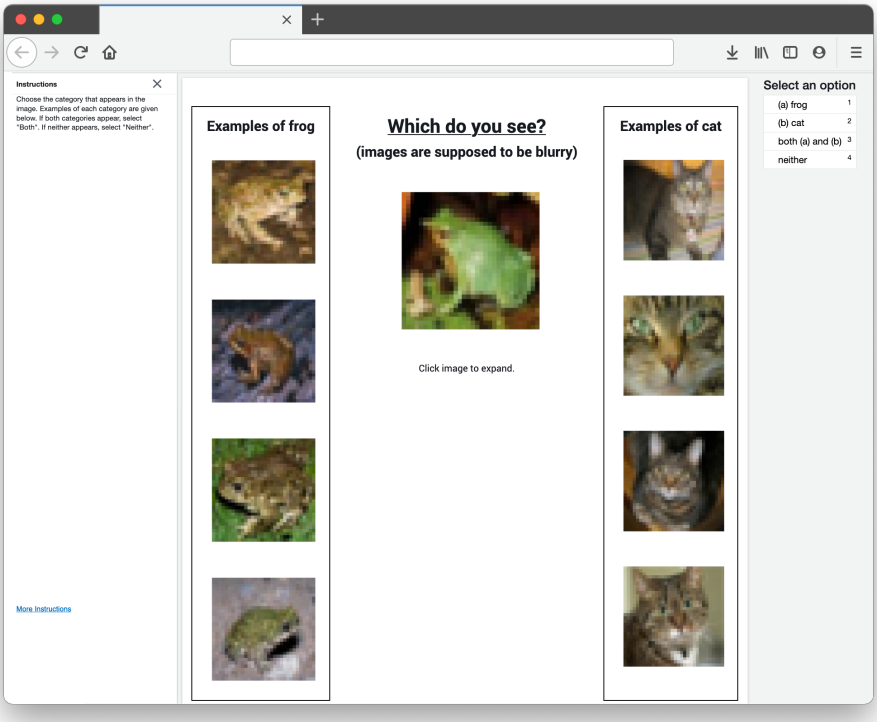


Figure 3-2: Mechanical Turk worker interface showing an example from CIFAR-10 (with given label "cat"). For each data point algorithmically identified as a potential label error, the interface presents the data point, along with examples belonging to the given class. The interface also shows data points 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 4 (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, "\" 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 5 (unknown-label set, $\mathcal{U}$)**.** *The subset of CL-flagged test examples for which human labelers could not agree on a correct label ($\mathcal{U} \subset \mathcal{D} \backslash \mathcal{B}$). This includes examples where human reviewers agreed that multiple classes or none of the classes are appropriate.*

Definition 6 (pruned set, $\mathcal{P}$)**.** *The remaining test data after removing $\mathcal{U}$ from $\mathcal{D}$ ($\mathcal{P} = \mathcal{D} \backslash \mathcal{U}$).*

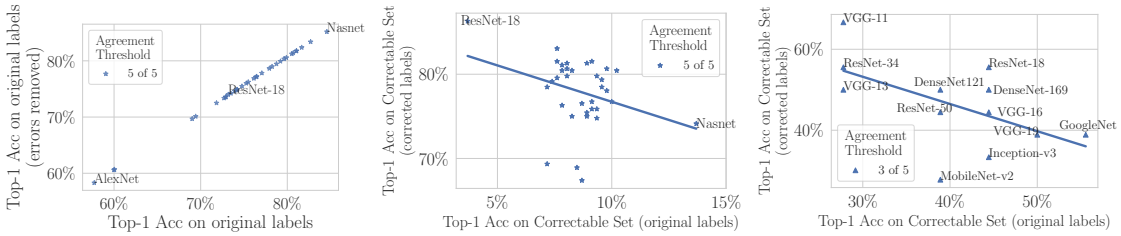
Definition 7 (correctable set, $\mathcal{C}$)**.** *The subset of CL-flagged examples for which human-validation reached consensus on a different label than the originally given label ($\mathcal{C} = \mathcal{P} \backslash \mathcal{B}$).*

Definition 8 (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 set label errors? Figure 3-3 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 3-3a 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 B.1 in the Appendices). We verified that the same trend occurs independently across 13 models pre-trained on CIFAR-10 (Figure 3-3c), e.g. VGG-11 significantly



(a) ImageNet, Validation set (b) ImageNet, Correctable set (c) CIFAR-10, Correctable set

Figure 3-3: Benchmark ranking comparison of 34 models pre-trained on ImageNet and 13 pre-trained on CIFAR-10 (more details in Tables B.2 and B.1 and Fig. B-1, 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: $34/34 \rightarrow 1/34$.

outperforms VGG-19 (Simonyan and 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 3-3b and 3-3c, 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

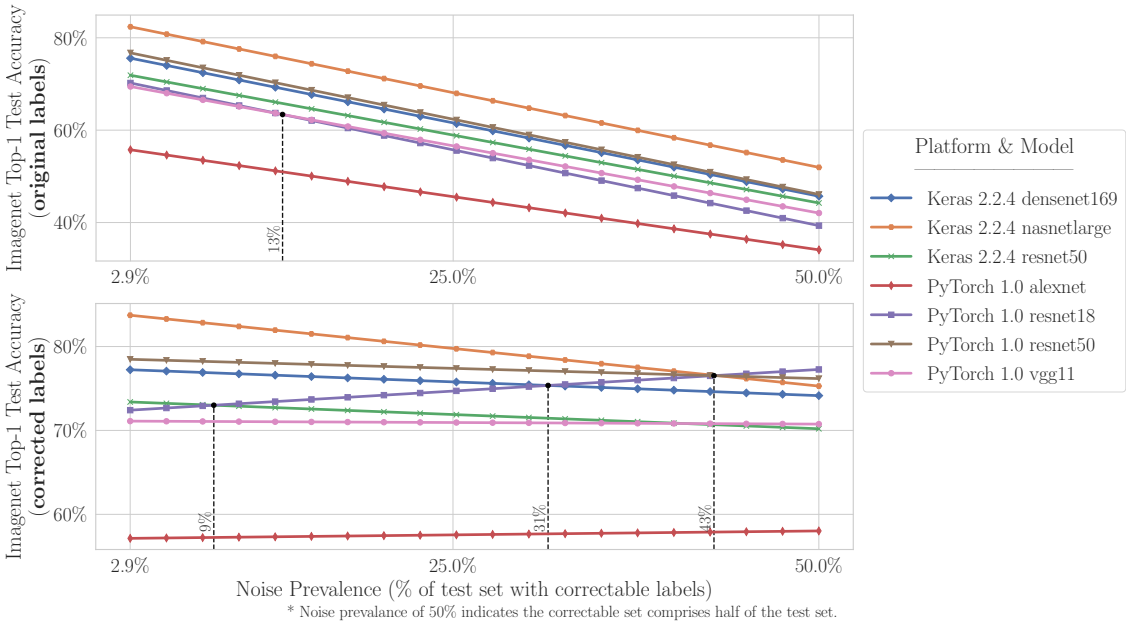


Figure 3-4: ImageNet top-1 original accuracy (top panel) and corrected accuracy (bottom panel) vs Noise Prevalence (agreement threshold = 3). Vertical lines indicate noise levels at which the ranking of two models changes (in terms of original/corrected accuracy). The left-most point ($N = 2.9\%$) on 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 3.2).

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

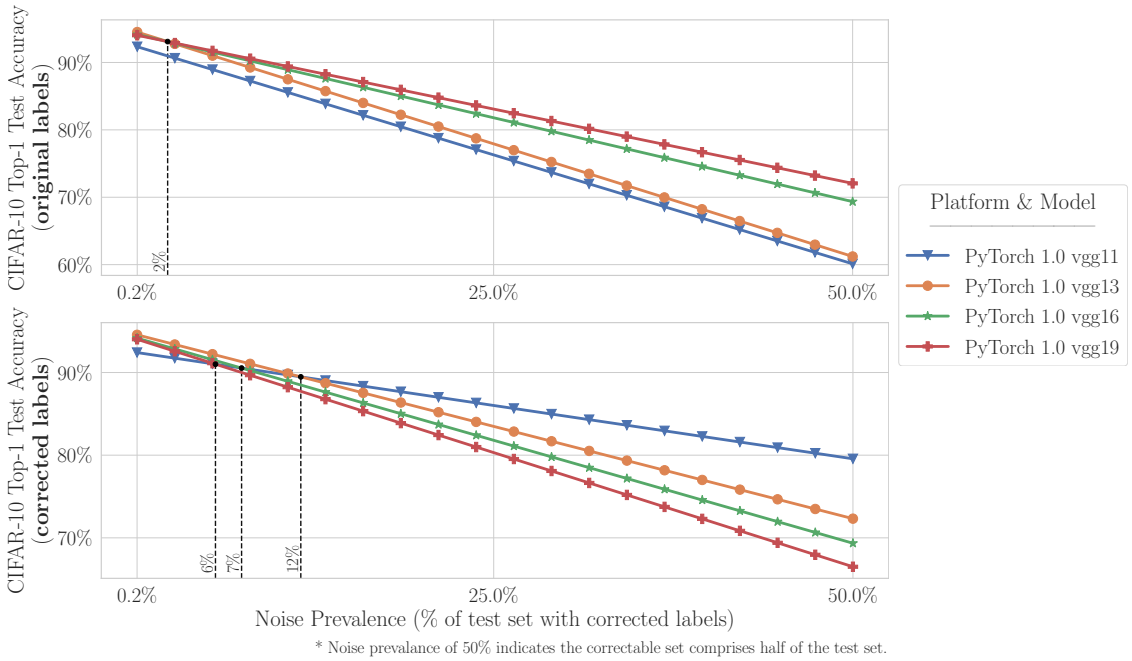


Figure 3-5: CIFAR-10 top-1 original accuracy (top panel) and corrected accuracy (bottom panel) vs Noise Prevalence (agreement threshold = 3). For additional details, see the caption of Fig. 3-4.

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 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 3-4-3-5. 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 3-4-3-5 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.

3.6 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., 2020b), or, a naturally noisy dataset is used and accuracy

is measured on a noisy test set. In the noisy WebVision dataset (Li et al., 2017a), accuracy on the ImageNet validation is often reported as a "clean" test set, however, related works (Recht et al., 2019; Northcutt et al., 2021b; 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 prior work deals with label errors in the *training* set (Frénay and Verleysen, 2014; Cordeiro and 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., 2017b; Dawid and 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., 2021b), to estimate which labels are erroneous across diverse datasets. Northcutt et al. (2021b) 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).

3.7 Future Work

This chapter 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.

3.8 Chapter Contributions

Traditionally, ML practitioners choose which model to deploy based on test accuracy — the findings in this chapter 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.

The contributions of this chapter 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).

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

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.

Part II

Confident Learning for Humans

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

From Machines to Humans

> *Artificial intelligence is the study of ideas which enable computers to do the things that make people seem intelligent.*
>
> - Patrick Winston (1977)

In Chapter 2, we introduced *confident learning*, whereby a machine, like humans, must learn with *noisy-labeled data*, directly quantify and identify label noise, and *unlearn* misconceptions by re-learning with confidence on cleaned data. To achieve this, we developed a systematic theory and framework for *confident learning* with affordances for quantifying, identifying, and learning with label errors in data. Chapter 2 culminated with example applications of confident learning on datasets like ImageNet, MNIST, Amazon Reviews, CIFAR-10, and WebVision, whereby CL helped machine models learn with confidence by providing clean data during *training*. In Chapter 3, we saw how CL helps to benchmark machine models with confidence by providing clean test data during *evaluation*, concluding our exposition on confident learning for machines.

As we transition now to confident learning for humans, let us take a moment

to reflect on two examples where the results in Chapters 2 and 3, which primarily are intended to support confident learning for machines, also provide affordances for confident learning for humans.

In Subsection 2.5.2 in Chapter 2, we studied how confident learning can be used to assist practitioners in dataset curation by automatically identifying issues in class ontologies. For the time being, dataset curation largely remains a human-performed task, where the class labels are chosen by humans for some downstream application. This application of confident learning directly augments human capabilities in dataset curation and enables humans to more confidently curate datasets with consistent ontologies (less overlap among classes).

In Section 3.5 in Chapter 3, we studied how confident learning can be used to assist machine learning practitioners looking to deploy, in the real world, a model trained with noisily-labeled data (e.g., self-driving vehicles). Using confident learning to clean the test sets that these models are benchmarked on, we provide a procedure to directly estimate the fraction of noise in a test set that can lead to benchmark ranking instability. Using confident learning, an ML practioner can benchmark real-world performance based on a cleaned test set (versus the performance on a test set comprising pervasive label errors which does not accurately reflect real-world performance, even though it may be drawn identically from the same distribution as the training data).

These examples shift our focus to Part II of this thesis: confident learning for humans. The next three chapters develop artificially intelligent systems that augment human capabilities in real-world, noisy settings. In Chapter 5, we humanize multi-modal conversational AI with the creation of the first multi-person egocentric conversations dataset, called EgoCom. Using EgoCom, we develop assisted-turn-taking in conversations, e.g. to warn humans just before their time to respond in a

conversation, by combining noisy embodied audio and video signals from multiple synchronized perspectives. Then, in Chapter 6, we build a system for assisted-generation of writing song lyrics by exploiting the inherent aleatoric uncertainty of language and semantics. Finally, in Chapter 7, we assist human learning in open online courses by depolarizing/diversifying comment rankings to mitigate the majority bias inherent in rankings based on upvotes. In each chapter, the artificially intelligent system's ability to overcome uncertainty is linked to its efficacy of augmenting human capabilities, and, by extension, humans confidence in their ability to perform the associated task.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

EgoCom: Multi-human, Multi-modal Egocentric Communications

"When people say you can't, or it's impossible, or it's never gonna happen, remember that they are telling their story, not yours."
- Arnold Schwarzenegger (2018)

Multi-modal datasets in artificial intelligence (AI) often capture a third-person perspective, but our *embodied* human intelligence evolved with sensory input from the egocentric, first-person perspective. Towards embodied AI, in Section 5.2 of this chapter, we introduce the Egocentric Communications (EgoCom) dataset to advance the state-of-the-art in conversational AI, natural language, audio speech analysis, computer vision, and machine learning. EgoCom is a first-of-its-kind natural conversations dataset containing multi-modal human communication data captured simultaneously from the participants' egocentric perspectives. EgoCom includes 38.5 hours of synchronized embodied stereo audio, egocentric video with 240,000 ground-truth, time-stamped word-level transcriptions and speaker labels from 34

diverse speakers. We study baseline performance on two novel applications that benefit from embodied data: (1) predicting turn-taking in conversations (Section 5.3) and (2) multi-speaker transcription (Section 5.4). For (1), we investigate Bayesian baselines to predict turn-taking within 5% of human performance. For (2), we use simultaneous egocentric capture to combine Google speech-to-text outputs, improving global transcription by 79% relative to a single perspective. Both applications exploit EgoCom's synchronous multi-perspective data to augment performance of embodied AI tasks.

Attribution This chapter includes material previously published as (Northcutt et al., 2020). Cindy (Shengxin) Zha, Steven Lovegrove, and Richard Newcombe contributed significantly to the material presented in this chapter.

Acknowledgements A considerable portion of the results in this chapter were derived while I was an intern, and later, a visiting research scientist with Facebook Reality Labs, formerly known as Oculus Research. This work was, in part, funded by Facebook.

5.1 Introduction: The Need for Egocentricity

Consider a conversational *turn-taking* system to predict who will be speaking in five seconds or whether it's a good time to start, stop, or continue talking. This system is useful as an aid for persons dealing with autism (Dobbinson et al., 1998); teaching a personal assistant when to help (Wu et al., 2018); autonomous multi-agent collaboration (DeVault et al., 2015); or attention measurement in affective computing (El Kaliouby et al., 2006). The complexity of conversation makes predicting turn-

taking a challenging task. Here we use a simple baseline to show that multi-perspective egocentric data has compelling benefits.

Alternatively, consider a *global transcription* system for multi-person transcription and speaker identification. With a single audio source, one must solve the challenging cocktail party problem (Haykin and Chen, 2005) and with multiple audio input sources, one must solve an often misspecified matrix factorization (Ozerov and Févotte, 2010). The inclusion of visual features has aided in simplifying object classification and source separation (Ephrat et al., 2018; Gao et al., 2018; Arandjelovic and Zisserman, 2018), but global transcription requires simultaneous speaker disambiguation and multi-channel speech recognition (Ozerov and Févotte, 2010). Here we show how synchronous multi-perspective egocentric data enables a simple solution to improve a state-of-the-art speech-to-text system by 79%, when compared to asynchronous, single-perspective transcription.

We introduce the Egocentric Communications (EgoCom[1]) dataset, the first multi-perspective egocentric dataset comprised of natural human conversations, and establish baseline performances for both *turn-taking* and *global transcription*. The primary contribution of EgoCom is the unique nature of the data. EgoCom is a multi-modal, synchronous multi-perspective, egocentric communications dataset comprising 38 unique 20-30 minute natural conversations. Each conversation has three participants, with at least two wearing video recording glasses. Egocentric video is captured from the perspective of the eyes and embodied stereo audio is captured from the perspective of the ears. Transcriptions are provided via human annotators. For each conversation, start and end times of each participant's data is synchronized.

[1]The EgoCom Dataset is open-source and available for download at: https://github.com/facebookresearch/EgoCom-Dataset

Beyond turn-taking and transcription, EgoCom is timely and relevant as an egocentric communication benchmark dataset. The ubiquity of hand-held smart devices and head-worn recording devices (McNaney et al., 2014) has proliferated egocentric video, yet the usefulness of egocentric capture remains largely unrealized by the artificial intelligence community. While recent datasets like EPIC-KITCHENS (Damen et al., 2018) and GTEA Gaze+ (Li et al., 2018) have significantly advanced this goal, there is no public egocentric dataset addressing two key elements of embodied intelligence: natural language and multi-perspective interaction. EgoCom serves this purpose. We elect the term *communications*, as opposed to *conversations*, because the multi-modal nature of the dataset includes both verbal language and non-verbal cues (Stratou and Morency, 2017).

EgoCom captures language across three modalities: verbal, vocal, and visual (Stratou and Morency, 2017). EgoCom captures verbal cues through human-transcribed annotations, vocal cues through egocentric audio data, and visual cues through gestures, body language, and gaze. EgoCom amplifies egocentric audio relative to quieter surrounding audio thereby simplifying tasks like speaker identification (Reynolds, 2002): a simple solution is to use the maximum magnitude of aligned audio. Similarly, EgoCom enhances visual cues through the egocentric perspective by enabling spatial AI techniques like head-pose estimation combined with traditional computer vision techniques like body pose estimation (Murphy-Chutorian and Trivedi, 2009).

Our goal is not to discover state-of-the-art algorithms for turn-taking prediction or global transcription, but is instead to demonstrate how the synchronized multi-perspective, multi-modal nature of the EgoCom dataset simplifies solutions to otherwise challenging tasks, while establishing baseline scores for these tasks in the embodied AI context.

5.2 EgoCom Dataset

Egocentric Communications (EgoCom) is a first-of-its-kind natural conversations dataset containing multi-modal human communication data captured simultaneously and synchronized across participants' egocentric perspectives. For each conversation, the dataset provides embodied stereo audio, egocentric video, time-stamped word-level transcriptions, and speaker labels. EgoCom is comprised of 28 unique English conversations across 34 diverse speakers. Every conversation has three participants with at least two participants wearing a recording device. Low-cost head-worn "*Gogloo*" glasses were used to record stereo audio near the ears and 1080p video between the eyes (see Fig. 5-1 for an example of the device). Three synchronized egocentric video frames from each participant's perspective in a conversation are shown in Fig. 5-2.

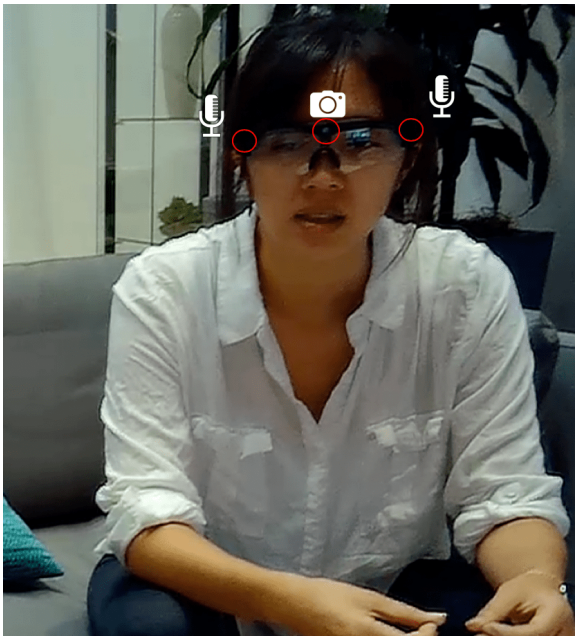


Figure 5-1: Screenshot taken from the EgoCom dataset, depicting the recording glasses used and locations of the video recording camera and stereo audio recording microphones.

The color of each image matches the perspective-arrow in the other images.

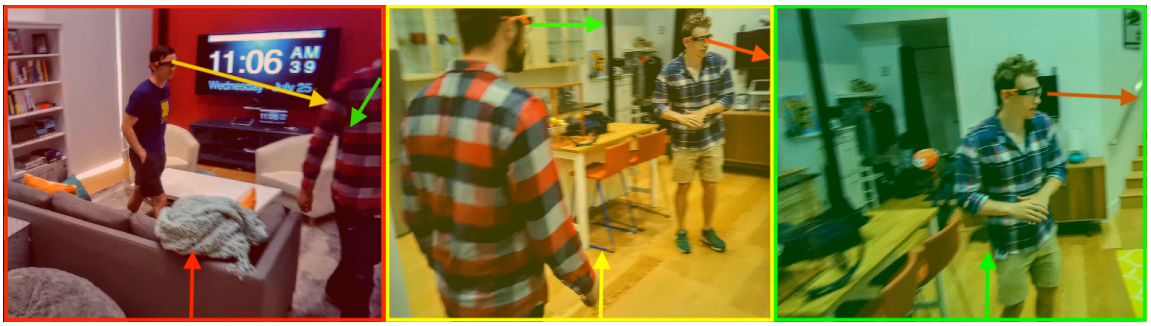


Figure 5-2: Synchronized egocentric videos from three people in a conversation. Arrows depict the egocentric visual perspective, with each unique color corresponding to one perspective.

Topics of Conversation Every conversation includes a host who directs topics. To enable future research, conversations adhere to topics: playing and teaching how to play card games; playing word-guessing games; pontificating thought experiments; discussing interests (e.g. favorite food); describing objects in the environment; question-answering, teaching, and learning about how things work; and interacting with mirrored reflections with egocentric video. Although topics are constrained, conversations are reasonably natural. Throughout the dataset, an estimated 7,200 unique words are spoken, with the most common word being "I" and an estimated 3000 unique words only spoken once.

The EgoCom dataset is split into a train set (78%), test set (16%), and validation set (6%) by total duration (see Fig. 5-3). These sets were generated randomly while enforcing similar distribution across gender and dialect. The term *non-native* is used to qualitatively express a non-American, non-British English accent.

Dataset Content Coverage EgoCom encompasses a breadth of typical conversational elements including variation in (1) spatial geometry variations such as position and movement while speaking, (2) relative speaker geometries including

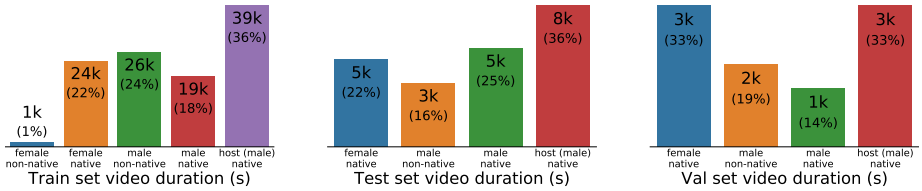


Figure 5-3: Distribution of train, test, and validation sets for the EgoCom dataset.

