Towards Good Practices for Efficiently Annotating Large-Scale Image Classification Datasets

Yuan-Hong Liao[1,2], Amlan Kar[1,2,3], Sanja Fidler[1,2,3]
[1] University of Toronto, [2] Vector Institute, [3] NVIDIA
{andrew, amlan, fidler}@cs.toronto.edu

Abstract

Data is the engine of modern computer vision, which necessitates collecting large-scale datasets. This is expensive, and guaranteeing the quality of the labels is a major challenge. In this paper, we investigate efficient annotation strategies for collecting multi-class classification labels for a large collection of images. While methods that exploit learnt models for labeling exist, a surprisingly prevalent approach is to query humans for a fixed number of labels per datum and aggregate them, which is expensive. Building on prior work on online joint probabilistic modeling of human annotations and machine-generated beliefs, we propose modifications and best practices aimed at minimizing human labeling effort. Specifically, we make use of advances in self-supervised learning, view annotation as a semi-supervised learning problem, identify and mitigate pitfalls and ablate several key design choices to propose effective guidelines for labeling. Our analysis is done in a more realistic simulation that involves querying human labelers, which uncovers issues with evaluation using existing worker simulation methods. Simulated experiments on a 125k image subset of the ImageNet100 show that it can be annotated to 80% top-1 accuracy with 0.35 annotations per image on average, a 2.7x and 6.7x improvement over prior work and manual annotation, respectively. [1]

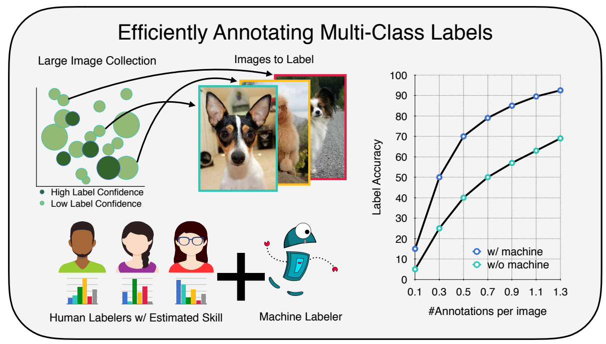

Figure 1: We tackle efficient model-assisted annotation of multi-class labels at scale. We propose improvements to prior work by incorporating self- and semi-supervised learning and address associated challenges. Extensive ablation of common design choices in realistically simulated experiments leads us to provide best practice recommendations to minimize human annotation effort.

1. Introduction

Data, the basic unit of machine learning, has tremendous impact on the success of learning-based applications. Much of the recent A.I. revolution can be attributed to the creation of the ImageNet dataset [12], which showed that image classification with deep learning at scale [25] can result in learning strong feature extractors that transfer to domains and tasks beyond the original dataset. Using citations as a proxy, ImageNet has supported at least 40,000 research projects to date. It has been unmatched as a pre-training dataset to downstream tasks, due to its size, diversity and the quality of labels. Since its conception, interest in creating large datasets serving diverse tasks and domains has skyrocketed. Examples include object detection [47], action-recognition- [10], and 3D reconstruction [32, 6], in domains such as self-driving [15, 3], and medical imaging [44].

ImageNet and its successors such as OpenImages [26] collected their data using search engines on the web, followed by human verification of either the search query term or automatically generated labels. Thus, their labeling is formulated as a verification task, *i.e*., does this image really belong to the class, allowing efficient annotation at scale.

In contrast to ImageNet labeling, in many practical use cases, the data and labels of interest are often known apriori. This departs from the case above where arbitrary images could be used by querying keywords online. A common approach used in practice is to query humans to get a fixed number of labels per datum and aggregate them [29, 22], presumably because of its simplicity and reliability. This can be prohibitively expensive and inefficient in human resource utilization for large datasets, as it assumes equal effort needed per datum. We build on prior work and investigate integration of modern learning methods to improve

[1]Code at: https://github.com/fidler-lab/efficient-annotation-cookbook

annotation efficiency for multi-class classification at *scale*.

Recent work [2] explored integrating a learnt classifier into the DS model [11] in an online setting. Their method allows principled online estimation of worker skills and label uncertainty. This is used to decide whether another human should be queried for a datum. We follow this framework, while noting that directions such as design of user-interfaces [13], computing optimal task assignment [20] etc. can provide complementary benefits.