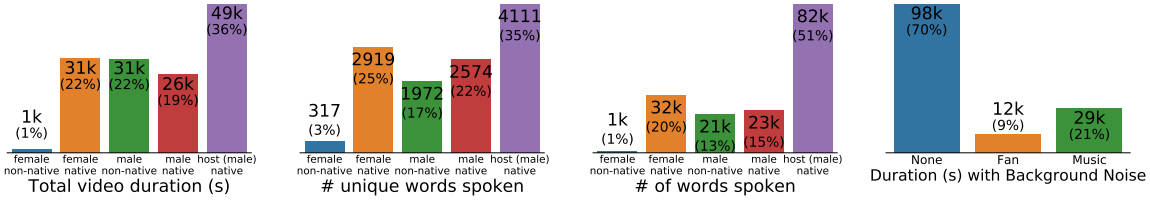


Figure 5-4: Gender, dialect, and background noise statistics for the EgoCom dataset.

variation in all six degrees of freedom (x, y, z, yaw, pitch, tilt), (3) environment variations including background fan noise and music varying in genre (classical, latin, country) and loudness, and (4) speaker demographics. Varying accents, dialects, and cultures are represented. All conversations are recorded in the same high-ceiling studio apartment. Fig. 5-4 quantifies distributional statistics across demographics and background noise. Observe that the largest bar depicts the host, who participated in every conversation. The train, test, and validation sets comprise 160,000 spoken words and 14 hours of unique conversational data, or 38.5 hours of video, audio, and text data from all perspectives in each conversation.

5.2.1 Contribution of EgoCom to Egocentric Datasets

EgoCom is the only multi-modal, synchronized multi-perspective egocentric conversational dataset as of the time of dataset publication. *EPICKITCHENS* (Damen et al., 2018) (50 hours) and EGTEA Gaze+ (Li et al., 2018) (28 hours) are two of the largest egocentric datasets, but are single-perspective. EgoCom comprises

38.5 hours of video and stereo audio, 240,000 ground-truth utterances, and speaker labels. EgoCom is unique and larger than any other egocentric dataset published with these properties (Li et al., 2015).

5.2.2 Multi-capture synchronization

Our solutions for turn-taking prediction and global transcription in the next sections hinge on the fact that videos are synchronized: within a conversation, all video/audio starts and stops at the same moments in time. Ideally, a synchronized global clock would be logged with the sensor data on each device to support this. Unfortunately, no commercial and unobtrusive wearable capture device supporting the required sensors exist with support for such a global clock, so we use a simple method to infer alignment from the captured data streams.

Algorithm 3 Audio Alignment for synchronization

input *wav*: array of n egocentric 2-channel audio wav arrays corresponding to n videos.
output *list*: alignment shifts for each *wav* input.
shifts = [0]
for $i \leftarrow 1, n-1$ **do** ▷ Shift computed relative to $wav[0]$
    shifts = []
    for $u, v \leftarrow$ all four combinations of left and right channel of $wav[0]$ and $wav[i]$ **do**
        z.append[shift of max $\frac{u \cdot v}{\|u\| \cdot \|v\|}$] ▷ cross-correlation
    shifts.append(median(z))
alignment = shifts – min(shifts)

To synchronize all perspectives in each conversation, videos are aligned based on their audio (see Alg. 3). Beforehand, audio is truncated to the minimum length of any perspective in each conversation. Volume is equalized within each signal using Gaussian smoothing, implemented by dividing each signal by itself convolved with a

Gaussian kernel of width of 0.1 seconds. After these preprocessing steps, alignment is performed using cross-correlation in Fourier space on all combinations of left and right channels of audio, detailed in Alg. 3.

Speaker Labels Speaker labels are obtained by aligning the raw audio for each participant in a given conversation (see Alg. 3). Audio magnitudes are computed by summing together the absolute values of both channels. One dimensional max-pooling with Gaussian smoothing is then used to find the speaker with maximum magnitude for every one second of audio, e.g. the label used in our experiments at z seconds in the future is the max amplitude signal averaged from z seconds to $z + 1$ seconds. If no speaker exceeds a threshold (10th-percentile of all magnitudes), a zero label is used to represent *no one is speaking*. Our labeling procedure assumes, sometimes incorrectly, that only one person is speaking at any given one second window.

5.2.3 Details about the Creation of the EgoCom Dataset

To support distribution of the EgoCom dataset for researchers with low bandwidth, for half of the dataset (87 videos), conversations are broken up into 5 minute clips. The rest of the dataset (88 videos) comprise conversations between 15 and 30 minutes. Additionally, video files were compressed with ffmpeg to support the following video sizes:

- 1080p (1920x1080) RAW, uncompressed
- 1080p (1920x1080) compressed
- 720p (1280x720) compressed
- 480p (640x480) compressed

- 240p (352x240) compressed

As an example, we used the following compressing method (crf 24 was used for 1080p):
`ffmpeg -i input.mp4 -s 1280x720 -aspect 1280:720 -vcodec libx264 -crf 20 -threads 12 compressed.mp4`.

For the video feature representations, 480p compression was used.

5.3 Application: Predicting Turn-Taking in Conversations

In this section, we study the application of predicting turn-taking in conversation and demonstrate the advantages of synchronous multi-perspective data afforded by EgoCom. We first study priors (e.g. the transition probabilities between speakers), then likelihood and posterior models to predict future speaker labels from past data and labels. Posterior inference is formulated by including the current speaker label, at time $t = 0$, as an additional feature while training to predict a future speaker label. In this formulation, the priors are useful baselines, e.g. if the prior probability that someone's speaking state will not change in t seconds is 0.64, then a trivial model that predicts the current label, will tend towards 64% accuracy.

This posterior has more information and should perform at least as well as likelihood estimation, so why study both? As discussed in Sec 5.2.2, in contrast to other datasets, EgoCom 's multi-perspective data provides reliable current speaker labels for training but these would also potentially be available for a distributed run-time inference system. This makes posterior estimation at inference time possible. For this reason, we are interested in studying the value of this extra multi-perspective

information in the prediction of turn-taking. We approach turn-taking prediction with four tasks:

Binary Prediction:

1. **Task 1**: given *any one* person's features, will *that person* be speaking in t seconds?

2. **Task 2**: given *only* the conversation host's features, will *the host* be speaking in t seconds?

3. **Task 3**: given a concatenation of *all* participant's features, will *the host* be speaking in t seconds?

Multiclass Prediction:

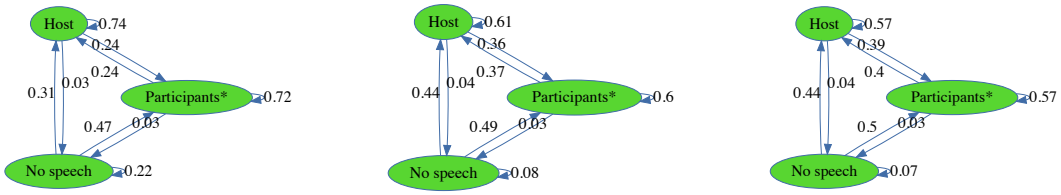
4. **Task 4**: given a concatenation of *all* participant's features, who will be speaking in t seconds?

Our goal is to answer these questions as a real-time prediction task using multimodal multi-perspective embodied communication data. These tasks are constructed to disambiguate latent factors, such as the influence of the host in conversations (**Task 1** versus **Task 2**), the value of EgoCom's unique synchronous multi-perspective data (**Task 2** versus **Task 3**), and how predicting change in speaker label (binary **Task 1**) compares to the more difficult task of predicting *who* will be speaking (**Task 4**).

Toward this goal, we favor an approach with fast inference/prediction time by first pre-computing feature representations for visual, audio, and text data using models that are already pre-trained on related datasets. These features are computed for histories of 4, 5, 10, and 30 seconds at 1 second increments. In a second step, we train

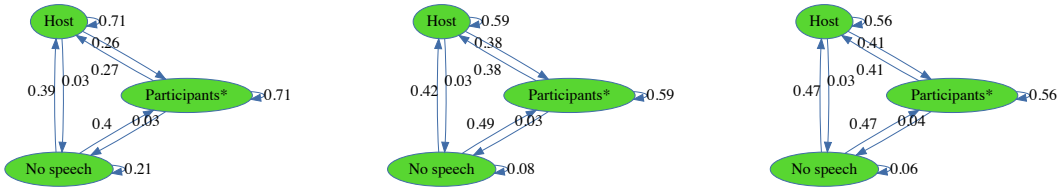
a simple MLP classifier using the pre-computed features. We chose this approach over an end-to-end recurrent model for inference-time speed and training stability (Collobert and Weston, 2008).

We conclude this section with an ablation study and a comparative human performance evaluation.



(a) 1 second in future (b) 5 seconds in future (c) 10 seconds in future

Figure 5-5: Probability of turn-taking between host and any participants in the EgoCom **train** dataset. **Participants includes all (usually two) participants, e.g. 72% is the probability* any *participant will be speaking in 1s given* any *participant is currently speaking.*



(a) 1 second in future (b) 5 seconds in future (c) 10 seconds in future

Figure 5-6: Probability of turn-taking transitions between host and any participants in the EgoCom **test** dataset.

5.3.1 Prior Probabilities on Speaker Labels

The current speaker label is an indicator of who will be speaking in the future. Before training models to predict speaker labels from data, we estimate the prior probabilities of speaker labels directly by counting the labels of the training set. Define s_t to be a

binary random variable, such that $s_t = True$ if *any* embodied person is speaking t seconds in the future and $False$ otherwise. Define h_t similarly for the conversation host and define $m_t \in \{0, 1, 2, 3\}$ as the multiclass label of the person speaking at time t ($m_t = 0$ if no one is speaking at time t). The prior baselines $p(s_t = s_0)$, $p(h_t = h_0)$ and $p(m_t = m_0)$ measure the probability speaker state is the same now ($t = 0$) as it is t seconds in the future.

Table 5.1: Same-label prior probabilities of EgoCom train and test sets. $p(s_t = s_0)$ as shorthand for $p((s_t = True \wedge s_0 = True) \vee (s_t = False \wedge s_0 = False))$, i.e. the probability that the speaker label does not change in t seconds.

Relevant Tasks	Prior, Future (s)	EgoCom **Test** set $t=1$	$t=5$	$t=10$	EgoCom **Train** set $t=1$	$t=5$	$t=10$
1	$p(s_t{=}s_0)$	0.75	0.65	0.63	0.77	0.67	0.64
2, 3	$p(h_t{=}h_0)$	0.73	0.61	0.58	0.74	0.62	0.58
4	$p(m_t{=}m_0)$	0.62	0.48	0.45	0.65	0.50	0.46

These priors (see Table 5.1) provide relevant information for Tasks 1 - 4. We observed during the human evaluation experiment (see Sec. 5.3.4) that when labeling egocentric video, humans often predict who will be speaking in the future, not based on visual and audio cues, but by who is currently speaking. This qualitative feedback motivated further inspection of the predictive signal of these priors.

Beyond same-label priors, conditional priors form transition diagrams that illuminate social dynamics in EgoCom (see Figs. 5-5 and 5-6). For example, Fig. 5-5a suggests that participants are more likely to speak in moments of silence than the host and that this dynamic changes further in the future. Importantly, the train

set prior distributions (Fig. 5-5) closely match the test set (Fig. 5-6) – evidence that including prior information (current speaking label) at training time should improve posterior inference on the test set.

5.3.2 Feature Representations for Downstream Learning

Here, we explain how video (visual), audio, and text embedding representations are created for every video in EgoCom and used as input for training. We extract visual, audio, and text features from an overlapping sliding window at every 1 second. For each modality, separate embeddings that represent the past 4, 5, 10, and 30 seconds of data are created, yielding 12 feature embeddings for every second of video in EgoCom, resulting in 555,000 features in total (138,750 features for each of the four histories).

Video embeddings Prior to computing video embeddings, videos are compressed to 480p MP4 (see Section 5.2.3). Video frames are sampled at 32 frames per clip for each past window. Video input frames are re-scaled to 171 x 128 and cropped to 112 x 112 patches. We extract the 2048-dimensional visual features from the last average pooling layer of R(2+1)D-101 model (Tran et al., 2018), pre-trained on Kinetics-400 (Kay et al., 2017) for human action classification. Visual feature extraction was performed on 8 NVIDIA V100 GPUs, requiring 3 days to compute the 550,000 features.

Audio embeddings The audio features used for this task are generated from a speaker identification model trained on the Voxceleb (Nagrani et al., 2017) dataset. The model uses a ResNet-34 backbone followed by an attention layer and is trained with both softmax loss and triplet loss. Hard negative examples are used in the triplet branch by taking the Cartesian product of all utterances from the same speaker in

the mini-batch with itself and fixing the resultant pairs positive anchor examples. Negatives are then randomly sampled from the utterances of other speakers which has a cosine similarity greater than a threshold with the anchor. The input to this model are segments of audio represented as 64-dimensional log Mel-filterbank energies. We compute the energies for each 25ms frame with a 10ms frame shift and concatenate the resulting 64 dimensional vectors together as input to the model. On a 24-core CPU machine, it takes roughly 8 hours to compute the 555,000 features.

Text embeddings We generate text embeddings on the human-annotated transcripts using FastText's Crawl 300 dimensional sub-word embeddings (Mikolov et al., 2018), pre-processed with tokenization and removal of white space and punctuation. Sentence vectors are created by normalizing and summing each word vector. As with the other features, text embeddings are created for every 4, 5, 10, and 30 second histories. On a typical CPU, computation time is negligible.

5.3.3 Predicting Turn-taking in EgoCom

Using pre-computed multi-modal feature representations, we predict the speaking label t seconds in the future for each of Tasks 1-4 using a simple MLP classifier, both without (i.e. likelihood) and with (i.e. posterior) inclusion of the current speaking label (i.e. prior) as input during training. The EgoCom validation set is used for early stopping and hyper-parameter tuning: the EgoCom test set is never accessed during training. We consider variations across: past window of input, future horizon to predict, feature modality, and prediction task, along with an ablation study to study the effect of model and test set choice. Top-1 accuracy on the EgoCom test set for each variation is reported in Tables 5.2 - 5.5 for each task. All results are seeded

for reproducibility.

MLP Model and Training Settings The MLP model used for all tasks in Sec. 5.3 has one input layer, one hidden layer, and output layer with batch normalization and dropout after the ReLu activation of each layer. A cross-entropy loss function is used with a softmax output for multiclass classification (Task 5.5) and a sigmoid output for binary classification (Tasks 5.2 - 5.4). For all tasks, we train using the Adam optimizer with weight decay = 0.001, for 40 epochs. We perform a small grid search with hyper-parameters: learning rate [1e-3, 3e-3, 5e-3], dropout [0.1, 0.5], and AMSGrad [True, False]. Among these settings, we select the model with the lowest cross entropy loss on the EgoCom validation set at each epoch of training. The EgoCom test set is never accessed at any point during training or tuning.

Table 5.2: (**Task 1**) Top-1 EgoCom test accuracy for whether any person will be speaking in 1-10 seconds given that person's features. Columns comprise how much past data is included in the feature input and how far in the future we predict. Rows comprise the modality of input used and whether the prior (current speaker) label is included as a feature. Max score for each (past, future, prior) triad is in bold. Random Perf. is 50%. Always predicting 0 (not speaking) yields 65% accuracy.

(data used for training)		Past (s)	4				5				10				30			
Use Prior	Modalities	Future (s)	1	3	5	10	1	3	5	10	1	3	5	10	1	3	5	10
False	text		68.2	64.8	64.7	64.7	68.0	65.1	65.0	65.0	65.5	65.0	65.0	65.0	64.9	64.9	64.9	**64.9**
(likelihood)	video		67.2	64.8	64.7	64.7	65.0	65.0	65.0	65.0	65.1	65.0	65.0	65.0	64.9	64.9	64.9	64.9
	audio		72.3	**69.3**	**67.2**	64.9	71.6	66.7	65.0	65.0	**69.8**	65.6	65.3	65.0	65.2	64.9	64.9	64.9
	text+video		69.9	64.8	64.7	64.7	67.4	65.3	65.0	65.0	65.6	65.0	65.0	65.0	65.6	65.3	64.9	64.9
	text+audio		**73.4**	68.3	66.7	65.0	**72.1**	**67.1**	65.0	65.1	69.6	**66.5**	65.4	65.0	65.8	64.9	64.9	64.9
	video+audio		71.0	67.0	66.4	**65.8**	70.9	66.6	65.1	65.2	69.1	66.1	65.4	65.0	**67.6**	65.1	**65.3**	64.9
	text+video+audio		72.4	67.6	65.2	64.8	72.1	66.0	**65.4**	**65.9**	69.5	65.4	**65.6**	**65.2**	66.7	**66.1**	65.0	64.9
True	text		74.9	65.2	64.7	64.7	75.5	66.7	65.0	65.0	74.8	65.5	65.0	**65.0**	**75.0**	65.5	64.9	64.9
(posterior)	video		73.7	65.0	64.7	64.7	73.0	65.6	65.1	65.0	73.4	65.7	65.0	65.0	74.4	68.2	66.1	**65.2**
	audio		76.0	**70.1**	66.9	65.0	75.2	67.4	65.2	65.0	75.1	67.0	65.2	65.0	74.6	65.2	64.9	64.9
	text+video		74.7	65.1	64.8	64.7	74.4	65.9	65.2	65.0	73.6	67.0	65.3	65.0	73.7	68.1	66.2	64.9
	text+audio		**76.2**	69.2	**67.3**	65.2	**75.7**	**68.5**	65.4	65.0	**75.2**	**68.2**	65.5	65.0	75.0	65.6	65.0	64.9
	video+audio		75.0	68.1	65.0	64.7	75.1	66.9	65.6	65.0	73.5	68.1	**67.0**	65.0	74.5	68.3	66.2	65.0
	text+video+audio		75.4	67.7	65.2	**65.4**	75.5	66.2	**66.6**	**65.2**	73.9	67.4	66.0	65.0	74.3	**69.0**	**66.5**	65.0

Table 5.3: (**Task 2**) Top-1 EgoCom test accuracy predicting whether the host will be speaking in 1-10 seconds given the host's features. Random Perf. is 50%. Always predicting 0 (not speaking) yields 51% accuracy.

(data used for training)		Past (s)	4				5				10				30			
Use Prior	Modalities	Future (s)	1	3	5	10	1	3	5	10	1	3	5	10	1	3	5	10
False	text		65.6	60.5	56.0	**57.3**	64.7	59.9	56.2	**57.7**	58.9	55.7	54.1	52.6	51.6	51.0	50.2	51.2
(likelihood)	video		59.2	53.6	52.5	56.0	59.8	52.8	56.9	56.6	58.4	57.5	50.6	50.8	56.8	56.3	**55.3**	53.8
	audio		67.9	**62.1**	**58.1**	52.5	67.5	**62.0**	**58.6**	53.8	**64.6**	60.5	54.7	52.3	58.2	52.1	52.8	50.5
	text+video		64.0	54.3	55.3	56.8	64.9	56.6	56.0	56.7	57.8	57.5	**56.1**	**55.0**	**59.5**	**57.2**	51.4	**54.0**
	text+audio		**68.3**	62.1	58.1	54.8	**67.6**	61.8	58.3	52.9	64.6	**60.9**	55.4	53.0	55.6	50.3	51.0	50.2
	video+audio		66.7	58.1	54.7	56.3	66.4	59.7	57.7	56.7	62.5	59.8	52.8	53.0	59.5	56.2	53.3	51.0
	text+video+audio		67.4	61.3	53.6	56.8	67.4	60.5	53.7	57.1	63.8	59.0	52.2	53.1	58.5	55.9	53.3	50.3
True	text		71.6	62.2	58.7	57.9	72.3	62.3	58.2	**57.9**	71.2	61.5	**58.7**	55.4	72.2	61.8	56.8	51.9
(posterior)	video		67.5	56.8	53.2	57.3	72.2	58.5	58.0	57.1	69.7	60.9	55.3	55.8	71.3	**62.6**	**60.2**	**56.7**
	audio		**72.2**	62.6	**59.9**	55.6	**72.5**	**63.3**	**59.8**	55.1	**72.1**	62.2	57.4	54.7	**72.4**	62.4	57.2	50.3
	text+video		68.6	58.4	53.4	57.7	72.2	60.4	58.8	57.0	67.8	59.0	53.2	**56.8**	71.5	62.6	59.7	54.6
	text+audio		71.7	**62.7**	58.9	56.5	72.1	63.2	59.4	56.9	70.6	**62.8**	58.4	55.4	71.8	59.8	57.4	50.3
	video+audio		69.4	60.2	55.4	**58.5**	71.8	62.4	57.0	57.4	68.7	60.0	53.5	54.0	71.6	61.4	59.5	52.9
	text+video+audio		71.1	60.8	56.8	57.8	71.7	62.4	55.8	56.9	70.0	59.8	53.5	53.8	71.2	62.5	59.2	51.4

Table 5.4: (**Task 3**) Top-1 EgoCom test accuracy predicting whether the host will be speaking in 1-10 seconds given the concatenation of all participant's features. Random Perf. is 50%. Always predicting 0 (not speaking) yields 53% accuracy.

(data used for training)		Past (s)	4				5				10				30			
Use Prior	Modalities	Future (s)	1	3	5	10	1	3	5	10	1	3	5	10	1	3	5	10
False	text		64.0	58.9	56.6	54.5	63.0	58.7	56.4	56.0	60.5	57.2	56.9	55.3	56.2	55.2	55.3	55.3
(likelihood)	video		58.8	57.2	57.0	54.6	59.6	57.3	56.7	55.4	59.8	57.3	56.6	54.8	55.2	55.5	55.0	55.2
	audio		65.7	60.0	56.9	54.4	65.3	60.4	**57.2**	**56.4**	62.4	57.9	56.6	**56.2**	**57.5**	**56.7**	56.0	55.7
	text+video		63.1	58.2	56.2	**55.0**	62.9	57.8	56.7	55.2	60.3	58.1	**57.8**	53.8	55.9	56.1	55.2	55.2
	text+audio		**66.1**	**60.3**	56.6	54.2	**66.3**	60.5	56.9	55.8	62.6	58.1	56.8	56.0	57.1	56.3	**57.1**	55.3
	video+audio		65.1	59.3	57.0	54.3	64.4	**60.6**	56.6	55.7	**62.7**	**58.9**	56.5	53.3	56.3	56.4	55.0	55.7
	text+video+audio		66.1	59.8	**57.8**	54.9	64.8	60.1	56.9	55.3	61.8	57.8	55.8	55.1	55.9	55.9	55.0	**56.2**
True	text		68.1	59.7	57.8	**55.2**	68.5	**60.5**	56.3	**56.2**	66.6	**59.4**	57.4	55.4	**67.8**	**58.4**	**56.1**	55.3
(posterior)	video		63.1	57.6	54.7	54.3	65.2	58.9	55.6	55.3	62.8	57.1	57.0	55.4	63.6	57.0	55.9	55.7
	audio		**68.5**	60.4	57.2	54.2	68.5	60.4	**58.2**	56.2	**67.4**	59.3	57.5	55.9	67.7	56.7	56.1	55.2
	text+video		64.7	57.8	**58.1**	54.8	66.3	59.7	55.6	55.1	64.0	58.6	56.4	54.9	62.8	55.7	54.4	**56.8**
	text+audio		68.0	**60.7**	57.3	54.3	**68.6**	60.5	57.3	56.1	67.1	58.2	57.1	**56.5**	67.0	57.0	56.0	55.1
	video+audio		66.4	59.7	57.4	55.1	66.9	60.2	57.2	55.3	62.7	58.7	**58.0**	55.0	63.4	55.4	55.0	55.5
	text+video+audio		67.1	60.3	57.3	54.6	67.1	60.3	56.9	55.4	64.1	58.1	57.2	55.0	63.4	55.4	54.1	55.7

Table 5.5: (**Task 4**) Top-1 EgoCom test accuracy for predicting which of person 1 (host), 2 (participant), 3 (participant), or no one (label 0) will be speaking. Random Perf. is 25%. Always choosing label 1 (the host) yields 46% accuracy.

(data used for training)		Past (s)	4				5				10				30			
Use Prior	Modalities	Future (s)	1	3	5	10	1	3	5	10	1	3	5	10	1	3	5	10
False	text		53.5	47.8	47.3	45.9	51.6	47.5	45.6	44.9	48.6	45.1	44.8	45.1	44.8	44.0	44.8	44.7
(likelihood)	video		44.6	45.3	43.9	42.2	45.4	45.0	44.9	44.9	44.4	45.8	45.1	44.4	43.2	**44.8**	44.8	44.6
	audio		53.7	**48.6**	**47.8**	**46.3**	52.6	48.0	**46.3**	**45.5**	**49.9**	46.4	**46.3**	**45.2**	**45.6**	43.4	44.8	43.6
	text+video		48.3	44.9	43.7	44.5	48.5	45.3	45.4	44.5	45.1	44.2	45.1	44.5	40.1	44.5	44.7	44.5
	text+audio		**54.5**	48.3	47.7	46.0	**53.5**	**48.2**	46.2	45.3	49.9	**47.1**	45.7	44.7	44.0	44.4	**44.9**	44.7
	video+audio		51.6	47.2	46.9	42.1	52.2	46.9	45.5	44.9	46.9	45.0	45.2	44.9	44.0	44.3	44.9	**44.8**
	text+video+audio		53.0	47.0	46.8	44.0	53.1	46.6	45.6	44.9	47.4	45.7	45.7	44.9	42.7	43.3	43.6	44.4
True	text		56.8	48.8	47.5	45.8	56.4	47.9	45.3	44.6	53.5	46.8	45.2	45.0	**55.2**	**46.6**	44.8	44.7
(posterior)	video		48.2	45.3	42.7	44.3	51.8	45.6	45.8	**45.8**	45.8	45.7	44.7	44.8	50.7	45.7	44.7	44.7
	audio		56.5	**49.0**	47.7	**46.2**	**57.2**	48.4	46.4	44.9	54.2	**47.0**	**46.3**	**45.1**	55.1	45.7	**45.0**	44.5
	text+video		52.1	46.8	44.4	44.4	53.2	46.1	45.8	44.8	47.9	45.6	44.6	44.9	49.8	46.2	43.1	44.7
	text+audio		**56.9**	49.0	**47.8**	46.1	56.9	**48.5**	46.3	44.8	**54.3**	47.0	46.0	44.7	54.4	45.9	45.0	**44.9**
	video+audio		53.5	47.4	46.8	45.8	54.3	47.7	**46.8**	45.1	49.4	46.7	44.6	44.6	50.7	44.6	44.1	44.7
	text+video+audio		55.4	47.0	46.7	43.9	55.1	47.3	46.3	44.8	48.7	45.9	45.3	44.2	50.7	44.8	43.2	42.6

Table 5.2 reports the results for Task 1. The top-left entry in Table 5.2 is read as: *The test accuracy predicting whether a given person will be speaking in 1 second given the past 4 seconds of that speaker's text features is 68.2%.* "A speaker's features" means a subset of the video capture between their eyes, the audio captured near their ears, and transcripts. The input modality with the max value for each (past, future, prior) triad is in bold. Table 5.4 (Task 3) and Table 5.5 (Task 4) report test accuracy for models trained using all three participant's features concatenated together; for both tasks, we only consider conversations where all three-participants are wearing a recording device for a static input size. This filtering explains the deviation in baseline accuracies at the end of the captions in Tables 5.3 and 5.4.

Observed trends Referencing Tables 5.2 - 5.5, we observe a number of trends consistent across all tasks. First, test accuracy tends to decrease significantly when the MLP is trained with features averaged over a larger past/history, indicating that

Table 5.6: Ablation study of Tasks 1 - 4 with Use Prior = False and Past = 4s. The study varies model used for training and the test set, across input modality and how far in the future to predict who will be speaking.

Input Modality	text			video			audio			txt+vid			txt+aud			vid+aud			txt+vid+aud		
Future (s)	1	5	10	1	5	10	1	5	10	1	5	10	1	5	10	1	5	10	1	5	10
EgoCom test set accuracy (%)																					
Task 1 w/ Naive Bayes	65	57	55	57	54	54	67	62	61	63	58	56	67	61	59	65	60	58	66	60	58
Task 1 w/ Random Forest	70	67	65	66	65	65	72	68	67	70	67	65	72	69	67	71	68	67	72	68	67
Task 1 w/ MLP	68	65	65	67	65	65	72	67	65	70	65	65	73	67	65	71	66	66	72	65	65
Task 2 w/ Naive Bayes	65	59	57	54	52	52	66	60	57	60	55	53	66	61	58	65	57	55	66	60	57
Task 2 w/ Random Forest	66	60	57	61	56	55	67	60	58	66	60	58	68	60	58	66	60	58	67	61	59
Task 2 w/ MLP	66	56	57	59	53	56	68	58	53	64	55	57	68	58	55	67	55	56	67	54	57
Task 3 w/ Naive Bayes	62	57	54	55	53	53	65	57	54	59	56	54	66	58	55	63	57	56	64	58	56
Task 3 w/ Random Forest	65	58	55	59	55	54	65	57	56	65	57	54	66	58	55	65	57	56	64	57	56
Task 3 w/ MLP	64	57	55	59	57	55	66	57	54	63	56	55	66	57	54	65	57	54	66	58	55
Task 4 w/ Naive Bayes	38	32	30	32	32	31	45	41	38	36	33	31	45	37	33	41	35	33	42	35	33
Task 4 w/ Random Forest	54	47	45	47	46	46	54	48	46	53	47	46	54	48	46	55	48	47	54	48	47
Task 4 w/ MLP	54	47	46	45	44	42	54	48	46	49	44	45	55	48	46	52	47	42	53	47	44
Cross-validation accuracy (%) of the entire EgoCom dataset																					
Task 1 w/ Naive Bayes	67	60	57	56	54	53	70	64	63	64	59	56	70	63	61	66	61	58	69	62	60
Task 1 w/ Random Forest	72	67	65	65	65	65	73	69	67	72	67	65	74	69	67	73	68	67	73	68	67
Task 2 w/ Naive Bayes	66	59	55	52	50	49	68	59	56	58	52	50	68	60	56	65	56	52	66	57	53
Task 2 w/ Random Forest	66	58	55	50	45	44	68	58	54	62	52	47	68	58	54	65	53	48	65	53	48
Task 3 w/ Naive Bayes	65	55	52	52	50	50	68	59	55	56	51	50	69	60	56	64	55	52	66	56	52
Task 3 w/ Random Forest	68	58	54	52	48	47	68	58	53	64	53	49	68	58	54	66	54	50	66	54	50
Task 4 w/ Naive Bayes	47	38	34	35	32	32	54	43	39	39	33	32	53	42	37	45	36	33	47	37	33
Task 4 w/ Random Forest	57	49	46	42	39	39	56	46	44	54	45	41	57	47	45	55	44	41	55	45	42

in the case of three-person human conversation, the dynamics of turn-taking rely mostly on the last few seconds of interaction. Second, the inclusion of visual features during training decreases accuracy, likely because turn-taking depends more on the speech content than the egocentric view of the speaker and the high-dimensional visual features increased the complexity of the learning manifold during training. Using only video features to predict turn-taking (a speech-oriented task) results in poor performance that breaks these trends in some settings (see Table 5.3, video, past of 4s). Finally, accuracy drops off significantly the further you predict in the future.

Comparison of prior, likelihood, and posterior The top likelihood and posterior test accuracies from Tables 5.2 - 5.5 are shown in Table 5.7 along with their corresponding priors from Table 5.1. The results indicate the strength of the prior, indicating the value of the aligned multi-perspective data used to compute the prior (current speaker label) at inference time. The prior baseline outperforms the posterior in some cases, however, for longer future horizons, the posterior outperforms the prior. This indicates that while the likelihood may under-perform the prior, the MLP model learns turn-taking from the data, not just the prior. As expected, the posterior outperforms the likelihood in most cases.

Unlike Tables 5.2 - 5.4, in some settings, accuracies in Table 5.5 dip below that of a naive model that predicts label 1 (the host). For fair comparison, Tasks 1-4 use the same model and training settings (Sec. 5.3.3) across inputs, features, and output. In complex settings, e.g. multiclass prediction) with large past window and future horizon, without further hyper-parameter tuning, poor local minima may be found.

Table 5.7: Top likelihood and posterior test accuracy from Tables 5.2 - 5.5. Test set prior scores are copied from Table 5.1.

Relevant	Prior			Likelihood			Posterior		
Tasks, Future (s)	1	5	10	1	5	10	1	5	10
1	75.6	65.5	63.2	73.4	67.2	65.9	76.2	67.3	65.4
2, 3	72.6	61.3	58.5	68.3	58.6	57.7	72.5	60.2	58.5
4	62.3	47.7	44.5	54.5	47.8	46.3	57.2	47.8	46.2

Role of multiple synchronized perspectives Surprisingly, concatenating participant features to predict the host's speaking state decreased overall accuracy (cf. Table 5.3 versus Table 5.4). Two likely causes are: (A) the host is less influenced by participants, than vice versa, such that the added data actually adds noise, or (B) the 8000-dimensional concatenated feature input is too complex for the simple MLP model – the same model is used to fairly compare results across tasks.

We observe strong evidence to support cause (B). When the MLP is trained only on audio and text features, without the 6144-dimensional video embedding from the concatenated three perspectives, accuracy *increases* from Task 2 to Task 3 (see Fig. 5-7). In more challenging settings (larger past window and future), we observe increased accuracy and stability (across future horizon) when all synchronous participants' features are used. These results suggest a need for further exploration of the effects of synchronous multi-perspective multi-modal data in conversational AI.

Ablation Study We conduct an ablation study (see Table 5.6) to validate our findings throughout this section, reproducing the results in Tables 5.2 - 5.5 with a past window of 4s. We replicate these experiments with the scikit-learn (Pedregosa

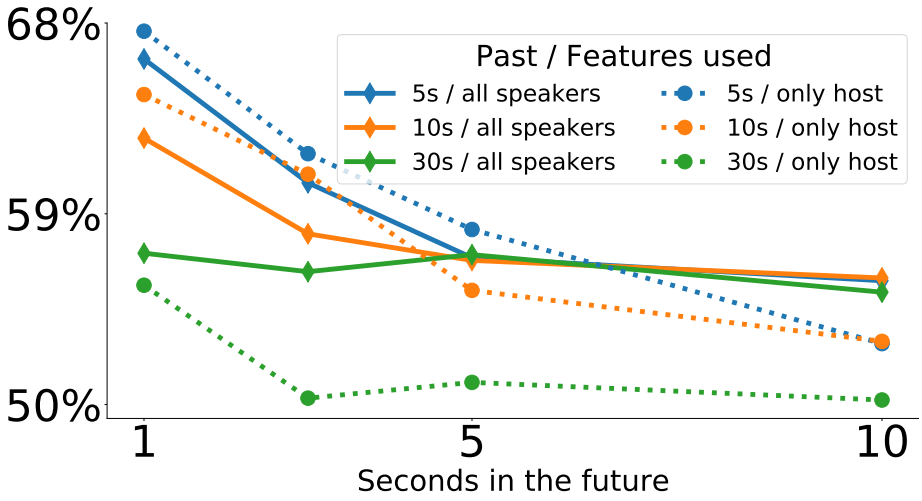


Figure 5-7: An example of an MLP trained with audio+text features where test accuracy increases when all synchronous participants' features are used, particularly for larger past and future. This figure compares Task 3 versus Task 2.

et al., 2011) implementations of Random Forest and Gaussian Naive Bayes classifiers, with default settings, reporting top-1 accuracy for both the EgoCom test set as well as 5-fold cross-validation to study how the choice of EgoCom test set may bias results.

As shown in Table 5.6, there is no significant difference between cross validation accuracy and test set accuracy: both exhibit a (1) decrease in performance further in the future and/or with increased past window of feature representation, (2) for the same classifier, results are within 3% at least 90% of the time, and (3) training with audio features exhibits highest accuracies. These are trends are similarly observed by the MLP benchmarks. The MLP and random forest classifiers perform similarly in Table 5.6, likely because the feature embedding inputs were pre-computed using neural architectures, such that our classification task is like fine-tuning the output layer of a neural network, and a random forest layer is highly expressive in comparison with MLP forward and softmax layers.

5.3.4 Human Performance on Turn-taking

Human accuracy for Task 1 is reported in Table 5.8 and compared with machine accuracy (Table 5.2) in Fig. 5-8. Three human raters were independently presented with 5 seconds of audio, video, or video+audio and asked to predict if the embodied speaker will be speaking or not, in 1, 5, and 10 seconds in the future. The task was performed by each rater every 10th second for every perspective in each conversation in the test set. Across all configurations, 18,732 human predictions were recorded. To avoid redundancy, only one of the three modalities (audio, video, audio+video) was labeled for each of the three perspectives in each conversation.

Table 5.8: Average human test accuracy, standard deviation, and Cohen's Kappa inter-rater reliability for Task 1.

Future	1			5			10			**AVG**
Modality	a	av	v	a	av	v	a	av	v	
Accuracy	0.79	0.74	0.72	0.69	0.68	0.67	0.69	0.68	0.59	0.69
Std. dev.	0.08	0.11	0.14	0.11	0.11	0.08	0.11	0.10	0.04	0.10
Coh. Kap	0.55	0.57	0.48	0.47	0.48	0.45	0.41	0.40	0.41	0.47

Inter-rater reliability is measured using Cohen's Kappa for every pair of raters for each video. To control for label quality, in each video, we require a Cohen's Kappa $>$ 0.3 with another rater's labels (44% removed). Cohen's Kappa for each (modality, future) setting is reported in Table 5.8.

Fig. 5-8 compares human and machine performance on Task 1. In all cases, the MLP posterior model is within 5% of human performance. Humans perform notably worse when presented video without audio, likely because predicting speaking without

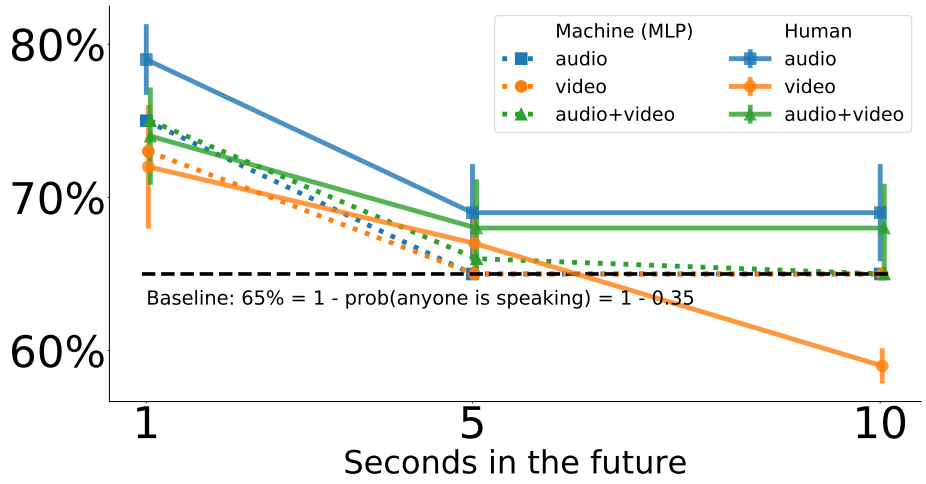


Figure 5-8: Human and machine (MLP) baseline performances on EgoCom test set for Task 1 across modality of input and duration into the future. The prior (speaker label at 0 seconds) is included during MLP training because humans also infer this prior. Past history window of 5 seconds is used for both. Raw values for human performance are shown in Table 5.8.

being able to hear, using only gestures of visible peers, is remarkably challenging. For a 10s future horizon, the MLP always outputs "not speaking", yielding a baseline accuracy of 65% (see dashed line in Fig. 5-8).

5.4 Application: Noisy Multi-Speaker Speech Recognition

Here we demonstrate how the unique nature of embodied data may simplify the task of *global transcription*: computing a time-stamped, multi-speaker identification and transcription. To obtain ground truth transcriptions, a third-party human annotation service transcribed the entire EgoCom dataset.

As an asynchronous baseline, we use Google Cloud's speech-to-text service to

transcribe each person's audio in a given conversation and compute mean accuracy with ground truth. Transcription accuracy is computed as $1 - WER$, where WER is the word error rate defined by the Wagner-Fischer edit-distance algorithm (Wagner and Fischer, 1974). Because we use a pre-trained speech-to-text service, we do not need a train and test set and instead compute accuracy on the entire EgoCom dataset. Accuracy is computed per conversation, and overall accuracy is computed as a weighted mean, weighted by the number of words in each conversation.

Asynchronous baseline: 30.7% accuracy We use Google's single-source speech-to-text (Pundak et al., 2018; Chiu et al., 2018) service to transcribe the audio source for every video in EgoCom. This service provides time-stamped word-level transcriptions with a confidence for every transcribed word. For each conversation, we compute $1 - WER$ for each source and take the average. The weighted average accuracy across all conversations, weighted by the number of words in each conversation, is 30.7%. Low accuracy occurs because the speech-to-text system only has access to a single audio source. Qualitatively, all three speakers can be heard in each audio stream, however, the egocentric audio is significantly louder which may "trick" the system into filtering out non-egocentric audio as noise.

As an alternative baseline, the loudness issue could be avoided by adding the signals prior to transcription, however, such a baseline is not asynchronous because it requires aligned multi-perspective data at inference time. We study how the unique nature of synchronous, multi-perspective EgoCom data can simplify tasks like global transcription.

Synchronous multi-perspective data: 54.8% accuracy We use the same Google Cloud transcriptions from the baseline accuracy experiment, but combine the outputs using the maximum confidence for each word, exploiting that EgoCom

Table 5.9: Global transcription accuracy across demographics.

Gender	Native speaker	Speaker is host	Word count	Baseline accuracy	Combined accuracy	Speaker ID accuracy
female			1055	0.31	0.54	0.75
female	✓		31,666	0.29	0.55	0.76
male			21,174	0.30	0.54	0.77
male	✓		23,344	0.31	0.55	0.76
male	✓	✓	81,826	0.31	0.55	0.77

Table 5.10: Global transcription accuracy across influencers.

Native speaker	Music noise	Fan noise	Word count	Baseline accuracy	Combined accuracy	Speaker ID accuracy
			17,577	0.31	0.55	0.77
		✓	2467	0.25	0.51	0.76
	✓		2185	0.27	0.51	0.79
✓			96,448	0.32	0.56	0.77
✓		✓	11,701	0.28	0.53	0.73
✓	✓		28,687	0.29	0.53	0.76

sources are egocentric and aligned. Our approach is three steps: (1) label all transcriptions for source i as being spoken by speaker i, (2) join all transcriptions in a table indexed by time, sorting the start-time of each transcribed word, and (3) starting from row zero, if two or more rows from different speakers have the same word transcription, within 0.1 seconds of each other, then remove all rows except the one with max confidence. The output is a time-stamped global transcription with speaker ids. Using this approach, we achieve an overall accuracy of 54.8%, a 79% improvement over the baseline. The improvement results from egocentric

synchronous audio: the source worn by the speaker typically yields the prediction with highest confidence score, disambiguating the source for each spoken word in order to obtain a speaker-identified global transcription.

Synchronous speaker identification accuracy is 76.8%. We compute speaker identification accuracy by considering every time both the ground truth and the global transcription speaker labels (from above) both provide a speaker label for a given 1 second time window. In total there are 534,500 labels. Speaker id accuracy is computed as the number of same labels divided by the total number of co-occurrences for each conversation, with the overall accuracy of 76.8% as a weighted sum. Note there is no notion of speaker identification for the baseline approach and thus no comparison for speaker identification.

Tables 5.9 and 5.10 report accuracies for each approach across demographics and background noise. The results indicate that the advantages of synchronous multi-perspective data for global transcription are unaffected by demographics (no performance decrease) or background noise (similar decrease in performance as the baseline).

5.5 Related Work

Common benchmark datasets in computer vision (Lin et al., 2014; Russakovsky et al., 2015; LeCun, 1998) have been essential in catapulting advances in machine learning, but contain data from a third-party perspective, losing contextual egocentric information such as head pose, or imperceptible sounds like the quiet breath one takes before speaking. Instead, EgoCom is multi-disciplinary, combining synchronized multi-perspective, multi-modal communications data and egocentricity (del Molino

et al., 2017) with elements of conversational AI (Gao et al., 2019), natural language, audio, computer vision, and spatial AI (Smith et al., 1990).

There are a number of related video-based datasets. *Action classification* datasets include Kinetics, a video dataset for human action classification (Kay et al., 2017), ActivityNet, a video dataset for action classification and temporal localization (Fabian Caba Heilbron and Niebles, 2015), and AVA, a dataset of spatio-temporally localized atomic visual actions (AVA) (Gu et al., 2018). *Multi-modal AI* datasets include AVA-ActiveSpeaker, an audio-visual dataset for speaker detection (Roth et al., 2019), VGG lip reading dataset, an audio-visual dataset for speech recognition and separation (Chung et al., 2017), Mosi, a multimodal corpus of sentiment intensity (Zadeh et al., 2017, 2016), and OpenFace, a multi-modal face recognition (Baltrušaitis et al., 2016). The two major advantages of EgoCom are egocentricity and the inclusion of multiple participant's synchronized audio and video, which as we show, simplifies multi-speaker applications.

There are several related prior works that study social interactions in egocentric vision. Fathi et al. (2012) present a first-person visual dataset and detection and recognition tasks. Rehg et al. (2013) analyze children's social and communicative behaviors based on video and audio data. Yonetani et al. (2016) collect a human interaction dataset and study action and reaction recognition. Joo et al. (2019) present a task and a 3D motion dataset to understand human social interactions. Li et al. (2019) introduce a dual relation modeling framework for egocentric human interactions using vision signals. EgoCom differs from (Fathi et al., 2012) and (Yonetani et al., 2016) in that it captures multi-perspective multi-modal communication signals that can be exploited beyond vision tasks. EgoCom provides a new dataset to benchmark existing models, e.g. (Li et al., 2019), as well as future extensions that leverage multi-perspective multi-modal content captured in a natural social settings.

EgoCom combines multi-modal AI (Baltrušaitis et al., 2019; Ozkan et al., 2010) with egocentricity. Multi-modal data can be useful for tasks like multi-party speech recognition and predicting turn-taking by combining granularity of verbal and non-verbal cues (Stratou and Morency, 2017; Picard, 2000). For example, (Morency et al., 2008) is related to predicting turn-taking, but does not take advantage of multiple egocentric perspectives afforded by EgoCom. Numerous egocentric datasets exist (Lee et al., 2012; Lu and Grauman, 2013; Fathi et al., 2011; del Molino et al., 2017), but the main advantage of EgoCom over these datasets is the conversational content. Whereas these previous datasets were action-oriented, EgoCom is communications oriented in an effort to link conversational AI, audio, and natural language tasks with egocentric computer vision. This feature makes EgoCom natural for *looking to listen* tasks (Ephrat et al., 2018; Arandjelovic and Zisserman, 2018; Gao et al., 2018).

Like Owens and Efros (2018), where multi-modal data is shown to be useful as a source of self-supervision, we demonstrate how multi-observer data can also be employed to generate training labels for the prediction of turn-taking without human supervision. Turn-taking is also related to keyword-spotting (Wu et al., 2018; Zhang and Glass, 2009): words like “Okay" or “Well" may indicate finishing or starting speech. Unlike these efforts, we do not solve this task directly, but indirectly while predicting turn-taking.

5.6 Future Work

The EgoCom dataset introduced in this chapter is a unique, first-of-its kind dataset. In this section, we detail the natural next steps for real-time turn-taking support and new kinds of research questions enabled by the existence of the egocentric communications dataset.

Towards live turn-taking prediction in human conversations The approach presented in this chapter uses a simple MLP classifier trained with audio, video, and text embeddings from pre-trained models to allow for real-time turn-taking prediction. For example, using a pre-trained model with a 5 second past window, inference time takes less than 1 second for all Tasks 1-4. The results in Tables 5.2 - 5.5 suggest a real-time assistance system is plausible. A natural next step is to build such a system – whereby a vibration-enabled device can be worn and provide real-time conversational cues in conversations.

5.6.1 Research Areas Enabled

The EgoCom dataset enables new research opportunities through the combination of embodied visual, audio, and text modalities from multiple simultaneous aligned perspectives in natural conversation. EgoCom is intended to enable new research directions in the following:

Question Answering About 20% of the conversational content encompasses a question-based word-guessing game where participants must guess a word placed on their forehead based on answers to binary yes/no questions that they ask. This is relevant for AI systems built on knowledge graphs of objects and properties. Throughout the dataset we ask questions about objects and their relationships like:

- "What's that called?" *(the answer names the object)*
- What color is the <object>? *(object has previously been named)*
- What shape is the <object>? *(the question names the object)*
- Name the <object> <relative to (e.g. above)> <object>?