Having a pool of workers that can be repeatably queried improves both skill estimation over time and reduces annotation noise typically found in *crowdsourcing*, where workers perform micro-tasks and their presence is fleeting. Thus, in this work we choose to focus on a *fixed worker pool*.[2]

We first investigate integrating advances in self-supervised learning in our setting. Next, we view online labeling as a semi-supervised problem and show consequent efficiency gains. These additions can sometimes lead to negative feedback cycles, which we identify and remedy. Finally, to encourage adoption into a practitioner's toolchain, we ablate several key design choices and provide a set of good practices and guidelines. We avoid the expense of running large experiments with human workers by proposing a more realistic annotator simulation that involves collecting statistics from human annotators. Prior work [2, 41] collected a large number of human labels for all experiments, leading to 1) smaller individual experiment scale and 2) a barrier for further research since these labels are not available and expensive to collect. We note that [41] also look into efficient multi-class annotation for large label sets, with a focus on efficient factorization and learning of worker abilities. This is important and orthogonal to our exploration into integration of learning methods. In summary, we make the following contributions:

- Explore the usage of advances in self-supervised learning to efficient annotation for multi-class classification
- Propose to view the annotation process as a semi-supervised learning problem, identify resulting instabilities and provide remedies
- Ablate several key design choices for the annotation process providing a set of best practices and guidelines to facilitate adoption into a practitioner's toolchain
- Provide a realistic annotator simulation to conduct such experiments at scale while avoiding the high cost of involving human annotators for every experiment
- Release a modular codebase to facilitate adoption and further research into efficient human-in-the-loop multi-class labeling

[2]This is also a growing trend in the industry, with large companies using trained labelers over offering micro-tasks.

We experiment on subsets of varying difficulty from ImageNet [12]. We show 87% top-1 label accuracy on a 100 class subset of ImageNet, with only 0.98 annotations per image. 80% top-1 label accuracy needed 0.35 annotations per image, a 2.7x reduction with respect to prior work and a 6.7x reduction over manual annotation. On the small-scale experiment using human annotations, we achieve 91% label accuracy with 2x fewer annotations.

2. Related Work

2.1. Image Annotation in Computer Vision

Large datasets [25, 29] have had a pivotal role in recent advances in computer vision. [24] make a comprehensive survey of crowdsourcing techniques. As an example, [37] integrates machines and human-labelers as an MDP and optimize questions asked to labelers to collect an object detection dataset. LSUN [50] interleaves worker annotation and training machine models, making a large high-quality dataset at a low cost. The machine models here are used to perform confirmation (ExistOrNot question). Instead, we adopt a probabilistic framework [11] that incorporates machine beliefs with human annotated labels in a principled manner. We follow the method proposed in [2, 41], who extend [11] to an online crowdsourcing framework that includes the learner as a prior. We however, consider the learning problem as a semi-supervised task. It is also akin to active learning, where the task to optimize for is the quality of the collected dataset itself.

2.2. Semi-supervised Learning

We treat the learning task in online image annotation as a semi-supervised problem enabling us to incorporate various algorithms. Graph-based semi-supervised learning [53, 43] leverages the structure in both labeled and unlabeled data for *transductive* learning. In neural networks, several methods aim at smoothing the decision boundaries of the learnt model by enforcing consistency between image augmentations [48, 34], leveraging pseudo labels on unlabelled data [39, 49, 38], interpolating between data points [42].

Recently, **self-supervised learning** to learn strong representations from unlabelled image collections has been shown to be highly performant, allowing learning tasks with limited labels. Multiple pre-text tasks are proposed to learn representation encoders from unlabelled images, such as predicting patch position [14], predicting image rotation [16], solving an image jigsaw puzzle [35], etc. Contrastive learning methods have gained popularity recently [19, 7, 17, 4, 5, 9, 36], where consistency across augmentations of an image and inconsistency across multiple images is used as a learning signal.

2.3. Truth Inference

Crowdsourcing provides low-cost noisy annotations. To infer a true label from noisy observations, one needs to infer the importance of each annotation. This problem was discussed 40 years ago in the medical domain with the Dawid-Skene model [11], optimized with EM. Many variants have since extended the DS model. GLAD [46] assumes a scenario with heterogeneous image difficulty and worker reliability. [45] consider each worker as a multidimensional entity with variables representing competence, expertise and bias. [41] parametrize worker skills factorized by a taxonomy of target concepts. Instead of using EM, BCC [23] infers the unobserved variables with bayesian inference. EBCC [28] extends BCC by considering underlying worker correlations and use rank-1 approximations to scale to large worker cohorts. However, there is still no dominant truth inference strategy [52]. Work in [2, 41] is the closest to ours. [2] extend the DS model to an online setting and incorporate a learning model as a prior. We follow this direction for multiclass classification, view the problem as a semi-supervised learning problem and rigorously ablate various design parameters as a means to provide best practice guidelines. [41] also work on multiclass classification, but focus on worker parametrization, whereas we focus on improving machine learning models in the procedure.