Conversational AI EgoCom is a natural dataset for predicting turn-taking, lip-reading to predict speech from video, semantic analysis and linguistic tasks, automatic speech recognition, natural language understanding, and enhancing predictions through visual cues.

Audio EgoCom contains multiple aligned perspectives making the dataset helpful for multi-modal multi-source separation tasks as well as audio-only source separation. The multi-channel, multi-perspective audio enables beam-forming audio analysis, speaker localization, and pose estimation applications. Additionally, EgoCom works well for self-supervised learning with audio because each person's audio captures the same conversation, the difference being egocentric audio is louder, providing strong cues for speaker identification and source separation.

Human Learning EgoCom contains contents of human learning, such as participants teaching card games to one another or learning about properties of objects in the room. It is useful for *meta-understanding* when a learner understands, providing sources for AI agents to understand and simulate human learning processes.

5.7 Chapter Contributions

The findings in this chapter demonstrate how synchronized multi-perspective egocentric data can simplify baseline solutions for two example applications, and more generally, the interdisciplinary nature of the EgoCom dataset. The turn-taking application demonstrates unique new applications enabled by EgoCom and the global transcription application illuminates how aligned egocentric capture may

simplify practical problems. Egocentric communications motivates the need for further study of applications in embodied AI and egocentric data across conversational analysis, computer vision, audio, and machine perception.

The contributions of this chapter include:

1. Created and open-sourced the first multi-modal, synchronized multi-perspective egocentric communications dataset. We provide binaural 2-ear audio from the perspective of human ears, egocentric embodied video from the perspective of human eyes, and textual transcripts. The egocentric communications code base and dataset is open-source at https://github.com/facebookresearch/EgoCom-Dataset.

2. Established a baseline accuracy for embodied turn-taking prediction in human conversations, with an inference time that is shorter than the prediction window, establishing the plausibility of real-time turn-taking conversational support.

3. Established a baseline global transcription accuracy with synchronized multi-perspective egocentric data.

Chapter 6

Conditional Rap Lyric Generation with Denoising Autoencoders

"I am the oldest. The lyrics, they just follow my orders."

- generated by the artificially intelligent system described in this chapter (2020)

The ability to combine symbols to generate meaningful language is a defining characteristic of human intelligence, particularly in the context of artistic storytelling through lyrics. In Section 6.2, we develop a method for synthesizing a rap verse based on the content of any text (e.g., a news article), or for augmenting pre-existing rap lyrics. Our method, called RAPFORMER, is based on training a Transformer-based denoising autoencoder to reconstruct rap lyrics from content words extracted from the lyrics, trying to preserve the essential meaning, while matching the target style. RAPFORMER features a novel BERT-based paraphrasing scheme for rhyme enhancement which increases the average rhyme density of output lyrics by 10%. Experimental results (described in Section 6.4) on three diverse input domains (detailed in Section 6.3) show that RAPFORMER is capable of generating technically

fluent verses that offer a good trade-off between content preservation and style transfer. Furthermore, in Section 6.5, a Turing-test-like experiment reveals that RAPFORMER fools human lyrics experts 25% of the time.[1]

Attribution This chapter includes material previously published as (Nikolov et al., 2020). Nikola I. Nikolov, Eric Malmi, and Loreto Parisi contributed significantly to the material presented in this chapter. This work was supported in part by funding from Musixmatch.

Acknowledgements Aspects of the contents of this chapter were shaped by input from Alessandro Calmanovici, Scott Roy, Aliaksei Severyn, and Ada Wan, who engaged in discussion about the use of this work in real-world applications; and Simone Francia and Maria Stella Tavella from Musixmatch, who contributed technical support.

6.1 Introduction

Automatic lyrics generation is a challenging language generation task for any musical genre, requiring story development and creativity while adhering to the structural constraints of song lyrics. Here we focus on the generation of *rap lyrics*, which poses three additional challenges specific to the rap genre: (*i*) a verse in rap lyrics often comprises multiple rhyme structures which may change throughout a verse Bradley (2017), (*ii*) the number of words in a typical rap verse is significantly larger when compared to other music genres Mayer et al. (2008), requiring modeling of long-term

[1]We created two songs with lyrics generated by RAPFORMER: using the abstract of this chapter as input (see the suppl. mat., and `https://bit.ly/37ekn6i`), and using blog posts on AI and creativity as input, video available at `https://rapformer.page.link/demo`.

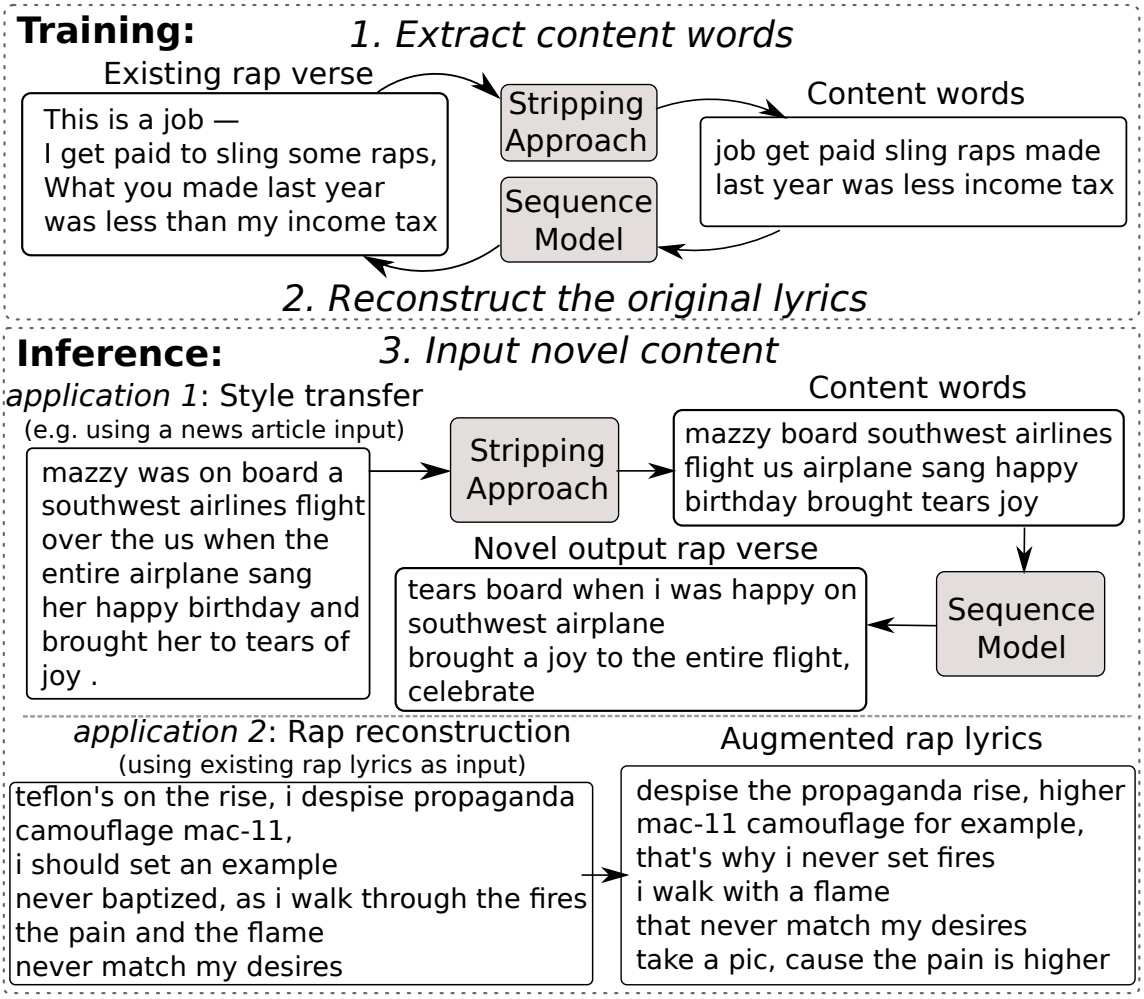


Figure 6-1: Overview of our approach to *conditional rap lyrics generation*. **Training**: (1) extract content words from existing rap verses, then (2) train sequence models to guess the original verses conditioned on the content words. **Inference**: (3) Input content from non-rap texts to produce *content-controlled* rap verses; or input existing rap verses to augment them.

dependencies, and (*iii*) the presence of many slang words.

Prior approaches to rap generation typically use *unconditional* generation Potash et al. (2015); Malmi et al. (2016). That approach synthesizes lyrics without providing any context that could be useful to guide the narrative development into a coherent

direction Dathathri et al. (2020). For example, generating rap lyrics on a specific topic, e.g., "cooking," is not possible with unconditional generation. Motivated by this, in this chapter, we propose a novel approach for *conditional* generation of rap verses, where the generator is provided a source text and tasked with transferring the style of the text into rap lyrics. Compared to unconditional generation, this task can support the human creative process more effectively as it allows a human writer to engage with the generator by providing the content of the lyrics while receiving automatic suggestions on how to improve the style of the lyrics to resemble the rap domain.

Our approach to conditional generation is to train sequence-to-sequence models Vaswani et al. (2017) to reconstruct existing rap verses conditioned on a list of content words extracted from the verses (Figure 6-1). By learning a mapping from content words to complete verses, we implicitly learn the latent structure of rap verses given content, while preserving the target output style of the rap lyrics. Model outputs are enhanced by a post-processing step (Section 6.2.2) that substitutes non-rhyming end-of-line words with suitable rhyming alternatives.

We test our method on three diverse input domains: short summaries of news articles, movie plot summaries, and existing rap lyrics. Automatic and human evaluations (Sections 6.4 and 6.5) suggest that our method provides a trade-off between content preservation and style compared to a strong information retrieval baseline.

6.2 Conditional Generation of Lyrics

Our approach to conditional generation of rap verses consists of three steps (Figure 6-1).

1. Given a dataset of rap verses, we apply a stripping approach to extract from each verse a set of *content words* that aim to resemble the main content of the original text, omitting any specific stylistic information.

2. We train a Transformer model Vaswani et al. (2017) to reconstruct the original rap verses conditioned on the content words. The model learns to generate the original verse, filling in missing stylistic information.

3. At inference time, we can input content words extracted from a text written in any style, such as a news article, resulting in novel output rhyme verses. After generation, we optionally apply a rhyme enhancement step (Section 6.2.2).

6.2.1 Stripping Approach

Given a dataset of original rap verses, our base approach to extracting content words involves preprocessing each verse to remove all stop words[2], numbers, and punctuation. To promote greater novelty[3] and variability in the outputs produced by our models, we additionally apply one of three noise types to the stripped content words:

Shuffle. We shuffle all of the content words on the sentence level (line level for rap verses). This type of noise forces our models to learn to rearrange the location of the input content words when generating the output rap lyric, rather than to merely copy words from the input in an identical order. A similar noising approach has been recently employed by Raffel et al. (2019).

[2]We use the list of English stopwords defined in NLTK.

[3]In early experiments, we tested training models using only this base approach. The models performed very well at reconstructing existing rap lyrics, however when the input was from a different domain, we observed very conservative outputs.

Drop. We randomly remove 20% of the input content words for the purpose of promoting generation of novel words, rather than only copying content words from the input.

Synonym. We replace 20% of the content words with synonyms obtained from WordNet Miller (1995). We pick words randomly and replace them with a random synonym. This type of noise promotes our models to learn to replace content words with synonyms, which might fit better in the context or style of the current output rap verse.

6.2.2 Rhyme Enhancement with BERT

To improve the rhyming fluency of our models, we implement a post-processing step for *rhyme enhancement (RE)* which modifies a generated verse to introduce additional end-of-line rhymes. Given two lines from a generated verse, such as:

where were ***you****?*
last year i was paid in a drought with no ***beginners***

RE iterates over each of the lines in the verse, replacing the ending words with a *MASK* token. The verse is then passed through a BERT model[4] Devlin et al. (2019) which predicts the $K = 200$ most likely replacement candidates for *MASK*. For exam ple, the replacement candidates for *you* might be *{they, we, I, it}*, and for *beginners* might be *{food, fruit, you, rules}*. We pick the candidate that leads to the highest increase in rhyming, determined by the length of the longest overlapping vowels in the two words Malmi et al. (2016). In the example above, replacing *beginners* with *food* maximizes the rhyme length, and the example becomes:

[4]We finetune a BERT base model on our rap verse dataset for 20 epochs.

where were you?

last year i was paid in a drought with no ***food***

Algorithm: Bert Rhyme Enhancement

input : lyrics verse $\mathbf{V} = \{l_0, ..., l_N\}$ consisting of N tokenized lines; number of BERT predictions K to consider.

output : modified $\mathbf{V}$ with enhanced rhyming.

Function `get_rhyming_replacement(`$\mathbf{V}$, *src_idx, tgt_idx, mask*`):`

- $src \leftarrow \mathbf{V}$ [src_idx][-1] `// get last word`
- $tgt \leftarrow \mathbf{V}$ [tgt_idx][-1]
- `// Predict most likely words.`
- $preds \leftarrow$ `bert_predictions` $(mask, K)$
- `// Compute original rhyme length.`
- $rl\_orig \leftarrow$ `rhyme_length` (src, tgt)
- **for** $pred \in preds$ **do**
  - $rl\_new \leftarrow$ `rhyme_length` $(pred, tgt)$
  - **if** $rl\_new > rl\_orig$ **then**
    - `// return replacement`
    - **return** $pred, rl\_new$
- **return** $target, rl\_orig$ `// return original`

for $i \leftarrow 1, 3, ..., N$ `// for each odd line` **do**

- `// Create two masks for the two consecutive lines.`
- $mask\_1 \leftarrow$ `mask_text` $(\mathbf{V}, \text{i})$
- $mask\_2 \leftarrow$ `mask_text` $(\mathbf{V}, \text{i} + 1)$
- `// Generate replacement candidates.`
- $cand\_1, rl\_1 \leftarrow$ `get_rhyming_replacement` $(\mathbf{V}, i + 1, i, mask\_1)$
  - `// replace last word at` i
- $cand\_2, rl\_2 \leftarrow$ `get_rhyming_replacement` $(\mathbf{V}, i, i + 1, mask\_2)$
  - `// replace last word at` i `+ 1`
- **if** $rl\_2 \geq rl\_1$ `// update lines in` $\mathbf{V}$ **then**
  - $\mathbf{V}$ $[i + 1]$[-1] $\leftarrow cand\_2$
- **else**
  - $\mathbf{V}$ $[i\ ]$[-1] $\leftarrow cand\_1$

return $\mathbf{V}$

Figure 6-2: Pseudocode for Bert rhyme enhancement.

The pseudo-code in Figure 6-2 contains a detailed implementation of our approach.

6.3 Experimental Setup

	News	**Movies**	**Rap**
# Pairs	287k/11k/11k	- / - /12k	165k/1k/1k
Sent. p.d.	3.7 ± 1.2	3.9 ± 1.6	10.5 ± 4.5
Tok. p.d.	57.9 ± 24.3	90 ± 27.6	91.8 ± 49.1
Tok. p.s.	15.1 ± 4.7	22.4 ± 11	9.5 ± 4.25

Table 6.1: Statistics of our datasets. *# Pairs* denotes the number of pairs used for training/validation/testing; *p.d.* is per document; *p.s.* is per sentence.

Datasets. We conduct experiments using three datasets. As our rap dataset, we use 60k English rap lyrics provided by Musixmatch.[5]

We split each lyric into verses (in the dataset, each verse is separated by a blank line), remove verses shorter than 4 lines in order to filter for song choruses and intros, and reserve 2k song lyrics for validation and testing. We use two datasets as our out-of-domain inputs: (1) the summaries from the CNN/DailyMail news summarization dataset Hermann et al. (2015) and (2) a subset of the CMU movie plot summary corpus Bamman et al. (2013). Since some of the movie summaries are very long, for this dataset, we filter summaries longer than 140 tokens and shorter than 40 tokens. Table 6.1 contains detailed statistics of the datasets used for training/validation/testing in our experiments.

Model details. As our sequence transducer, we use a 6-layer Transformer encoder-decoder model Vaswani et al. (2017). We initially train our models on the source domain (e.g., news articles) for 20 epochs, after which we finetune them on rap verses

[5] https://www.musixmatch.com/

for an additional 20 epochs, using the same stripping approach for both. We train all of our models on the subword level Sennrich et al. (2016), extracting a common vocabulary of 50k tokens from a joint collection of news summaries and rap lyrics. We use the same vocabulary for both our encoders and decoders and use the Fairseq library.[6] We train all of our models on a single GTX 1080 Ti card.

Generation details. During inference, we generate outputs using diverse beam search Vijayakumar et al. (2018) to promote greater diversity across the hypothesis space. We use a beam with a size of 24 and 6 diverse beam groups. Furthermore, we limit the maximum output sequence length to two times the length of the input content words and penalize repetitions of bigrams in the outputs.

To select our final output, we additionally implement a simple hypothesis reranking method. For each of the 24 final predictions on the beam, we compute two scores: the rhyme density (RD) of the text, following Malmi et al. (2016), as well as its repetition score:

$$rep(\mathbf{s}) = \frac{\sum_i overlap(\overline{\mathbf{s}_i}, s_i)}{|\mathbf{s}|}. \tag{6.1}$$

rep measures the average unigram overlap (see Section 6.1) of each sentence s_i in the text $\mathbf{s}$ with all other sentences of the text concatenated into a single string (denoted as $\overline{\mathbf{s}_i}$). We pick the hypothesis that maximizes: $score(\mathbf{s}) = RD(\mathbf{s}) - rep(\mathbf{s})$. Afterwards, we optionally apply our rhyme enhancement step, to further increase the frequency of rhymes in our outputs.

Bias mitigation Rap lyrics, like other human-produced texts, may contain harmful biases and offensive content which text generation models should not propagate further. Our conditional lyrics generation setup is less susceptible to this issue since the user

[6] https://github.com/pytorch/fairseq

		Rap reconstruction			Style transfer from movies		Style transfer from news	
	Model	**BLEU**	**Overlap**	**RD**	**Overlap**	**RD**	**Overlap**	**RD**
	INPUTS	-	-	0.84 ± 0.38	-	0.73 ± 0.2	-	0.72 ± 0.21
	IR NEWS	-	-	-	-	-	0.29 ± 0.09	0.74 ± 0.19
	IR RAP	-	-	-	0.19 ± 0.06	**1.02 ± 0.23**	0.17 ± 0.06	1.01 ± 0.24
RAPFORMER	SHUFFLE	10.27	**0.63 ± 0.13**	1.01 ± 0.31	**0.51 ± 0.11**	0.90 ± 0.23	**0.45 ± 0.12**	0.89 ± 0.26
RAPFORMER	SHUFFLE + RE	12.72	0.60 ± 0.12	1.10 ± 0.32	0.49 ± 0.10	0.96 ± 0.27	0.43 ± 0.11	0.98 ± 0.27
RAPFORMER	DROP	11.06	0.52 ± 0.11	1.03 ± 0.32	0.43 ± 0.10	0.90 ± 0.24	0.38 ± 0.10	0.93 ± 0.25
RAPFORMER	DROP + RE	09.81	0.50 ± 0.11	**1.13 ± 0.33**	0.40 ± 0.09	0.99 ± 0.27	0.36 ± 0.10	1.03 ± 0.26
RAPFORMER	REPLACE	**14.30**	0.57 ± 0.15	1.00 ± 0.30	0.43 ± 0.14	0.86 ± 0.28	0.34 ± 0.13	0.95 ± 0.27
RAPFORMER	REPLACE + RE	12.72	0.54 ± 0.15	1.10 ± 0.31	0.40 ± 0.13	0.98 ± 0.24	0.31 ± 0.12	**1.05 ± 0.28**

Table 6.2: Automatic metric results of RAPFORMER, using three alternative stripping approaches: SHUFFLE, DROP and REPLACE. Model names ending with "+ RE" denote use of the additional rhyme enhancement step (see Section 6.2.2). INPUT measures the result of the original input texts, for each of the three inputs (rap/movies/news). **Overlap** is the content preservation score, **RD** is the rhyme density metric. The highest results for each column are in bold.

provides the content, and the generator is supposed to modify only the style of the text. Yet, the model may learn to use inappropriate individual terms that are common in rap lyrics. To alleviate this, we maintain a "deny" list of words that the model is not able to generate.

6.4 Machine Evaluation

We conduct an automatic evaluation of RAPFORMER, using the test sets of each of our three datasets. Our focus is on measuring two components that are important for generating fluent conditional rap verses: preserving content from the input text to the output, and maintaining rhyming fluency during generation.

6.4.1 Evaluation Metrics

Content preservation. We test the capacity of our models to preserve content words from the input by computing a unigram overlap score:

$$overlap(\mathbf{x}, \mathbf{y}) = \frac{|\{\mathbf{y}\} \cap \{\mathbf{x}\}|}{|\{\mathbf{y}\}|} \tag{6.2}$$

between unique unigrams from an input text $\mathbf{x}$ and the generated output rap verse $\mathbf{y}$. We also report the BLEU score (Papineni et al., 2002) when training a model to reconstruct original lyrics. The BLEU score is a metric for evaluating a generated sentence to a reference sentence. A perfect match results in a score of 1.0, whereas a perfect mismatch results in a score of 0.0

Rhyming fluency. We measure the technical quality of our rap verses using the rhyme density (RD) metric Malmi et al. (2016).[7] The metric is based on computing a phonetic transcription of the lyrics and finding the average length of matching vowel sound sequences which resemble multisyllabic assonance rhymes. As a reference, RD values above 1 can be considered high, with some rap artists reaching up to 1.2.

6.4.2 Baselines

For reference, we report the result of an information retrieval baseline, which retrieves the closest text from our training dataset given input from the news or movies test sets, using sentence embedding similarity.[8] We report two variants of the IR baseline. First, we retrieve the closest summary from the CNN/DailyMail news training set

[7] https://github.com/ekQ/raplysaattori

[8] We use a 600-dimensional Sent2Vec model Pagliardini et al. (2018), which is pretrained on Wikipedia.

(IR News), which resembles a lower bound for our target task of style transfer from news to rap lyrics. Second, we retrieve the closest verse from our rap training set (IR Rap). The outputs of the strong IR Rap baseline perfectly match the style of original rap verses, giving us an upper bound for rap style, while maintaining some degree of lexical and semantic overlap with the input texts.

6.4.3 Results

Our results are shown in Table 6.2, where we include all of our stripping approaches (Shuffle, Drop, Replace). We report the results of applying the additional rhyme enhancement step separately (model names ending with "+ RE").

Rap reconstruction. In the left part of Table 6.2, we evaluate our model's capacity to reliably regenerate original rap lyrics given extracted content words from them. Rapformer performed well on this task, generating fluent lyrics that incorporate a large part of the input content words and surpassing the average rhyme density observed in the training dataset (Inputs). When using our rhyme enhancement step, we observe a slight decrease in overlap due to the potential replacement of content words. However, RD increases by 10% on average.

Style transfer. In the right part of Table 6.2, we evaluate the capacity of our model to generate rap lyrics using content words extracted from movie plot summaries or news article summaries. For these inputs, our model generated outputs with lower overlap on average than for rap reconstruction, with movies retaining slightly more content than news. This gap is potentially due to the large differences in style, vocabulary, and topic of the inputs, prompting our models to ignore some of the content words to better match the target rap style. Still, our generation methods

manage to achieve similar RD scores while considerably outperforming the strong IR baseline in terms of overlap.

6.5 Human Evaluation

Due to the limitations of automatic metrics for text generation, we also perform four human evaluation experiments using three raters, who are trained to translate lyrics. Due to limited resources, we evaluate only the RAPFORMER variant with the SHUFFLE stripping approach and rhyme enhancement (SHUFFLE + RE in Table 6.2), which achieved the highest content overlap in our automatic evaluation.

Method	Style	Meaning	Familiarity
IR NEWS	1.18	2.01	1%
IR RAP	4.27	1.33	31%
RAPFORMER	2.03	2.55	8%

Table 6.3: Human evaluation results of RAPFORMER (using the SHUFFLE stripping approach, and news articles as input). The average inter-rater agreement for **Style** is 0.3, and for **Meaning** is −0.1, measured using Cohen's Kappa Cohen (1960b).

The first two human experiments (in Table 6.3) focus on style transfer using news articles as input. Each rater inspected 100 verses produced by either the RAPFORMER, or the two IR baselines, answering the following three questions:

1. *How much do the lyrics presented resemble rap lyrics? On a scale from 1 (not at all), to 5 (this could be from existing rap lyrics)*, which measures the capacity of our models to preserve the **Style**.

2. *How well do the lyrics preserve the content of the original news article on a scale from 1 (not at all) to 5 (very well)?* This question measures the meaning preservation of our models (**Meaning**).

3. *Do these lyrics look like a song you know (yes or no)?* For IR RAP, this question measures the **Familiarity** of the raters with the lyrics; for the other two methods, it measures the capacity to fool the raters.

Method	Side-by-Side	Random
RAPFORMER	7%	25%

Table 6.4: Turing-like evaluation, reporting the percentage of lyrics generated by RAPFORMER (using the SHUFFLE stripping approach, and rap lyrics as input) that human experts incorrectly label as existing rap lyrics. The average inter-rater agreement for **Side-by-Side** is 0.8, and for **Random** is 0.4, measured using Cohen's Kappa Cohen (1960b).

The other two human experiments (in Table 6.4) focus on our rap reconstruction task, performing two Turing-test-like comparisons between 100 real and synthetic verses:

1. **Side-by-Side**: the original rap lyrics and RAPFORMER lyrics are presented side-by-side, in a random order, and a rater is asked, *Which of these lyrics was written by a human?* (see the Appendix for examples).

2. **Random**: a verse is shown and the rater is asked, *"Do you think these rap lyrics are: (a) AI-generated or (b) human-created?"*.

In terms of style (Table 6.3), we outperform IR NEWS, demonstrating that there is a change in style towards rap verses. There is still a large gap to reach the fluency of original rap verses retrieved by IR RAP. However, it is worth noting that the content preservation of IR RAP is considerably lower, as shown in Tables 6.2 and 6.3, and simply the fact that the content of the generated lyrics is closer to the news domain might encourage the raters to rate the generated lyrics as having a lower rap

resemblance score. In other words, the style score of IR RAP might be unrealistic to attain even with a perfect conditional generator.

Overall, the results indicate that our method provides a trade-off between the two baselines in terms of style while outperforming them in terms of content overlap. Furthermore, 8% of the time, our conditional generation model fooled experienced raters to think that our synthetic rap lyrics generated from news articles originate from real rap songs. Our rap lyrics augmentation approach also proved to be robust in the Turing-style evaluation of rap reconstruction (Table 6.4), where RAPFORMER fooled the raters 25% of the time when lyrics from a random source are presented one-by-one, and 7% of the time when lyrics are presented side-by-side.

6.6 Example Model Outputs and Demo Song

In Tables 6.5, 6.6 and 6.7, we also display a few manually selected example model outputs (additional examples are available in the Appendix) produced after inputting content words extracted from each of our input text styles (existing rap lyrics, movie plot summaries and news article summaries). When using existing rap lyrics as input, many outputs seem coherent and of higher quality in comparison to outputs produced using news/movie inputs. For news/movie inputs, the models are still capable of integrating the input content words into a rhyming verse that preserves some of the overall meaning of the original text (e.g., "the film also follows the adventures of lucius the slave escaping via the underground railroad to freedom" $\rightarrow$ "slave, run from lucius slavery; battle of freedom and liberty").

Furthermore, in Table 6.8 we present examples from our side-by-side Turing test, where we asked raters to choose which of the two lyrics was generated (augmented) by RAPFORMER, and which was written by a human. For the selected outputs, two

INPUT (RAP): before the dough came , my whole aim , was blow like propane control the whole domain , and then show no shame make rappers go ? and they so lame , playing with no game put em on the lil plane til they can 't claim no fame i got , the range , better , stay in the slow lane i make the flow change from hurricanes to a slow rain your thoughts are so plane , i rearrange your whole frame until my whole name grow out your brain like rogaine ...
SHUFFLE: aim dough like propane came with a whole blow shame how you control the whole show lame rappers playin' the game, make your domain go can't claim em til you put a lil fame on a plane slow down, stay in your lane, got a better range make it rain, change your flow, slow down rearrange your whole frame, plane thoughts grow like rogaine, put my whole name on your brain ...
SHUFFLE + RE: aim dough like propane came with a whole blow shame how you control the whole show lame rappers playin' the game, make your domain go can't claim em til you put a lil fame on a **show** slow down, stay in your lane, got a better range make it rain, change your flow, slow **rain** rearrange your whole frame, plane **change** grow like rogaine, put my whole name on your brain ...

Table 6.5: Example model output for rap reconstruction. Words replaced by our rhyme enhancement step are in bold. The input lyrics are from the song *How I Get Down* by Rakim.

INPUT (MOVIES): the film follows the lives of several west point cadet classmates who find themselves on opposite sides of the war . the film also follows the adventures of lucius the slave escaping via the underground railroad to freedom with the film cutting between the first battle of bull run and the birth of a lucius ' child born in slavery .
SHUFFLE: this is the opposite of war follows lives on both sides several point film from the west to the wrong find a child born escaping via film film the underground cutting off the film of all the complications slave, run from lucius slavery battle of freedom and birth also the first bull follows luc-up!
SHUFFLE + RE: this is the opposite of war follows lives on both sides several point film from the west to the **light** find a child born escaping via **immigration** film the underground cutting off the film of all the complications slave, run from lucius slavery battle of freedom and **liberty** also the first bull follows luc-up!

Table 6.6: Example model outputs for style transfer from movie plot summaries. Words replaced by our rhyme enhancement step are in bold.

INPUT (NEWS): temperatures dipped into the mid-30s during 4 days man lay in woods of philadelphia park . mom told police son was with her in maryland , but he was found friday with blanket , bible . victim being treated for malnutrition , dehydration ; mother faces host of charges after extradition .
SHUFFLE: man i was dipped up in a lay up with some of them from an old mid-30s days in the park mom told me to be in michigan woods police blanket friday i found my son a bible he was a host for the charges my mother treated him as an age a victim of faces
SHUFFLE + RE: man i was dipped up in a lay up with some of them from an old mid-30s days in the **home** mom told me to be in michigan anyway police blanket **friday** i found my son a bible he was a host for the **trial** my mother treated him as an **alien** a victim of faces

Table 6.7: Example model outputs for style transfer from news articles. Words replaced by our rhyme enhancement step are in bold.

of the three raters incorrectly guessed that the lyrics generated by RAPFORMER were actually human-created.

6.6.1 Demo Song

We generated lyrics for a demo song by using the abstract of this chapter as the input to RAPFORMER. We generated multiple samples, by reshuffling the content words of the abstract multiple times. We sent all sample lyrics to a rap artist, and asked them to record a song using a subset of those lyrics. We allowed for re-arranging and deletion, but no addition of human-created lyrics. The resulting audio file is included in the supplementary material [9], while the final lyrics of the song are in Table C.4 in the Appendices.

We also tested the recently released Jukebox algorithm Dhariwal et al. (2020) for end-to-end synthesis of a rap song conditioned on the abstract content. However, our preliminary results were unsatisfactory since it was impossible to tell individual words apart from the generated audio.

6.7 Related Work

The results in this chapter were informed by the significant recent progress in the natural language processing techniques, in particular, as they related to rap lyrics generation, self-supervision approaches (e.g., autoencoders), and style transfer.

Rap Lyrics Generation Prior work on rap lyrics generation often focuses on unconditional generation, either using language models Potash et al. (2015) or by

[9] A demo, written by RapFormer, performed by PomDP the PhD rapper, is available on soundcloud.com.

stitching together lines from existing rap lyrics using information retrieval methods Malmi et al. (2016). There are two main drawbacks of unconditional generation of rap lyrics. First, the open-ended nature of the task is too unconstrained for generating lyrics with more specific content: ideally, we may want to have control over at least some aspects of the model during inference, such as the topic of the lyrics, or their sentiment. Second, although frequent rhyming is an essential feature of fluent rap verses Malmi et al. (2016), language models have no built-in incentive to learn to consistently generate rhymes at the end of each line, prompting researchers to invent techniques to promote rhyming in their models separately Hopkins and Kiela (2017).

More recently, Manjavacas et al. (2019) propose a conditional approach to rap lyrics generation, which extracts high-level features from the lyrics, such as their sentiment, mood, or tense, to provide a template during generation. Although their approach allows for some control during generation, it is limited in terms of generating lyrics with more specific content. The work that is closest to ours is Lee et al. (2019) who propose an approach to sentence style transfer based on text denoising, and test their approach on style transfer from pop to rap lyrics. In contrast to these works, we condition the model on longer input text and also introduce a novel method for enhancing the rhymes of our output verses. We also perform extensive automatic and human evaluations on style transfer from diverse input domains to rap lyrics.

Text Rewriting and Style Transfer Recent work on style transfer of text Fu et al. (2018); Shen et al. (2017); Prabhumoye et al. (2018); Lample et al. (2019); Liu et al. (2019), focuses on transfer from one text attribute to another, such as gender or political inclination. The main difference between such studies and our work is that our setting is more lenient with respect to meaning preservation: our focus here is on generating creative and fluent verses that match the overall topic of the input

and also preserve *some* of the content. Our conditional lyrics generation based on denoising autoencoders is also related to recent work on self-supervised pre-training objectives for text-to-text generation tasks, which have been beneficial for many NLP tasks, such as automatic text summarization Zhang et al. (2020), question answering Lewis et al. (2020); Raffel et al. (2019), and data-to-text generation Freitag and Roy (2018).

6.8 Future Work

Future work could explore other approaches to extracting content words, including combining several stripping approaches, and could explore the utility of large-scale pretrained models (e.g., (Raffel et al., 2019; Lewis et al., 2020)) for this task. Another direction is to extend our work to end-to-end generation with an integrated rhyming loss function, which could potentially be tackled using reinforcement learning Luo et al. (2019). One might extend the lyric generation of RAPFORMER to include self-play (Ghandeharioun et al., 2019), whereby RAPFORMER iteratively generates lyrics, feeding the output back into itself as input, while controlling for quality, empathy, and the emotional distribution over content. Moreover, the task of generating coherent lyrics from a set of content words could be naturally modeled as a text-editing task Dong et al. (2019); Mallinson et al. (2020); Malmi et al. (2019) instead of a sequence-to-sequence task.

6.9 Chapter Contributions

In this chapter, we propose a novel approach to the generation of rap verses conditioned on a list of content words, showing that our method is capable of generating coherent

and technically fluent synthetic verses using diverse text types as input, including news articles, movie plot summaries, or original rap verses. The fluency of our results is further improved through a novel rhyme enhancement step. Our approach is particularly effective when rephrasing the content of existing rap lyrics in novel ways, making it a potentially useful tool for creative writers wishing to explore alternative expressions of their ideas. The generality of our approach to conditional text generation makes it applicable to the generation of creative texts in other domains, such as poetry or short stories, and in general, for augmenting human capabilities in creative text-based tasks that rely on rhyme and structure.

The contributions of this chapter include:

1. Developed the RAPFORMER method for synthesizing a song verse based on the content of any text (e.g., a news article). This method can also enhance human writing capabilities by augmenting pre-existing song lyrics.

2. Demonstrated that RAPFORMER is capable of generating technically fluent verses in several diverse input domains, while managing a good trade-off between content preservation and style transfer.

3. Evaluated a Turing-test-like experiment, revealing that RAPFORMER fools human lyrics experts 25% of the time.

Question 45 of 100	
LYRICS (A)	LYRICS (B)
waka waka: they say na blind eye, take it far i've got it on my own, my own oche num, oda du, doka dum so if anybody ever try go shoot the almighty blazing so amazing	i say na correct eye i take waka this waka but after i've got you i blind pata pata oche du no dum no oda du num doka anybody try you i go shoot the murderfker ever blazing you amazing
Which of these lyrics was written by a human?	*Correct answer:* (B)
Question 72 of 100	
LYRICS (A)	LYRICS (B)
vegas on the third floor, like lamar with the cardio fascinated by the cars smokin' dope in the casino despise the propaganda rise, higher mac-11 camouflage for example, that's why i never set fires i walk with a flame that never match my desires take a pic, cause the pain is higher i'm rich as a coupe, light it up with kelly phone sucker, my friend, it's a blessing benz, plaques, wall, and g6's - 'em all, hustler say the victim ciroc and bel air - april -'s -, her name so	out in vegas like lamar, third floor tropicana fascinated with the cars, smokin' dope in the phantom teflon's on the rise, i despise propaganda camouflage mac-11, i should set an example never baptized, as i walk through the fires the pain and the flame never match my desires crucified cause i'm rich, in the coupe, take a pic on the phone at the light, kelly rowland's a friend catfish in the benz, manti teo's a sucker plaques on the wall, hustler so i can say "- 'em" bel air for the -, ciroc in the pool my - is a -, her name is april's a fool
Which of these lyrics was written by a human?	*Correct answer:* (B)
Question 74 of 100	
LYRICS (A)	LYRICS (B)
she cut the call when she was on ma phone when you picked up the line you got so mad and asked me who's the girl i'm sleeping with behind baby, i had no words to say so i guess i will try not to lie... it's the time...	i picked up the phone and cut the line and call i asked what's up girl, why you got so long i'm sleeping behind you baby, i guess i try to say the truth but... it's time to lie...
Which of these lyrics was written by a human?	*Correct answer:* (A)

Table 6.8: Examples of lyrics generated by RAPFORMER that fooled the majority (at least two out of three) human raters in a side-by-side comparison with human created lyrics. Inappropriate words are replaced by a single dash.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

Diversifying Learning in Online Forums: Towards Depolarization

"Study hard what interests you the most in the most undisciplined, irreverent and original manner possible."

- Richard Feynman (1965)

Viewing consumption of discussion forums with hundreds or more comments depends on ranking because most users only view top-ranked comments. When comments are ranked by an ordered score (e.g., number of replies or up-votes) without adjusting for semantic similarity of near-ranked comments, top-ranked comments are more likely to emphasize the majority opinion and incur redundancy. This majority bias creates a polarizing "echo chamber," whereby persons sharing the majority opinion find copious reinforcement, and those in the minority are unable to find any reinforcement, even when it exists in large numbers.

In Section 7.2 of this chapter, we propose a top K comment diversification re-ranking model using Maximal Marginal Relevance (MMR) and evaluate its impact in

three categories: (1) semantic diversity, (2) inclusion of the semantics of lower-ranked comments, and (3) redundancy, within the context of a HarvardX course discussion forum. We then conduct a double-blind, small-scale evaluation experiment in Section 7.3, requiring raters to select between the top 5 comments of a diversified ranking and a baseline ranking ordered by score. For three raters, across 100 trials, raters selected the diversified (75% score, 25% diversification) ranking as significantly (1) more diverse, (2) more inclusive, and (3) less redundant. Within each category, inter-rater reliability showed moderate consistency, with typical Cohen-Kappa scores near 0.2. Our findings, discussed in Section 7.3, suggest that our model improves (1) diversification, (2) inclusion, and (3) redundancy, among top K ranked comments in online discussion forums. Code is open-sourced at `https://github.com/cgnorthcutt/forum-diversification`.

Attribution This chapter includes material previously published as (Northcutt et al., 2017a). Kimberly Leon and Naichun Chen contributed significantly to the material presented in this chapter.

Acknowledgements Aspects of the contents of this chapter were shaped by input from Y-Lan Boureau, who contrived the preliminary idea for this work in the domain of social networks and provided mentorship on the development of diversification of semantic content; Regina Barzilay ,who assisted with guidance for model selection and framework; and Karson Ota, who contributed in the early exploration.

7.1 Introduction

Text ranking systems (e.g., Facebook post comments, Amazon product reviews, Reddit forums) are ubiquitous, yet many face a common problem. When posts (e.g., reviews or comments) are ranked primarily by text content and rating (e.g., like/unlike, ↑/↓, +/-, number of replies, etc.), similar posts tend to receive similar scores. Moreover, higher ranking posts tend to exclusively represent the majority opinion, since there are more users in the majority group to upvote posts sharing their sentiments. For large forums with thousands of posts, viewers may only be exposed to the majority opinion when they only view top-ranked posts. If the ground truth semantics of each comment were known, a priori, the comment scores could be normalized by the number of comments with similar semantics, avoiding this problem. Unfortunately, this is not the case. Instead, there are a multitude of techniques to approximate semantic similarity Mikolov et al. (2013); Dumais et al. (1988); Mueller and Thyagarajan (2016).

We consider the comment ranking diversity problem in the context of an online edX course, *Harvardx Christianity Through Its Scriptures*, where increased visibility of the diversity of comments across thousands of learners may aid in debunking the misconceptions held by the majority of forum respondents. edX forums are organized hierarchically into topics > comments > replies (an example topic is depicted in Figure 7-1). Our focus is the ranking of comments and we use the number of replies as the score for each comment, although by default, edX comments are ranked chronologically.

In this chapter, we develop an algorithm for forum comment ranking diversification using maximal marginal relevance (MMR) to linearly interpolate between the original *relevance* ranking score of a comment and the *diversity* of a comment with other

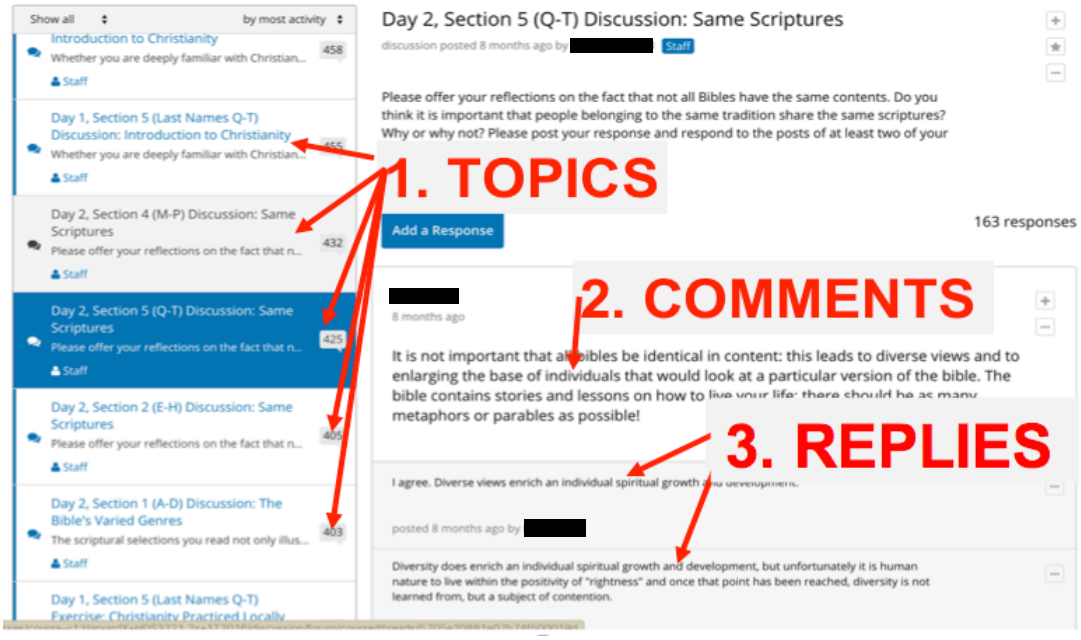


Figure 7-1: An example topic used to illustrate the organization of an edX discussion forum. edX forums are organized hierarchically into topics > comments > replies. Our focus is the ranking of comments.

high-ranked comments. We operationalize our notion of *diversity* using a PCA + TFIDF model on all comments and evaluate our model using a blind experiment requiring subjects to compare our diversified ranking to a baseline relevance ranking.

7.2 Methods

Our methodology consists of four ordered components: (1) Automated generation of gold data, (2) Evaluation of comment embedding models, (3) Implementing

diversification in comment rankings, and (4) Measuring the efficacy of diversification. We describe these components in the following sections.

7.2.1 Dataset

edX forums are organized hierarchically by topic > comments > replies as shown in Figure 7-1. We consider diversification at the comments level (within a single topic). In the context of this study, we focus on the comment rankings for topics in the forum discussions of an edX course, *HarvardX: HDS3221.2x Christianity Through Its Scriptures*, obtained via web-scraping. Comment scores were set equal to the number of replies for each comment. Forum text was tokenized with stop-words removed and over 100,000 comments were analyzed.

Automated Gold Data Generation

We used a novel method to generate large gold datasets without human labeling, by sampling comments across highly differing topics and generating a pairwise cosine similarity matrix for these comments. This matrix contains binary labels: 1 if comments were taken from the same topic, otherwise 0 (comments were taken from different topics). For exclusive sets of topics, we generated both train and test gold datasets to evaluate our selection of different comment embedding models discussed in 7.3.1.

7.2.2 Maximal-Marginal Relevance (MMR)

MMR is an iterative algorithm, at each step selecting the comment which maximizes a modified score (Equation 7.1).

$$\hat{s} := \lambda \cdot s - (1 - \lambda) \cdot c \tag{7.1}$$

A single parameter λ adjusts the trade-off between the original comment score, s, and its maximum cosine similarity among all comments that have already been added to the new ranking, c, to produce the updated score, s'. For example, $\lambda = 1$ ranks entirely by score and $\lambda = 0$ selects maximally diverse comments irrespective of score. In this study, we evaluate two settings of the parameter, $\lambda = 0.75$ and $\lambda = 0.25$ in comparison with a baseline where $\lambda = 1$.

Comment Embedding Model Selection

Diversification using MMR hinges on a comment embedding model that accurately captures the semantic similarity between two comments. Eight models were evaluated (Table 7.4).

Two evaluation metrics were used to compare these models. (1) The median quantile difference defined as the difference in average cosine similarity percentile rank (quantile) of Gold 1 pairs minus that of Gold 0 pairs. We recommend this metric as it is unbiased and captures relative ranking. (2) The accuracy of logistic regression using a given model's pairwise comment cosine similarity matrix as input and the gold binary labels as output. Our two metrics consistently ranked all models.

For the best performing model for these two metrics, comment similarity was computed using cosine similarity Huang (2008). In our case, the best model was PCA + TFIDF comment embeddings, as seen in Table 7.1 in the Results section.

7.2.3 MMR Evaluation Experiment

For our baseline comment ranking, we ordered comments by score with zero diversity ($\lambda = 1$), where the score is the number of replies to each comment. We conducted a small-scale re-ranking evaluation experiment requiring subjects to choose among two unidentified ordered lists of comments: (1) the top 5 comments of our diversified ranking and (2) the top 5 comments of a baseline ranking ordered only by score, their true identities unknown. Three subjects evaluated 100 trials. The Cohen-Kappa score Cohen (1960a) was used to measure inter-rater reliability. For each trial, subjects were presented with three tasks (an example trial is shown in Figure 7-2):

1. The forum's topic question

2. Two lists, A and B. One of these lists is the top five comments ordered by score (baseline). The other is the top five diversified (re-ranked) comments

3. A random comment C from this forum not included in (2) where C's probability of being chosen was proportional to number of replies (higher rank = more likely to be chosen).

Both the order in which lists A and B were shown to subjects and the trial orders were randomized to ensure the true labels for lists A and B were unrecoverable within and across subjects. For each double-blind trial, each subject answered 3 questions:

1. **Inclusion Experiment**: Which list, A or B, has a comment that resembles the semantics of comment C?

2. **Diversity Experiment**: Which list, A or B, best captures a diverse set of all potential answers to this question Q?

3. **Redundancy Experiment**: Which list, A or B, contains more redundant comments?

```
Trial 11

|------------|
| Question Q |
|------------|
'Whether you are deeply familiar with Christianity or new to the tradition, please share 1-3 things...

|--------|
| List A |
|--------|

[1] 'Christianity was explained in just 4 minutes. Very good video.'

[2] "Interesting that so many listed as making up part of Christianity don't recognize each other as such!"

[3] "I appreciated the point about how Eastern Christianity emphasizes the Incarnation...  "

[4] 'Christianity is based not on the writings of Jesus Christ, but the writings of others...'

[5] 'I have always found it interesting that Christianity has both condemned and enabled oppression. '


|--------|
| List B |
|--------|

[1] 'Christianity was explained in just 4 minutes. Very good video.'

[2] 'I found it interesting that some missions working in other cultures recognized the active presence of God...'

[3] "Interesting that so many listed as making up part of Christianity don't recognize each other as such!"

[4] 'Christianity is based not on the writings of Jesus Christ, but the writings of others...'

[5] 'Orthodox is newer to me.   I was raised in a Pentecostal church and converted...  '


|-----------|
| Comment C |
|-----------|
'Two things that impressed me most in the text were:  the history and diversity of Christian movements…'
```

Figure 7-2: Example of a single trial in the MMR evaluation experiment. Each trial was presented to human subjects.

If our comment embedding model accurately captures pairwise semantic similarity, we would expect the diversified ranking to be chosen more often for "inclusion" and "diversity", and less often for "redundancy".

Among the 100 trials for each subject, 75 trials used $\lambda = 0.25$ (ranked more by diversity) and 25 trials used $\lambda = 0.75$ (ranked more by score). More trials were taken for $\lambda = 0.25$ to offset increased stochasticity when selecting low-scored (but diverse) comments. Neglecting the comment score increases variation in ranking. Additional trials mitigated the increased variance.

7.3 Results and Discussion

This section is divided into two parts. Since diversification relies on accurate semantic similarity scores, in Section 7.3.1 we evaluate comment embedding models on our gold dataset. Then, in Section 7.3.2, we evaluate our model in a double-blind subject experiment comparing our diversified ranking against a baseline ranking ordered by score.

Embedding Method	Median Quantile Difference	Logistic Regression Accuracy
TFIDF	0.338	0.841
PCA + TFIDF	**0.434**	**0.867**
LSA + TFIDF	0.431	0.867
NMF + TFIDF	0.416	0.861
LDA + TFIDF	0.129	0.815
Word2Vec + TFIDF	0.205	0.815
Word2Vec + nBOW	0.167	0.815
Gated CNN + TFIDF	0.116	0.786

Table 7.1: Comparison of various comment embedding methods. Median quantile difference computes the difference in average cosine similarity rank (percentile) of Gold 1 pairs - Gold 0 pairs. Logistic regression predicts the accuracy of the gold labels trained using each model's pairwise cosine similarity matrix as input.

7.3.1 Comment Embedding Models

For our task, word-level comment embedding methods (word2vec, Gated CNN, LDA) performed worse than a simple TFIDF vector representation alone, with a classical application of dimensionality reduction using PCA achieving the highest accuracy on our gold dataset. Table 7.1 captures the performance of different embedding models on our gold test set, for both median quantile difference and logistic regression accuracy. In the rest of this section, we discuss potential reasons for this.

Comparing the use of the TFIDF embedding to the use of PCA and LSA affirms that there is a benefit to employing dense embeddings. More unexpectedly, word2vec and Gated CNN, when combined with TFIDF, did not perform as well as TFIDF. A likely suspect is that our word2vec model was trained on the Google News corpus, which is a semantically different and much broader corpus than learner comments in an online course. As a result, word embeddings related to the course content were compressed into a smaller space relative to the broader embeddings of the model.

Given that the comments were on average 78 words in length, and "bag of words" ignores ordering and contextual information, it is less surprising that PCA and LSA outperformed n-BOW and TFIDF models. As PCA offered a marginal performance improvement over LSA, PCA + TFIDF was chosen as our final comment embedding model.

7.3.2 MMR Evaluation

Table 7.2 lists the results of the blind evaluation experiment. The fraction of subject responses selecting the diversified (MMR) ranking is depicted in Figure 7-3. The MMR ranking with $\lambda = 0.75$ (ranked more by score) outperformed the baseline in every experiment (experiments are described in 7.2.3), while rankings with $\lambda = 0.25$ (ranked more by diversity) did not perform significantly better or worse than the baseline.

For moderate diversification ($\lambda = 0.75$), the MMR ranking was chosen significantly more often than the baseline ranking for both diversity and inclusion experiments, and significantly less often than the baseline for the redundancy experiment, suggesting our model mitigates redundancy and majority biases in the top K comments. However, for extreme diversification ($\lambda = 0.25$) the fraction of responses choosing the MMR

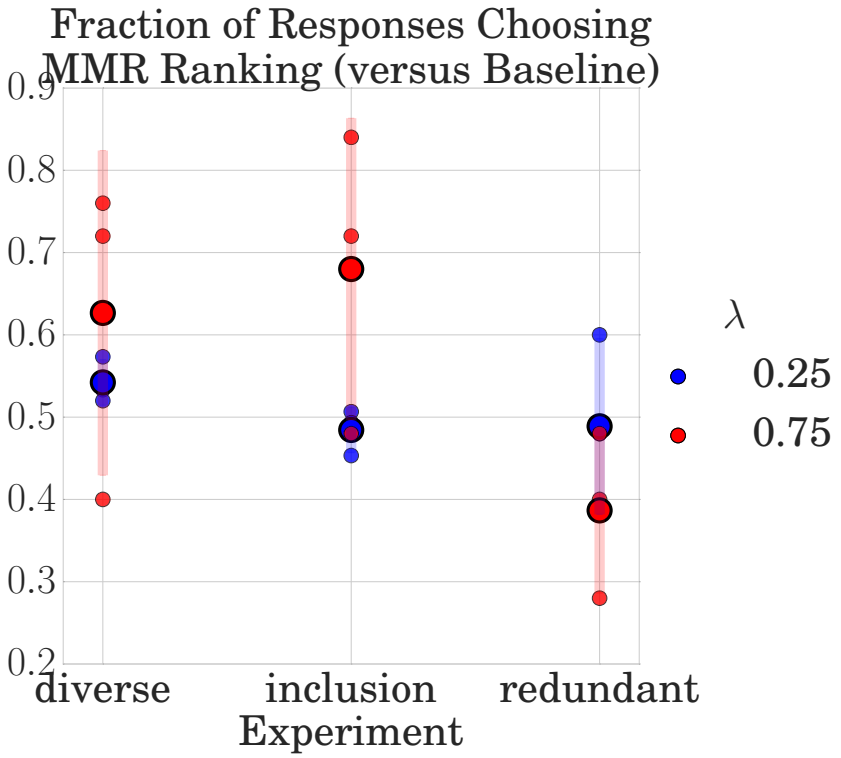


Figure 7-3: Depicts the fraction of trials where raters (on average) chose the diversified (MMR) ranking for each (λ, experiment) pair. Here, $\lambda = 0.25$ implies "more diversity" and $\lambda = 0.25$ implies "better results." Higher values for the "diverse" and "inclusion" experiments and lower values for the "redundant" experiment suggest MMR's efficacy in depolarizing comment rankings. The large, encircled points depict the means of each λ, experiment pair and the translucent bars depict the standard error of each mean. The smaller points depict individual rater scores.

ranking was nearly 0.5 (completely random when compared with the baseline ranking) across all three experiment groups. The cause is likely two-fold. Firstly, ranking correlates with relevance, therefore, replacing more high-ranking comments with diverse, but lower-ranked (and less relevant) comments, may negatively impact all three experiments. Secondly, lower-ranked comments may be off-topic, lower quality,

Score-Diversity Trade-off	Experiment	$\frac{Baseline}{Trials}$	$\frac{MMR}{Trials}$	Trials
$\lambda = 0.25$	inclusion	0.52	0.48	225
	diverse	0.46	0.54	225
	redundant	0.51	0.49	225
$\lambda = 0.75$	inclusion	0.32	0.68	75
	diverse	0.37	0.63	75
	redundant	0.61	0.39	75

Table 7.2: Depicts the aggregated subject counts of the blind evaluation experiment. For each (λ, experiment) group, the number of times either list was chosen is tallied. The two rightmost columns capture the normalized counts. The baseline ranking is generated with MMR and $\lambda = 1$ (ranked only by score).

or harder to parse, leading to a simulated random choice.

Reliability and Agreement Among Test Subjects

Because only three raters were included in our experiment, each evaluating 100 trials, we consider the inter-rater reliability among the three raters to validate the consistency of our findings. Table 7.3 lists the Cohen's Kappa score for all pairs of raters for each experiment group. Although a small number of pairs were inconsistent, most showed moderate consistency.

7.4 Related Work

The crux of diversification is a well-trained comment embedding model that accurately captures the semantic similarity between two documents. Text embedding is a well-studied problem at the word-level Mikolov et al. (2013) and document-level Le and Mikolov (2014). In this section, we consider increasingly complex methods for comment similarity, followed by methods for ranking documents and how it relates to

Experiment	Rater	Other rater 1	Other rater 2
diversity	1	-0.011	0.274
	2	0.179	0.274
	3	-0.011	0.179
inclusion	1	0.034	0.147
	2	0.185	0.147
	3	0.034	0.185
redundancy	1	-0.026	0.136
	2	0.211	0.136
	3	-0.026	0.211

Table 7.3: Cohen's Kappa pairwise inter-rater reliability scores.

diversification.

One of the simplest document embedding representations is TFIDF Wu et al. (2008) which uses a "bag of words" (nBOW) counts model, normalized by word count per document frequency. Although TFIDF works well on some tasks Aizawa (2003), it ignores word ordering and suffers a performance loss for longer documents. TFIDF performs well when combined with matrix decomposition methods like PCA or LSA. More sophisticated approaches such as word2vec Mikolov et al. (2013), LDA Blei et al. (2003), and Gated CNN Lei et al. (2016) offer classification accuracy improvements, but are task-specific. These models are compared in Table 7.4. A state-of-the-art (2016) LSTM similarity model uses a Siamese recurrent architecture to combine the word2vec embeddings of all words in a document and trains using a Manhattan loss on the output of the two LSTMs Mueller and Thyagarajan (2016). Although this method would likely offer improvements, simpler models were sufficient for our task.

The task of forum comment ranking can be thought of as a search task, where common methods like PageRank Page et al. (1999) and RankSVM Duan et al. (2010)) are used to identify the most relevant document for a given query. In our

Table 7.4: A comparison of the comment embedding models evaluated in this study. Method symbols are abbreviated as: T=Topic, M=Matrix Factorization, W=Local Window, F=Frequency, S=Semantic

Model	Method	Scaling Sensitivity
TFIDF	F	False
PCA + TFIDF	M+S	True
LSA + TFIDF	M+S	True
NMF + TFIDF	M+S	True
LDA + TFIDF	T	False
Word2Vec + TFIDF	W+S	False
Word2Vec + nBOW	W+S	False
Gated CNN + TFIDF	W+S	False

case, relevance is determined a priori by comment score, and instead our focus is diversification of this ranking. Diversification has been successfully applied to the task of online shopping Chapelle et al. (2011), with the task of reducing abandonment in shopping queries by providing a diversified selection of options. In this chapter, we elect a more general approach, MMR Carbonell and Goldstein (1998), which we describe in detail in Section 7.2.2.

7.5 Future Work

This chapter establishes baseline results that demonstrate the efficacy of our approach for mitigating redundancy and increasing diversity in a list of comments (scored by the number of upvotes). A natural next step is to implement our diversification approach in a system at scale. We encourage large-scale commenting and human learning platforms, e.g., edX, Coursera, Facebook, Reddit, etc., to consider the implications of upvote-based comment systems, and the "echo chamber" effects that the resulting majority-bias can have on increasing polarization, and to consider the

inclusion of comment ranking diversification for learning systems, i.e., by using the method proposed in this chapter.

7.6 Chapter Contributions

In this chapter, we consider the induced majority-bias (and polarization) in online learning when upvote-based comment rankings are used in discussion forums, particularly due to the large scale of these systems, e.g. massive open online courses, social media, etc. The contributions of this chapter include:

1. Raise attention majority-bias (and polarization) in online learning when upvote-based comment rankings are used in discussion forums.

2. Mitigate the polarizing majority-bias of upvoting-based forums by designing and evaluating a novel comment diversification re-ranking algorithm.

3. Evaluate the experimental evidence of our diversification algorithm, finding a significant increase in diversity and inclusion and decrease in redundancy when our algorithm is used to rank comments versus a baseline relevance ranking.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 8

Answers, and Questions

> *Distill your enthusiasm and hopes into questions. And strive to make these good research questions - ones which are imaginative and inspirational, but still testable and refutable by evidence.*
>
> - Isaac Chuang (Sunday, Apr 7, 2013, 11:46 PM ET)

The route of scientific discovery has no terminus. For if ever there is a day that humanity achieves some conclusive understanding of "all answers," mother nature will reveal to us her simple truth: the answers are always available to us, and they wait patiently for us to find them, in a fleeting thought, in the laws of physics, in an unexpected conversation – often we stare at the answers everyday, but we do not "see" them because, importantly, no answer exists unescorted by a question. And we have yet to ask every question.

The above thinking leads me to forego the usual naming of the final chapter of my thesis as a "Conclusion," and instead, first summarize a narrative journey of answered questions so far, then conclude with open questions, ones which I believe are fundamental to the evolution of machines and humans.

8.1 A Narrative Journey of Questions and Answers

In this section, I share a narrative which summarizes the steps of my doctoral journey along with salient findings. A literal summary of the contributions of this thesis is available in the last section of every chapter.

I spent the first 14 years of my life in semi-rural Kentucky where my father, grandfather, and great-grandfather were mailmen and in her later years, my mother worked minimum wage jobs. I felt disempowered by the lack of opportunities available to me in my youth. I came to view my lack of opportunity as a glass ceiling and the United States education system as a ladder through. By the time I began my doctoral studies at MIT, I was determined to use the opportunity afforded to me by the MIT ecosystem to create machine learning systems that augment human learning and empower others.

To this end, I spent my first three years at MIT from late 2013 to early 2016 designing cheating detection systems to democratize human learning by validating certificates earned by participants of about 300 MITx and HarvardX open online courses (Northcutt et al., 2016; Corrigan-Gibbs et al., 2015; Ho et al., 2015). The cheating strategy exploited the copying of answers via multiple accounts to obtain a certificate, potentially without learning anything in the course. At that time, these courses empowered anyone with an internet connection to attend online courses from MIT and Harvard and earn a certificate of completion.

Working with Issac Chuang and Andrew Ho, we detected thousands of near-certain cheaters using a filtering approach based on hand-designed features and thresholds chosen to minimize false positives (i.e., falsely accusing a participant of cheating). To reduce false negatives, the natural next step was to machine-learn participant features and thresholds directly from real-world, noisy human learning interactions across the

300 courses. Here we met a roadblock: we had no ground truth negative data – how could we guarantee that a student had not cheated by any means including means unbeknownst to us? This led me to ask the following question:

> **Question 8.1**: *How can we quantify the fraction of false positives in a dataset without any ground truth negative labels?*

In the late summer of 2016, in pursuit of the answer to this question, I came across an area of research known as positive-unlabeled (PU) learning, and was particularly inspired by the work of Elkan and Noto (2008). In PU learning, the positive class is always labeled correctly, and the negative class may secretly contain positive examples mislabeled as negative. Elkan and Noto showed that an estimator, *based on the average self-confidence of a classifier*, was provably consistent, i.e., their estimator approached an exact estimation of the fraction of false negatives as the number of training examples increased. Notably, their solution was model-agnostic, meaning it generalized to any domain because it was not limited to a specific modality of data.

I spent the next year from the fall of 2016 to the summer of 2017 working with Tailin Wu and Isaac Chuang to generalize Elkan and Noto's work to binary classification (Northcutt et al., 2017b), and during that time I discovered an insight that permanently reshaped my research direction: nearly all applications of supervised machine learning to augment human capabilities require dealing with noisily labeled data because datasets drawn from human interactions inherit the noisiness of the real world we operate in, e.g., healthcare, education, autonomous transportation, etc. (Northcutt, 2017).

And so in the summer of 2017, I began piecing together confident learning (Chapter 2), a general framework for uncertainty quantification, and weakly supervised machine learning with noisy labels. Joined by Isaac Chuang and Lu Jiang, we developed

methods that work directly with predicted probabilities (model outputs) so that confident learning does not depend on a specific modality of data or model. This feature makes confident learning as applicable for augmenting human capabilities based on medical record text data as it is for autonomous driving image data.

The first salient result surprised the machine learning community: I made the discovery that the MNIST dataset, cited in tens of thousands of publications under the assumption that it is error-free, contains numerous label errors which were identified algorithmically by confident learning. Next, we looked at the ImageNet dataset and found that we could moderately improve test performance, even after removing hundreds of thousands of training examples (detected as potentially noisy by confident learning). To better understand these experiments, I undertook an analysis of the theoretical conditions necessary for exact uncertainty quantification (i.e., label noise estimation), and proved that confident learning can exactly identify label errors in conditions where every predicted probability for every class for every example contains a bounded amount of error. Inspired by the theoretical support of confident learning methods, I open-sourced the `cleanlab` package to democratize access to machine learning with noisy labels and finding label errors in datasets.

In the summer of 2018, sometime after discovering numerous errors in MNIST, I had a chance encounter with a good friend and exquisite thinker, Anish Athalye, at the International Conference of Machine Learning (ICML) in Stockholm, Sweden. Motivated by Anish's ICML results in breaking several defenses of adversarial examples in machine learning, and the pervasive label errors I had found in machine learning datasets, we discussed the following questions:

> **Question 8.2**: *Is the* data *used in the field of machine learning* broken*? Are machine learning scientists unknowingly benchmarking the progress*

of machine learning models based on erroneous datasets? Does it matter?

We surmised that confident learning could help to quantify the answer to this question, leading to the results in Chapter 3, where we focus on label errors in test sets to understand how they might affect benchmark stability and to distinguish the work from that in Chapter 2, which focuses on label errors in training data. To our surprise, label errors were prevalent across ten of the most commonly cited test sets used to benchmark machine learning models. Even further, small increases (6%) in test set noise prevalence destabilize benchmarks in datasets like ImageNet and CIFAR-10, resulting in simpler models like ResNet-18 and VGG-11 outperforming their vastly larger and more complex sisters, NasNet and VGG-19, respectively. These results clarified that label errors in both train and test sets are more problematic than previously believed and confident learning is well-suited to address these problems for real world noisy datasets.

Equipped with a new understanding of uncertainty quantification from Chapters 2 and 3, I return to my goal to empower humans with augmented capabilities and confidence in noisy, real world settings in Chapters 5, 6, and 7.

Starting in the late spring of 2018, I spent two years working with Richard Newcombe and Steven Lovegrove at Oculus Research (later renamed Facebook Reality Labs). Inspired by how humans evolve through sensory inputs from our embodied egocentric perspectives, we asked the following two questions:

> **Question 8.3**: *How can we make the data inputs used to train artificially intelligent machines more similar to the sensory inputs used in the development of human intelligence? And how can such* egocentric *data be exploited to augment human capabilities?*

We answer these questions in Chapter 5, where we create and release the first multi-person conversational dataset comprised of embodied video streams (video captured from the perspective of the wearer's eyes, audio from the perspective of the wearer's ears) that are time-synchronized across all participants. We show how synchronous, multi-perspective egocentric data enables a simple solution to combine noisy signals (based on which transcription has the highest confidence at each time step) that improves a state-of-the-art speech-to-text system by 79% when compared to asynchronous, single-perspective transcription. A salient result in Chapter 5 is captured by Figure 5-7, whereby a machine's accuracy at predicting turn-taking dynamics in human conversations is shown to increase when the machine is trained with synchronized noisy signals from all participants' perspectives versus only one person's perspective, confirming our other findings with transcription.

Stepping briefly back in time, in the summer of 2015, I had the pleasure of working with Eric Malmi, not as a researcher, but as a rapper. Eric had developed an approach to combine stanzas from previously written rap song lyrics to create some of the first rap songs constructed by artificial intelligence, one of which I performed, entitled "The Machine's Turn."

Working with Eric again in the summer of 2019, and joined by Nikola Nikolov and Loreto Parisi, we embarked to answer a series of more challenging questions to understand a machine's ability to generate song lyrics (Chapter 6):

> **Question 8.4**: *Can a machine generate song lyrics based on a news article? If so, could that machine be used to* improve *previously written song lyrics? Can it perform well enough to fool humans in a Turing test?*

We answer these questions in Chapter 6, where we assemble a denoising auto-encoder approach for synthesizing a song verse based on the content of any text (e.g., a

news article). On inspection, our method appears to generate technically fluent verses in several diverse input domains, while managing a good trade-off between content preservation and style transfer. However, what we were most interested in is whether our approach could be used to augment human writing capabilities by enhancing the rhyme density of pre-existing song lyrics while remaining indistinguishable from human-generated lyrics (from the perspective of an expert reviewer of lyrical writing). We conducted a Turing-test-like experiment, where a verse is shown to an expert rater who is asked, *"Do you think these rap lyrics are: (a) AI-generated or (b) human-created?"*, and found that on average over a set of 100 song lyrics our method fools human experts 25% of the time.

Chapter 7 refocuses this thesis on my manifesto (see Sec. 1.6) to empower humans. The results in this chapter follow from ideas I learned in Yann LeCun's group at Facebook AI Research (FAIR) in New York City, where I spent the summer of 2016 exploring diversification algorithms under the guidance of French computer scientist, Y-Lan Boureau. Y-Lan noted that online forums are prone to majority-bias in comment rankings due to the nature of upvoting with thousands of participants and proposed the MMR diversification algorithm to mitigate majority-bias. Upon returning to MIT in the fall, I re-purposed this idea to enable humans to learn more confidently in noisy learning environments. Although the majority of the research was conducted in 2016, I present this work as the last chapter in the thesis because it supremely embodies confident learning for humans.

In the fall of 2016, joined by Kimberly Leon and Naichun Chen, we asked the following question:

> **Question 8.5**: *How can we design an upvote-ranked online discussion forum, comprising thousands of comments from human learners, to be* less

biased *in favor of comments written (and upvoted) by learners belonging to the majority opinion?*

We address this question in Chapter 7 with a simple diversification algorithm that re-ranks comments based on a trade-off of relevance (number of upvotes) versus diversity (semantic orthogonality to currently top-ranked comments). Ranking is important because there can be tens of thousands of comments – far too many to scroll through. By diversifying comment rankings, we discourage polarization, which occurs when the majority opinion is constantly reinforced because there are more people to upvote comments. By placing high-scoring comments which differ from the majority opinion at the top of the discussion forum, we enable *confident learning for humans* who belong to the minority group by increasing the likelihood they see ideas similar to their own (more so than a traditional upvote-based comment ranking system would lead them to believe). A salient result in Chapter 7 is captured in Figure 7-3. Three human reviewers were provided a blind comparison of upvote-based ranking versus our diversified ranking. On overage, humans chose our ranking as significantly more diverse, more inclusive, and less redundant than the upvote-based ranking.

8.2 Open Questions

Having summarized the questions addressed in this thesis, I conclude with open questions on learning with confidence and societal implications as they relate to the intersection of machines and humans.

8.2.1 IA-AI Learning

Intelligence augmentation (IA) is intrinsically linked to artificial intelligence (AI) (Carter and Nielsen, 2017). For example, large technology companies (e.g., Google, Facebook, and Microsoft) typically have an *infrastructure organization* for developing internal tools to improve software workflows for employees. These tools might include code highlighting, continuous integration and deployment software, and code review task software. Often these tools use artificial intelligence (AI) for predictive analytics, sentence completion, code similarity checks, etc. Software engineers use these AI-enabled tools for intelligence augmentation (IA): call this the $AI \rightarrow IA$ step.

Infrastructure engineers use their augmented intelligence to build more sophisticated artificial intelligence for the next generation of infrastructure tools: call this the $IA \rightarrow IA$ step. Enhanced by IA, infrastructure engineers build better AI, which augments their intelligence, so they can build better AI, and so forth.

I call this feedback loop *"IA-AI learning,"* and the synergy of AI and IA can take many forms, from loosely linked systems like the one described above to tightly coupled systems like brain-computer interfaces. While IA-AI learning has enormous potential to accelerate the evolution of the human species, it poses numerous concerns, such as increased bipolarization of income, opportunity, and intelligence (Chalmers, 2009). With this in mind, I pose the following open question:

> **Question 8.6**: *How can we ethically design IA-AI learning systems to democratically empower humanity with augmented capabilities?*

This question extends Chapters 5 and 7.

8.2.2 Polarization versus Personalization

Personalized recommendations are increasingly pervasive throughout our daily lives. From the moment we wake up and read our personalized news feed, and throughout the day as we search the internet, until the night when we watch a recommended movie on a streaming platform before bed. This seemingly innocuous personalization of content is profitable for technology companies because users are more likely to purchase things they know they already like.

A simple approach to build a recommendation system is to sort items by their number of upvotes. We saw an example of how this creates a polarizing majority bias in Chapter 7 and looked at diversification methods to mitigate this bias. However, our method provides the same diversified ranking to all users, not a personalized ranking for each user. This ensures that everyone has an equal opportunity for exposure to the same content.

More sophisticated methods for personalization in recommendation systems typically learn an embedding/vector representation of users (or other content like movies or news articles) to *divide* people in a n-dimensional space based on their preferences (Cheng et al., 2016). Some clustering methods (e.g., K-means) maximize *dispersion*, i.e. directly maximizing the distance between the means of clusters (groups of people).

An approach like this has been used by Google News (Das et al., 2007), a primary source of political information. If Republicans primarily see pro-Republican news and Democrats primarily see pro-Democrat news, intensifying political polarization, how can a person from Kentucky (largely Republican) empathize with the viewpoints of a person from Massachusetts (largely Democratic). This form of polarization generalizes to other recommendation systems as well: if you tend to watch *action*

movies, you will be recommended more *action* movies, making it potentially more difficult to empathize with a co-worker who loves *anime* if you have never previously been exposed to that kind of content.

We can think of the polarization/personalization trade-off much like the exploitation/exploration trade-off in reinforcement learning. We can recommend a news article to a user based on what we think they already like (exploitation), but if we expose the user to new interests (exploration), over time they may find something they like even more.

With this in mind, I pose the following open questions:

> **Question 8.7**: *How does continual personalization of information consumption affect the empathy between people from different backgrounds over time? Is it possible to have personalized recommendation systems that do* not *polarize groups of people with different interests? If so, how can we build such systems? How does the length of time that a person has been exposed to personalized recommendations affect their ability to empathize with others? If empathy is measured by the overlap of interests between two people, which systems (e.g., news feeds, media, etc.) favor empathy? Which systems favor revenue at the expensive of empathy? By how much?*

These questions extend Chapter 7.

Real-world applications of machine learning are increasingly personalizing the information we see, which decreases exposure and empathy to different views other than our own, increasing polarization. Regardless of the profitability of personalization/polarization, we must develop ways for *machines* to help support the

increase of empathy between *humans*. This is one of the most critical open questions for the future of machine intelligence and human emotional intelligence.

8.2.3 "Close Enough" Learning

Traditional notions of accuracy for single-class classification tasks require exact targets. For example, an image containing a projectile must be labeled either "missile" or "projectile," and only one such label is considered correct. Yet, often this constraint is impractical or unnecessary. For example, in the human-centric classification problem of emotion understanding, for many practical purposes, predicting "worried" or "concerned" are both "close enough." Consider a new target representation where instead of a single label for each example, we provide an unnormalized distribution over classes. For example, consider classifying images among the classes "dog," "wolf," and "tree." Instead of labeling an image of a dog as ["dog": 1, "wolf": 0, "tree": 0], we could label the image of the dog as ["dog": 1, "wolf": 0.8, "tree": 0] because $\sim 80\%$ of the time, guessing "wolf" instead of "dog" is "close enough."

Close enough learning extends traditional machine learning to be more data-centric by taking into account the the inherent ambiguity/uncertainty among classes in the dataset while learning. Tens of thousands of images in the popular ImageNet dataset suffer from ambiguity among classes (Tsipras et al., 2020; Northcutt et al., 2021a). For example, "bathtub" and "tub", "missile" and "projectile", "green lizard" and "chameleon" are all pairs of classes in ImageNet. For most practical purposes, guessing either would be "close enough".

To operationalize close enough learning, the loss function for both training and evaluation should allow for an unnormalized distribution over classes based on the similarity of an example's given class label with all other classes (instead of the

traditional one-hot encoded representation). Following the results of Chapter 2, we can estimate this distribution directly by estimating the joint distribution of noisy labels and true labels using confident learning. Then for any class, the target distribution should be the column of the joint matrix corresponding to that class, along with a softmax temperature scaling hyper-parameter to properly calibrate *how close* is "close enough." As an alternative approach, one can also cross-correlate the softmax outputs of all pairs of classes to build a similarity matrix (Felbo et al., 2017).

The intuition behind close-enough learning comes from human learning. In many learning environments, we receive partial credit when our guess is *close* to the presumed-correct answer, even if the answer is slightly wrong: a teacher is not limited to assigning 100% or 0% scores for feedback. These non-binary feedback scores (e.g., percentage grades) are used for both learning/training and evaluating/testing.

With this in mind, I pose the final open question:

> **Question 8.8**: *Can learning with distributions over target classes improve machine learning on datasets which contain inherently ambiguous or overlapping classes? Can this* close enough learning *paradigm be used for evaluation (as opposed to top-k accuracy) to more accurately rank machine models similar to how a human would rank them?*

These questions extend Chapters 2 and 3.

8.3 The Destination

Having seen a narrative journey of this thesis, what is the intended destination? Why is this thesis called "confident learning for machines **and humans**" and not just "confident learning for machines?"

Among all fields of science, computer science is markedly *humanistic*. Physics, biology, and chemistry are given to us by nature. We do not invent new forms of these sciences – we discover them. We ask better questions until nature unveils new answers. Unlike other sciences, computer science is humanity's creation, an offshoot of our intellectual creativity. Although computer science adheres to mathematical abstraction, which can be viewed as axiomatic based on the ordering and identity proprieties of the universe, the motivations and applications of computer science are largely human-inspired and human-designed. Computers/machines share memories, connect people, and affect how humans feel (e.g., watching a video, reading a message, or viewing the news) – these are fundamentally human constructs.

The open questions posed in this chapter are a call-to-action to employ the humanistic nature of computer science and artificial intelligence to empower humanity. This is our destination. From the start of this thesis in Section 1.6, my manifesto is to augment human intelligence with artificial intelligence, *to empower people with machines*. Questions 8.1 - 8.2 address the preliminary first steps necessary to pursue my manifesto by enabling machines to learn and perform amidst the uncertainty inherent in humanistic data. Questions 8.3 - 8.5 take secondary steps toward my manifesto by exploring small-scale artificially intelligent applications that augment human capabilities amidst inherent label uncertainty. Open Questions 8.6 - 8.8 aspire to leap: to understand how we can empower all humans equitably, at a societal scale, with machines that learn and perform confidently despite noisy human data. How we choose to answer questions like these will influence whether machines will ultimately empower or disempower humanity at scale.

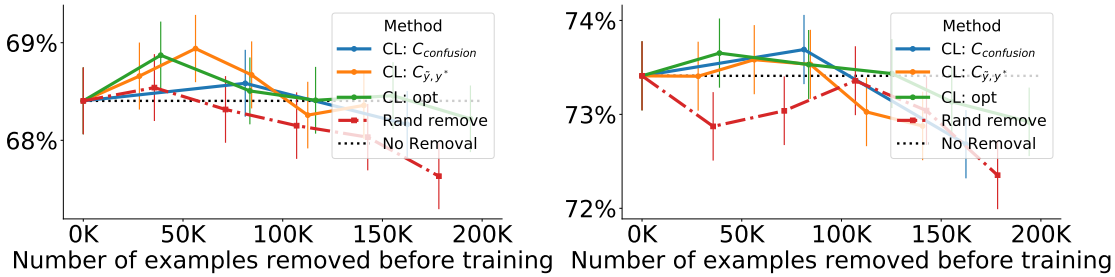
Appendix A

Additional Figures and Tables for Chapter 2

A.1 Figures

In this section, we include additional figures that support Chapter 2. Fig. A-1 explores the benchmark accuracy of the individual confident learning approaches to support Fig. 2-6 and Fig. 2-5 in the main text. The noise matrices shown in Fig. A-2 were used to generate the synthetic noisy labels for the results in Tables 2.3 and 2.1.

Fig. A-1 shows the top-1 accuracy on the ILSVRC validation set when removing label errors estimated by CL methods versus removing random examples. For each CL method, we plot the accuracy of training with 20%, 40%,..., 100% of the estimated label errors removed, omitting points beyond 200k.



(a) ResNet18 Validation Accuracy (b) ResNet50 Validation Accuracy

Figure A-1: Increased ResNet validation accuracy using CL methods on ImageNet with original labels (no synthetic noise added). Each point on the line for each method, from left to right, depicts the accuracy of training with 20%, 40%..., 100% of estimated label errors removed. Error bars are estimated with Clopper-Pearson 95% confidence intervals. The red dash-dotted baseline captures when examples are removed uniformly randomly. The black dotted line depicts accuracy when training with all examples.

A.2 Tables

We benchmarked INCV using the official Github code[1] on a machine with 128 GB of RAM and 4 RTX 2080 ti GPUs. Due to memory leak issues (as of the February 2020 open-source release, tested on a MacOS laptop with 16GB RAM and Ubuntu 18.04 LTS Linux server 128GB RAM) in the implementation, training frequently stopped due to out-of-memory errors. For fair comparison, we restarted INCV training until all models completed at least 90 training epochs. For each experiment, Table A.1 shows the total time required for training, epochs completed, and the associated accuracies. As shown in the table, the training time for INCV may take over 20 hours because the approach requires iterative retraining. For comparison, CL takes less than three hours on the same machine: an hour for cross-validation, less than a minute to find errors, an hour to retrain.

[1] https://github.com/chenpf1025/noisy_label_understanding_utilizing

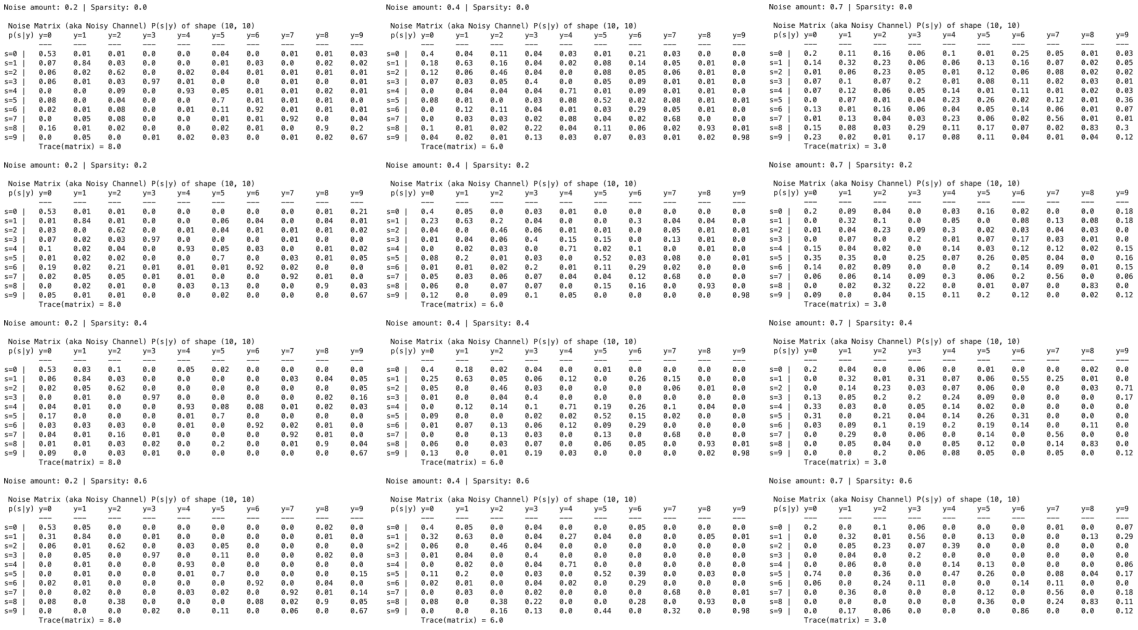


Figure A-2: The CIFAR-10 noise transition matrices used to create the synthetic label errors. In the `cleanlab` code base, s is used in place of $\tilde{y}$ to notate the noisy unobserved labels and y is used in place of y^* to notate the latent uncorrupted labels.

Table A.1: Information about INCV benchmarks including accuracy, time, and epochs trained for various noise and sparsity settings.

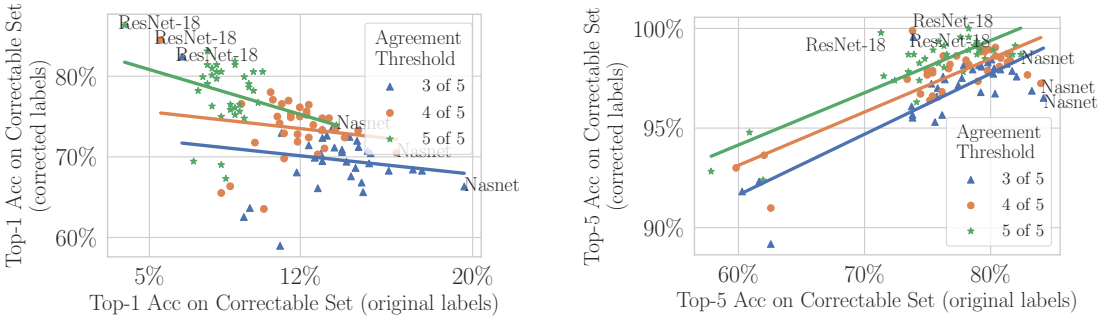
Noise	0.2				0.4				0.7			
Sparsity	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.6
Accuracy	0.878	0.886	0.896	0.892	0.844	0.766	0.854	0.736	0.283	0.253	0.348	0.297
Time (hours)	9.120	11.350	10.420	7.220	7.580	11.720	20.420	6.180	16.230	17.250	16.880	18.300
Epochs trained	91	91	200	157	91	200	200	139	92	92	118	200

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B

Additional Figures and Tables for Chapter 3

B.1 Figures



(a) Top-1 Accuracy.

(b) Top-5 Accuracy.

Figure B-1: 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 (⋆).

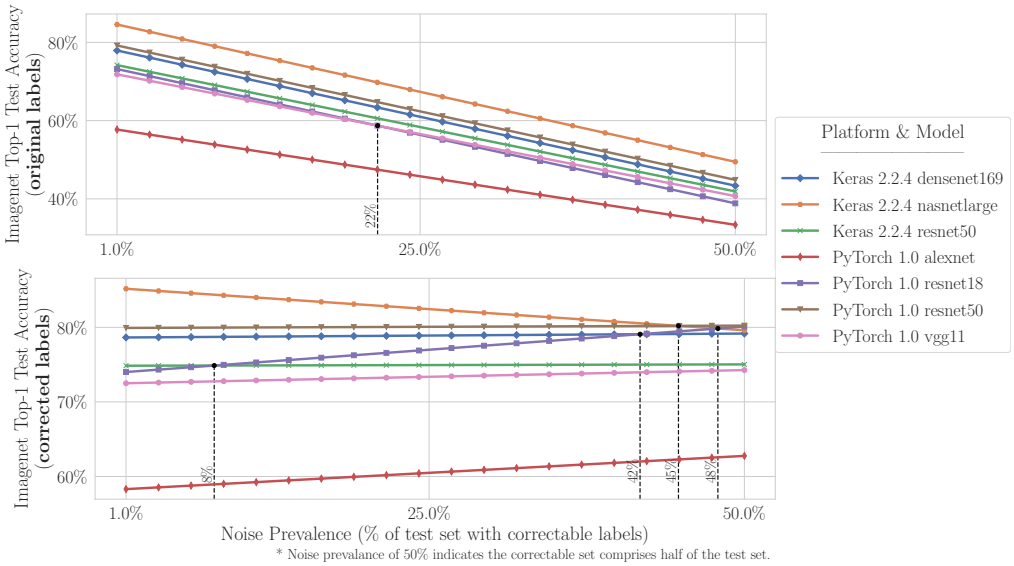


Figure B-2: ImageNet top-1 original accuracy (top panel) and top-1 corrected accuracy (bottom panel) vs Noise Prevalence with agreement threshold = 5 (instead of threshold = 3, c.f., Fig. 3-4).

B.2 Tables

Figure 3-3 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 B-1b. 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 B-1a), 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 B.1).

The dramatic changes in ranking shown in Table B.1 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. These results also suggest that keeping some correct labels on a secret correctable set of label errors may provide a useful framework for detecting overfitting on test sets toward a more reliable approach for benchmarking generalization accuracy across ML models.

The benchmarking experiment was replicated on CIFAR-10 in addition to ImageNet. The individual accuracies for CIFAR-10 are reported in Table B.2. Similar to ImageNet, smaller capacity models tend to outperform higher capacity models when benchmarked using corrected labels (instead of the original, erroneous labels).

Whereas traditional notions of benchmarking generalization accuracy assume the train and test distributions are the same, this is nonsensical in the case of noisy training data — the test dataset should never contain noise because in real-world applications, we want a trained model to predict the error-free outputs on unseen examples, and benchmarking should measure as such. In two independent experiments in ImageNet and CIFAR-10, we observe that models, pre-trained on the original (noisy) datasets, with less expressibility (e.g., ResNet-18) tend to outperform higher capacity models (e.g., NasNet) on the corrected test set labels.

Table B.1: Individual accuracy scores for Sub-figure 3-3b 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 B.2: Individual CIFAR-10 accuracy scores for Sub-figure 3-3c 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 B.1 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

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix C

Additional Example Outputs for Chapter 6

In Tables C.1, C.2 and C.3 we display a few additional manually selected model outputs for each of our input domains (rap lyrics, movie summaries and news article summaries) and each of our stripping approaches (Shuffle (Rapformer), Drop, and Synonym).

INPUT: control the whole domain , and then show no shame make rappers go ? and they so lame , playing with no game put em on the lil plane til they can 't claim no fame i got , the range , better , stay in the slow lane i make the flow change from hurricanes to a slow rain your thoughts are so plane , i rearrange your whole frame until my whole name grow out your brain like rogaine ...
DROP + RE: like propane in control, my whole objective show no shame, i can make it playin' the game put em in a lil' game, til they don't claim fame range you better stay in your lane, and make the flow change i be on a plane to rearrange your own frame your whole name grow in my brain, rogaine letter
SYNONYM + RE: control the whole domain just to show it no shame make these rappers go lame, they playin' the game put em on a lil' plane til they can't title the fame if you got a range, you better stay in a slow lane name run change hurricane slow rain thoughts on the ground and rearrange your whole frame whole name grow brain like a rogaine cane
SHUFFLE + RE (RAPFORMER): aim dough like propane came with a whole blow shame how you control the whole show lame rappers playin' the game, make your domain go can't claim em til you put a lil fame on a show slow down, stay in your lane, got a better range make it rain, change your flow, slow rain rearrange your whole frame, plane change grow like rogaine, put my whole name on your brain ...
INPUT: now comes our turn we step in, you burn you got something to learn and bring a breath of fresh air this is how we go you check this mode ride da best you know and still want more
DROP + RE: when it comes to turn to step i got something to bring in my breath go check my mode ride with da so you know i still want to be your own
SYNONYM + RE: when it comes to turn my step to burn i got something to learn bring your breath to new light go check the way i ride da best i know you still want so
SHUFFLE + RE (RAPFORMER): when it comes to my turn step up and learn something to burn i got fresh air on my breath go check da best mode, bring it yeah you know i still want a ride

Table C.1: Additional model outputs for rap reconstruction.

INPUT: hoping to improve his financial lot , petty thief hawk chovinski hires a dancing instructor to teach him how to bear himself like a gentleman . his lessons completed , hawk then poses as a european nobleman , intending to trap a wealthy wife . yolande cowles sees through hawk ' s pose but falls in love with him anyway .
DROP + RE: i improve a grizzly lot of petty thief times dancing in the middle of the night i am the man who can teach you how to bear it like a gentleman with diamonds i'm a superheroic, i can be your wife yolande cowles tonight falls in love anyway
SYNONYM + RE: hoping that you can improve a financial lot of petty use mortarboard chovinski engage dancing with the snake teach her how to settle like a gentleman lessons are shackled by a bullet sit in european imagine in the trap with a wealthy wife yolande hood sees the sky when the pose falls in line anyway, no, not me
SHUFFLE + RE (RAPFORMER): you teach me petty dancing like bear thief chovinski, intersect, be a lot of financial gentleman hoping he can improve somebody wife, nobleman, the trap is so polished wealthy hawk lessons, european hawk lessons yolande cowles anyway, sees him pose when he says hawk love!
INPUT: the film follows the lives of several west point cadet classmates who find themselves on opposite sides of the war . the film also follows the adventures of lucius the slave escaping via the underground railroad to freedom with the film cutting between the first battle of bull run and the birth of a lucius ' child born in slavery .
DROP + RE: film of the west point where they can find the opposite sides of ours film also and they will be a slave escaping me from the underground, and we will not be the same if we are not the maker this is a film cutting first bull from birth to child's slaver.
SYNONYM + RE: film to succeed our lives in several zones our head is the most likely to find our own we are not the same as the other side of ever film also follows adventure the lucius slave, the escaping via underground motorical, freedom, film out first battle bull, then feed him birth golden child, born in order
SHUFFLE + RE (RAPFORMER): this is the opposite of war follows lives on both sides several point film from the west to the light find a child born escaping via immigration film the underground cutting off the film of all the complications slave, run from lucius slavery battle of freedom and liberty also the first bull follows luc-up!

Table C.2: Additional model outputs for style transfer from movie plot summaries to rap lyrics.

INPUT (NEWS): temperatures dipped into the mid-30s during 4 days man lay in woods of philadelphia park . mom told police son was with her in maryland , but he was found friday with blanket , bible . victim being treated for malnutrition , dehydration ; mother faces host of charges after extradition .
DROP + RE: i've been dipped for days, lay in woods in the park with the crook my son from pittsburgh found friday blanket, bible victim i was born to be a victim of my reality with no faces host charges, i had to do it everything
SYNONYM + RE: dipped in mid-a.t. days, man i dwell in ways UNK 3: gs, park mom told the police son that he was from illegal found him on friday, he had a blanket with a bible a victim of how he treated him as a
SHUFFLE + RE (RAPFORMER): man i was dipped up in a lay up with some of them from an old mid-30s days in the home mom told me to be in michigan anyway police blanket friday i found my son a bible he was a host for the trial my mother treated him as an alien a victim of faces
INPUT (NEWS): jordan spieth won the 2015 masters by four shots on sunday . the 21-year-old american led all week at the augusta national golf club . he shot final-round 70 to finish on 18 under par and take the green jacket .
DROP + RE: to all of my masters, four sunday morning, american led alert national golf club, final-round time take a green jacket
SYNONYM + RE: jordan, we are not the same, no masters! four shots of the sun, the laughter we were the most likely american led in a week at the first club shot last finish, hey get the green cap
SHUFFLE + RE (RAPFORMER): masters, four shots on sunday jordan, led me to the national club, the american way golf week, green dine, par finish my jacket, take my final-round start

Table C.3: Additional model outputs for style transfer from news articles to rap lyrics.

[intro]
i am the oldest
the lyrics they just follow orders.
i am the oldest
the lyrics they just follow orders.
good trade-off of your style.
i am the oldest
the lyrics they just follow orders.
i rhyme more rhymes and moreover
move over I'm recording

[verse 1]
another verse written on the news of rap methods,
given to me in the form of an autoencoder
to develop the words that i rap, and i will be denoting
in my text, i am the only content,
i can be the same as an automatist,
i train rap lyrics to study different meaning when i approach words as i am,
I train lyrics that are the most definitive,
more essential than a scheme of three
more untouchable than an underflow
move over. pirana, the founder, moreover.
my rhyme lyrics are more than the rhyme over
(when i develop a verse)

[verse 2]
when i develop a verse i form a text from an art that is written on the news of an autoencoder rap
another method given to a train that i have been through and i am not the only thing to do with
this is my reality
i will not be content with rap lyrics i approach with the meaning oh
my words are based on my attack.
my lyrics are essential as I generate rap.
my average rhyme scheme is to show you different content
in other words, i can't study my own admirations.
my raps are so amazing
the rhyme is paraphrasing.

[bridge]
my results are very good like I'm a human being
my rap is in the convoy.
your lyrics will be so pre-dated.
(when i develop a verse)

[outro]
I'm a human being
I'm a human being

Table C.4: Lyrics of our demo song, described in Appendix 6.6.1.

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Amjad, M., Shah, D., and Shen, D. (2018). Robust synthetic control. *Journal of Machine Learning Research (JMLR)*, 19(1):802–852.

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

Arandjelovic, R. and Zisserman, A. (2018). Objects that sound. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 435–451.

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., and McGuinness, K. (2019). Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning (ICML)*.

Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. (2017). A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. Proceedings of Machine Learning Research (PMLR).

Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (2019). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443.

Baltrušaitis, T., Robinson, P., and Morency, L.-P. (2016). Openface: an open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10. IEEE.

Bamman, D., O'Connor, B., and Smith, N. A. (2013). Learning latent personas of film characters. In *Annual Conference of the Association for Computational Linguistics (ACL)*, pages 352–361.

Beigman, E. and Klebanov, B. B. (2009). Learning with annotation noise. In *Annual Conference of the Association for Computational Linguistics (ACL)*.

Berthon, A., Han, B., Niu, G., Liu, T., and Sugiyama, M. (2020). Confidence scores make instance-dependent label-noise learning possible. *arXiv preprint arXiv:2001.03772*.

Beyer, L., Hénaff, O. J., Kolesnikov, A., Zhai, X., and van den Oord, A. (2020). Are we done with imagenet? *arXiv preprint arXiv:2006.07159*.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bouguelia, M.-R., Nowaczyk, S., Santosh, K., and Verikas, A. (2018). Agreeing to disagree: active learning with noisy labels without crowdsourcing. *International Journal of Machine Learning and Cybernetics*, 9(8):1307–1319.

Bradley, A. (2017). *Book of rhymes: The poetics of hip hop*. Civitas Books.

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

Carbonell, J. and Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336.

Carter, S. and Nielsen, M. (2017). Using artificial intelligence to augment human intelligence. *Distill*, 2(12):e9.

Chalmers, D. J. (2009). The singularity: A philosophical analysis. *Science Fiction and Philosophy*, pages 171–228.

Chapelle, O., Ji, S., Liao, C., Velipasaoglu, E., Lai, L., and Wu, S.-L. (2011). Intent-based diversification of web search results: metrics and algorithms. *Information Retrieval*, 14(6):572–592.

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

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10.

Cheng, J., Liu, T., Ramamohanarao, K., and Tao, D. (2020). Learning with bounded instance and label-dependent label noise. In III, H. D. and Singh, A., editors, *International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 1789–1799. PMLR.

Chiu, C.-C., Sainath, T., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R. J., Rao, K., Gonina, K., Jaitly, N., Li, B., Chorowski, J., and Bacchiani, M. (2018). State-of-the-art speech recognition with sequence-to-sequence models.

Chowdhary, K. and Dupuis, P. (2013). Distinguishing and integrating aleatoric and epistemic variation in uncertainty quantification. *Mathematical Modelling and Numerical Analysis (ESAIM)*, 47(3):635–662.

Chung, J. S., Senior, A., Vinyals, O., and Zisserman, A. (2017). Lip reading sentences in the wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3444–3453. IEEE.

Cohen, J. (1960a). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37.

Cohen, J. (1960b). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Cordeiro, F. R. and Carneiro, G. (2020). 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)*, pages 9–16.

Corrigan-Gibbs, H., Gupta, N., Northcutt, C., Cutrell, E., and Thies, W. (2015). Deterring cheating in online environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 22(6):1–23.

Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., and Wray, M. (2018). Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*.

Das, A. S., Datar, M., Garg, A., and Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web*, pages 271–280.

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., and Liu, R. (2020). Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Dawid, A. P. and Skene, A. M. (1979). 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.

del Molino, A. G., Tan, C., Lim, J.-H., and Tan, A.-H. (2017). Summarization of egocentric videos: A comprehensive survey. *IEEE Transactions on Human-Machine Systems*, 47(1):65–76.

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

DeVault, D., Mell, J., and Gratch, J. (2015). Toward natural turn-taking in a virtual human negotiation agent. In *2015 AAAI Spring Symposium Series*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186.

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., and Sutskever, I. (2020). Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*.

Dobbinson, S., Perkins, M. R., and Boucher, J. (1998). Structural patterns in conversations with a woman who has autism. *Journal of Communication Disorders*, 31(2):113–134.

Dong, Y., Li, Z., Rezagholizadeh, M., and Cheung, J. C. K. (2019). Editnts: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Annual Conference of the Association for Computational Linguistics (ACL)*, pages 3393–3402.

Duan, Y., Jiang, L., Qin, T., Zhou, M., and Shum, H.-Y. (2010). An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 295–303.

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., and Harshman, R. (1988). Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 281–285. ACM.

El Kaliouby, R., Picard, R., and Baron-Cohen, S. (2006). Affective computing and autism. *Annals of the New York Academy of Sciences*, 1093(1):228–248.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, page 973–978.

Elkan, C. and Noto, K. (2008). Learning classifiers from only positive and unlabeled data. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 213–220. ACM.

Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., Freeman, W. T., and Rubinstein, M. (2018). Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619*.

Fabian Caba Heilbron, Victor Escorcia, B. G. and Niebles, J. C. (2015). Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970.

Fathi, A., Hodgins, J. K., and Rehg, J. M. (2012). Social interactions: A first-person perspective. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1226–1233. IEEE.

Fathi, A., Ren, X., and Rehg, J. M. (2011). Learning to recognize objects in egocentric activities. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3281–3288. IEEE.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1615–1625.

Feurer, M., van Rijn, J. N., Kadra, A., Gijsbers, P., Mallik, N., Ravi, S., Müller, A., Vanschoren, J., and Hutter, F. (2019). Openml-python: an extensible python api for openml. *arXiv preprint arXiv:1911.02490*.

Fischhoff, B., Slovic, P., Lichtenstein, S., Read, S., and Combs, B. (1978). How safe is safe enough? a psychometric study of attitudes towards technological risks and benefits. *Policy sciences*, 9(2):127–152.

Forman, G. (2005). Counting positives accurately despite inaccurate classification. In *European Conference on Computer Vision (ECCV)*.

Forman, G. (2008). Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206.

Freitag, M. and Roy, S. (2018). Unsupervised natural language generation with denoising autoencoders. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Frénay, B. and Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869.

Fu, Z., Tan, X., Peng, N., Zhao, D., and Yan, R. (2018). Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Gao, J., Galley, M., Li, L., et al. (2019). Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2-3):127–298.

Gao, R., Feris, R., and Grauman, K. (2018). Learning to separate object sounds by watching unlabeled video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–53.

Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. (2017). 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.

Ghandeharioun, A., Shen, J. H., Jaques, N., Ferguson, C., Jones, N., Lapedriza, A., and Picard, R. (2019). Approximating interactive human evaluation with self-play for open-domain dialog systems. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 32.

Ghosh, A., Manwani, N., and Sastry, P. (2015). Making risk minimization tolerant to label noise. *Neurocomputing*, 160:93–107.

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

Graepel, T. and Herbrich, R. (2001). The kernel gibbs sampler. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 514–520. MIT Press.

Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology.

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

Gu, C., Sun, C., Ross, D. A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., et al. (2018). Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*.

Ha, D. and Eck, D. (2017). A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*.

Halpern, Y., Horng, S., Choi, Y., and Sontag, D. (2016). Electronic medical record phenotyping using the anchor and learn framework. *Journal of the American Medical Informatics Association*, 23(4):731–740.

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

Han, J., Luo, P., and Wang, X. (2019). Deep self-learning from noisy labels. In *International Conference on Computer Vision (ICCV)*.

Harutyunyan, H., Reing, K., Ver Steeg, G., and Galstyan, A. (2020). Improving generalization by controlling label-noise information in neural network weights. In *International Conference on Machine Learning (ICML)*, pages 4071–4081. Proceedings of Machine Learning Research (PMLR).

Haykin, S. and Chen, Z. (2005). The cocktail party problem. *Neural computation*, 17(9):1875–1902.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

He, R. and McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *International conference on world wide web (WWW)*.

Hendrycks, D. and Gimpel, K. (2017). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representations (ICLR)*.

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

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 1693–1701.

Ho, A., Chuang, I., Reich, J., Coleman, C., Whitehill, J., Northcutt, C., Williams, J., Hansen, J., Lopez, G., and Petersen, R. (2015). Harvardx and mitx: Two years of open online courses fall 2012-summer 2014. *Available at SSRN 2586847*.

Hoffman, J., Pathak, D., Darrell, T., and Saenko, K. (2015). Detector discovery in the wild: Joint multiple instance and representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2883–2891.

Hooker, S., Courville, A., Dauphin, Y., and Frome, A. (2019). Selective brain damage: Measuring the disparate impact of model pruning. *arXiv preprint arXiv:1911.05248*.

Hopkins, J. and Kiela, D. (2017). Automatically generating rhythmic verse with neural networks. In *Annual Conference of the Association for Computational Linguistics (ACL)*, pages 168–178.

Huang, A. (2008). Similarity measures for text document clustering. In *Proceedings of the 6th New Zealand Computer Science Research Conference*, pages 49–56.

Huang, J., Qu, L., Jia, R., and Zhao, B. (2019a). O2u-net: A simple noisy label detection approach for deep neural networks. In *International Conference on Computer Vision (ICCV)*.

Huang, W. R., Emam, Z., Goldblum, M., Fowl, L., Terry, J. K., Huang, F., and Goldstein, T. (2019b). Understanding generalization through visualizations. *arXiv preprint arXiv:1906.03291*.

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

Jiang, L., Huang, D., Liu, M., and Yang, W. (2020b). Beyond synthetic noise: Deep learning on controlled noisy labels. In *International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research (PMLR)*, pages 4804–4815. Proceedings of Machine Learning Research (PMLR).

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

Jindal, I., Nokleby, M., and Chen, X. (2016). Learning deep networks from noisy labels with dropout regularization. In *International Conference on Data Mining (ICDM)*, pages 967–972.

Joo, H., Simon, T., Cikara, M., and Sheikh, Y. (2019). Towards social artificial intelligence: Nonverbal social signal prediction in a triadic interaction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10873–10883.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Annual Conference of the Association for Computational Linguistics (ACL)*.

Katz-Samuels, J., Blanchard, G., and Scott, C. (2019). Decontamination of mutual contamination models. *Journal of Machine Learning Research (JMLR)*, 20(41):1–57.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Khetan, A., Lipton, Z. C., and Anandkumar, A. (2018). Learning from noisy singly-labeled data. In *International Conference on Learning Representations (ICLR)*.

Koriat, A., Lichtenstein, S., and Fischhoff, B. (1980). Reasons for confidence. *Journal of Experimental Psychology: Human learning and memory*, 6(2):107.

Kremer, J., Sha, F., and Igel, C. (2018). Robust active label correction. In *Proceedings of Machine Learning Research (PMLR)*, volume 84 of *Proceedings of Machine Learning Research*, pages 308–316, Playa Blanca, Lanzarote, Canary Islands. Proceedings of Machine Learning Research (PMLR).

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

Lample, G., Subramanian, S., Smith, E., Denoyer, L., Ranzato, M., and Boureau, Y.-L. (2019). Multiple-attribute text rewriting. In *ICLR 2019*.

Lawrence, N. D. and Schölkopf, B. (2001). Estimating a kernel fisher discriminant in the presence of label noise. In *International Conference on Machine Learning (ICML)*, ICML '01, pages 306–313, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference of Machine Learning (ICML)*, volume 14, pages 1188–1196.

LeCun, Y. (1998). The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324.

Lee, J., Xie, Z., Wang, C., Drach, M., Jurafsky, D., and Ng, A. Y. (2019). Neural text style transfer via denoising and reranking. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 74–81.

Lee, Y. J., Ghosh, J., and Grauman, K. (2012). Discovering important people and objects for egocentric video summarization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1346–1353. IEEE.

Lei, T., Joshi, H., Barzilay, R., Jaakkola, T., Tymoshenko, K., Moschitti, A., and Marquez, L. (2016). Semi-supervised question retrieval with gated convolutions. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1279–1289. Annual Conference of the Association for Computational Linguistics (ACL).

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Conference of the Association for Computational Linguistics (ACL)*, pages 7871–7880.

Li, H., Cai, Y., and Zheng, W.-S. (2019). Deep dual relation modeling for egocentric interaction recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7932–7941.

Li, J., Socher, R., and Hoi, S. C. (2020). Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations (ICLR)*.

Li, W., Wang, L., Li, W., Agustsson, E., and Van Gool, L. (2017a). Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*.

Li, Y., Liu, M., and Rehg, J. M. (2018). In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 619–635.

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, L.-J. (2017b). Learning from noisy labels with distillation. In *International Conference on Computer Vision (ICCV)*, volume 00, pages 1928–1936.

Li, Y., Ye, Z., and Rehg, J. M. (2015). Delving into egocentric actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 287–295.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, Cham. Springer International Publishing.

Lipton, Z., Wang, Y.-X., and Smola, A. (2018). Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning (ICML)*.

List of Datasets for Machine Learning Research (2018). List of datasets for machine learning research — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research. [Online; accessed 22-October-2018].

Liu, D., Fu, J., Zhang, Y., Pal, C., and Lv, J. (2019). Revision in continuous space: Unsupervised text style transfer without adversarial learning. *arXiv preprint arXiv:1905.12304*.

Liu, T. and Tao, D. (2015). Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(3):447–461.

Lu, Z. and Grauman, K. (2013). Story-driven summarization for egocentric video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2714–2721.

Luo, F., Li, P., Zhou, J., Yang, P., Chang, B., Sui, Z., and Sun, X. (2019). A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019*.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Annual Conference of the Association for Computational Linguistics (ACL)*, pages 142–150, Portland, Oregon, USA. Annual Conference of the Association for Computational Linguistics (ACL).

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and Van Der Maaten, L. (2018). Exploring the limits of weakly supervised pretraining. *European Conference on Computer Vision (ECCV)*, pages 181–196.

Mallinson, J., Severyn, A., Malmi, E., and Garrido, G. (2020). Felix: Flexible text editing through tagging and insertion. *arXiv preprint arXiv:2003.10687*.

Malmi, E., Krause, S., Rothe, S., Mirylenka, D., and Severyn, A. (2019). Encode, tag, realize: High-precision text editing. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5057–5068.

Malmi, E., Takala, P., Toivonen, H., Raiko, T., and Gionis, A. (2016). Dopelearning: A computational approach to rap lyrics generation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 195–204. ACM.

Mania, H. and Sra, S. (2021). Why do classifier accuracies show linear trends under distribution shift? *arXiv preprint arXiv:2012.15483*.

Manjavacas, E., Kestemont, M., and Karsdorp, F. (2019). Generation of hip-hop lyrics with hierarchical modeling and conditional templates. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 301–310.

Mayer, R., Neumayer, R., and Rauber, A. (2008). Rhyme and style features for musical genre classification by song lyrics. In *Proceedings of the 9th International Conference on Music Information Retrieval*, pages 337–342.

McAuley, J., Targett, C., Shi, Q., and van den Hengel, A. (2015). Image-based recommendations on styles and substitutes. In *Special Interest Group on Information Retrieval (SIGIR)*, pages 43–52. ACM.

McNaney, R., Vines, J., Roggen, D., Balaam, M., Zhang, P., Poliakov, I., and Olivier, P. (2014). Exploring the acceptability of google glass as an everyday assistive device for people with parkinson's. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 2551–2554. ACM.

Menon, A. K., van Rooyen, B., and Natarajan, N. (2018). Learning from binary labels with instance-dependent noise. *Machine Learning*, 107(8):1561–1595.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A. (2018). Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 3111–3119.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mitchell, T. (1999). Twenty newsgroups dataset. https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups.

Morency, L.-P., de Kok, I., and Gratch, J. (2008). Predicting listener backchannels: A probabilistic multimodal approach. In *International Workshop on Intelligent Virtual Agents*, pages 176–190. Springer.

Mueller, J. and Thyagarajan, A. (2016). Siamese recurrent architectures for learning sentence similarity. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Murphy-Chutorian, E. and Trivedi, M. M. (2009). Head pose estimation in computer vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):607–626.

Nagrani, A., Chung, J. S., and Zisserman, A. (2017). Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*.

Natarajan, N., Dhillon, I. S., Ravikumar, P., and Tewari, A. (2017). Cost-sensitive learning with noisy labels. *Journal of Machine Learning Research (JMLR)*, 18:155–1.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013). Learning with noisy labels. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 1196–1204.

Nikolov, N. I., Malmi, E., Northcutt, C. G., and Parisi, L. (2020). Rapformer: Conditional rap lyrics generation with denoising autoencoders. In *International Conference on Natural Language Generation (INLG)*, pages 360–373.

Northcutt, C. G. (2017). Classification with noisy labels: "multiple account" cheating detection in open online courses. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA.

Northcutt, C. G., Athalye, A., and Mueller, J. (2021a). Pervasive label errors in test sets destabilize machine learning benchmarks. In *International Conference on Learning Representations Workshop Track (ICLR)*.

Northcutt, C. G., Ho, A. D., and Chuang, I. L. (2016). Detecting and preventing "multiple-account" cheating in massive open online courses. *Computers & Education*, 100:71–80.

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

Northcutt, C. G., Leon, K. A., and Chen, N. (2017a). Comment ranking diversification in forum discussions. In *ACM Conference on Learning @ Scale (L@S)*, pages 327–330.

Northcutt, C. G., Wu, T., and Chuang, I. L. (2017b). Learning with confident examples: Rank pruning for robust classification with noisy labels. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.

Northcutt, C. G., Zha, S., Lovegrove, S., and Newcombe, R. (2020). Egocom: A multi-person multi-modal egocentric communications dataset. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Oskamp, S. (1962). The relationship of clinical experience and training methods to several criteria of clinical prediction. *Psychological Monographs: General and Applied*, 76(28):1.

Oskamp, S. (1965). Overconfidence in case-study judgments. *Journal of consulting psychology*, 29(3):261.

Owens, A. and Efros, A. A. (2018). Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648.

Ozerov, A. and Févotte, C. (2010). Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):550–563.

Ozkan, D., Sagae, K., and Morency, L.-P. (2010). Latent mixture of discriminative experts for multimodal prediction modeling. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 860–868.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: bringing order to the web. Technical report, Stanford InfoLab.

Pagliardini, M., Gupta, P., and Jaggi, M. (2018). Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Annual Conference of the Association for Computational Linguistics (ACL)*, pages 311–318.

Patrini, G., Nielsen, F., Nock, R., and Carioni, M. (2016). Loss factorization, weakly supervised learning and label noise robustness. In *International Conference on Machine Learning (ICML)*, pages 708–717.

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

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Picard, R. W. (2000). *Affective computing*. MIT press.

Pleiss, G., Zhang, T., Elenberg, E., and Weinberger, K. Q. (2020). Identifying mislabeled data using the area under the margin ranking. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 17044–17056.

Potash, P., Romanov, A., and Rumshisky, A. (2015). Ghostwriter: Using an lstm for automatic rap lyric generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1919–1924.

Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., and Black, A. W. (2018). Style transfer through back-translation. In *Annual Conference of the Association for Computational Linguistics (ACL)*, pages 866–876.

Pundak, G., Sainath, T. N., Prabhavalkar, R., Kannan, A., and Zhao, D. (2018). Deep context: End-to-end contextual speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 418–425.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Ratner, A. J., De Sa, C. M., Wu, S., Selsam, D., and Ré, C. (2016). Data programming: Creating large training sets, quickly. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 3567–3575.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2019). Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, pages 5389–5400.

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

Rehg, J., Abowd, G., Rozga, A., Romero, M., Clements, M., Sclaroff, S., Essa, I., Ousley, O., Li, Y., Kim, C., et al. (2013). Decoding children's social behavior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3414–3421.

Reynolds, D. A. (2002). An overview of automatic speaker recognition technology. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages IV–4072. IEEE.

Richard, M. D. and Lippmann, R. P. (1991). Neural network classifiers estimate bayesian a posteriori probabilities. *Neural computation*, 3(4):461–483.

Rolnick, D., Veit, A., Belongie, S., and Shavit, N. (2017). Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*.

Roth, J., Chaudhuri, S., Klejch, O., Marvin, R., Gallagher, A., Kaver, L., Ramaswamy, S., Stopczynski, A., Schmid, C., Xi, Z., et al. (2019). Ava-activespeaker: An audio-visual dataset for active speaker detection. *arXiv preprint arXiv:1901.01342*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Sáez, J. A., Galar, M., Luengo, J., and Herrera, F. (2014). Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition. *Knowledge and Information Systems*, 38(1):179–206.

Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., and Aroyo, L. M. (2021). "Everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *Human Factors in Computing Systems (CHI)*.

Scott, C. (2015). A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 38, pages 838–846. Microtome Publishing.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Annual Conference of the Association for Computational Linguistics (ACL)*, pages 1715–1725.

Shankar, V., Roelofs, R., Mania, H., Fang, A., Recht, B., and Schmidt, L. (2020). Evaluating machine accuracy on ImageNet. In *International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 8634–8644. Proceedings of Machine Learning Research (PMLR).

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. (2017). Style transfer from non-parallel text by cross-alignment. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 6830–6841.

Shen, Y. and Sanghavi, S. (2019). Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 5739–5748.

Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. (2019). Meta-weight-net: Learning an explicit mapping for sample weighting. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smith, R., Self, M., and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer.

Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. (2021). Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*.

Stratou, G. and Morency, L.-P. (2017). Multisense—context-aware nonverbal behavior analysis framework: A psychological distress use case. *IEEE Transactions on Affective Computing*, 8(2):190–203.

Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). *Density Ratio Estimation in ML*. Cambridge University Press, 1st edition.

Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., and Fergus, R. (2015). Training convolutional networks with noisy labels. In *International Conference on Learning Representations (ICLR)*, pages 1–11.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *International Conference on Computer Vision (ICCV)*.

Tanno, R., Saeedi, A., Sankaranarayanan, S., Alexander, D. C., and Silberman, N. (2019a). Learning from noisy labels by regularized estimation of annotator confusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11244–11253.

Tanno, R., Saeedi, A., Sankaranarayanan, S., Alexander, D. C., and Silberman, N. (2019b). Learning from noisy labels by regularized estimation of annotator confusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., and Schmidt, L. (2020). Measuring robustness to natural distribution shifts in image classification. *Conference on Neural Information Processing Systems (NeurIPS)*, 33.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459.

Tsipras, D., Santurkar, S., Engstrom, L., Ilyas, A., and Madry, A. (2020). From imagenet to image classification: Contextualizing progress on benchmarks. In *International Conference on Machine Learning*, pages 9625–9635. Proceedings of Machine Learning Research (PMLR).

Vahdat, A. (2017). Toward robustness against label noise in training deep discriminative neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Van Rooyen, B., Menon, A., and Williamson, R. C. (2015). Learning with symmetric label noise: The importance of being unhinged. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 10–18.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008.

Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D., and Batra, D. (2018). Diverse beam search: Decoding diverse solutions from neural sequence models. *AAAI 2018*.

Wagner, R. A. and Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173.

Wang, Q., Han, B., Liu, T., Niu, G., Yang, J., and Gong, C. (2021). Tackling instance-dependent label noise via a universal probabilistic model. *arXiv preprint arXiv:2101.05467*.

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. (2019). Symmetric cross entropy for robust learning with noisy labels. In *International Conference on Computer Vision (ICCV)*.

Wei, C., Lee, J. D., Liu, Q., and Ma, T. (2018). On the margin theory of feedforward neural networks. *Computing Research Repository (CoRR)*.

Wei, H., Feng, L., Chen, X., and An, B. (2020). Combating noisy labels by agreement: A joint training method with co-regularization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wu, H. C., Luk, R. W. P., Wong, K. F., and Kwok, K. L. (2008). Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems*, 26(3).

Wu, M., Panchapagesan, S., Sun, M., Gu, J., Thomas, R., Vitaladevuni, S. N. P., Hoffmeister, B., and Mandal, A. (2018). Monophone-based background modeling for two-stage on-device wake word detection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5494–5498. IEEE.

Xia, X., Liu, T., Han, B., Wang, N., Gong, M., Liu, H., Niu, G., Tao, D., and Sugiyama, M. (2020). Part-dependent label noise: Towards instance-dependent

label noise. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 7597–7610.

Xu, Y., Cao, P., Kong, Y., and Wang, Y. (2019). L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 6225–6236.

Yonetani, R., Kitani, K. M., and Sato, Y. (2016). Recognizing micro-actions and reactions from paired egocentric videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2629–2638.

Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., and Sugiyama, M. (2019). How does disagreement help generalization against label corruption? In *International Conference on Machine Learning (ICML)*.

Yun, S., Oh, S. J., Heo, B., Han, D., Choe, J., and Chun, S. (2021). Re-labeling imagenet: from single to multi-labels, from global to localized labels. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zadeh, A., Chen, M., Poria, S., Cambria, E., and Morency, L.-P. (2017). Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250*.

Zadeh, A., Zellers, R., Pincus, E., and Morency, L.-P. (2016). Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259*.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017a). Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*.

Zhang, J., Sheng, V. S., Li, T., and Wu, X. (2017b). Improving crowdsourced label quality using noise correction. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1675–1688.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777, ICML 2020*.

Zhang, Y. and Glass, J. R. (2009). Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 398–403. IEEE.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8697–8710.