3. Background

In this section, we first formulate our problem and introduce the notation used (Sec. 3.1). Next, we describe the DS model [11] for probabilistic label aggregation (Sec. 3.2) and its extension to an online data collection setting with a learning algorithm in the loop [2] (Sec. 3.3).

3.1. Problem Formulation

Given a dataset with N images $\mathcal{X} = \{x_i\}_{i=1:N}$ and a set of K target labels, the goal is to infer the corresponding "true" label $\mathcal{Y} = \{y_i | y_i \in [K]\}_{i=1:N}$ from worker annotated labels $\mathcal{Z}$. Labels are sampled from M workers $\mathcal{W} = \{w_j\}_{j=1:M}$. Worker annotations are noisy, hence each image is labeled by few workers $\mathcal{W}_i = \{j | z_{ij} \in \mathcal{Z}\}$, where z_{ij} is the label assigned by worker w_j on image x_i. In an online setting (Sec. 3.3), at each time step t, a requester constructs a batch (of size B) of Human Intelligence Tasks (HITs) and assigns them to B workers. Online estimates of the true label $\mathcal{Y}^t$ at each step can be inferred from all previous annotations $\mathcal{Z}^{1:t}$. The process ends until the requester is satisfied with the current $\mathcal{Y}^t$ or if a time horizon (related to having a budget) T is reached. We omit t for simplicity.

3.2. Dawid-Skene Model

The Dawid-Skene model [11] views the annotation process as jointly inferring true labels and worker reliabilities. The joint probability of labels $\mathcal{Y}$, worker annotations $\mathcal{Z}$, and worker reliability $\mathcal{W}$ (overloaded notation for simplicity) is defined as $P(\mathcal{Y}, \mathcal{Z}, \mathcal{W}) = \prod_{i \in [N]} p(y_i) \prod_{j \in [M]} p(w_j) \prod_{i,j \in \mathcal{W}_i} p(z_{ij} | y_i, w_j)$, where $p(y_i)$ is the prior over K possible labels, $p(w_j)$ is the prior reliability of worker j, and $p(z_{ij}|y_i, w_j)$ models the likelihood of worker annotations. The worker reliability is usually represented as a confusion matrix over the label set. In practice, inference is performed using expectation maximization, where parameters for one image or worker are optimized at a time,

$$\bar{y_i} = \arg\max_{y_i} p(y_i) \prod_{j \in \mathcal{W}_i} p(z_{ij} | y_i, \bar{w_j}) \tag{1}$$

$$\bar{w_j} = \arg\max_{w_j} p(w_j) \prod_{i \in \mathcal{I}_j} p(z_{ij} | \bar{y_i}, w_j) \tag{2}$$

where $\mathcal{I}_j = \{i | z_{ij} \in Z\}$ is the set of images annotated by worker j. We refer readers to [52] for a comprehensive explanation and comparison of other work on truth inference.

3.3. Extension to Online Labeling

Lean Crowdsourcing [2] extends the DS model with a learning model in the loop and implements it in an online setting. The authors replace the label prior $p(y_i)$ with predicted probabilities from a learnt model $p(y_i | x_i, \theta)$. At each time step, after running Eq. 1 and Eq. 2, they additionally optimize the model parameters θ from $\mathcal{D} = \{x_i, \bar{y_i} | |\mathcal{W}_i| > 0\}$, *i.e.* using the current label estimate for images with at least one human annotation. Their learning model involves a fixed feature extractor ϕ with a classifier head. In this work, we use a 2 layer MLP, optimized with gradient descent.[3] Its parameters are learnt from scratch at every step by minimizing a loss function H.

$$\bar{\theta} = \arg\min_{\theta} \mathbb{E}_{(x_i, y_i) \sim \mathcal{D}} H(\bar{y_i}, p(y_i | \phi(x_i), \theta)) \tag{3}$$

To construct B HITs for the next step, they compute the bayesian risk of $\bar{y_i}$ as the expected cost of mis-labeling,

$$\begin{aligned}\mathcal{R}(\bar{y_i}) &= \sum_{y_i=1}^{K} H(\bar{y_i}, y_i) p(y_i | \mathcal{Z}_i, \theta) \\ &= \sum_{y_i=1}^{K} H(\bar{y_i}, y_i) \frac{p(y_i | x_i, \bar{\theta}) \prod_{j \in \mathcal{W}_i} p(z_{ij} | y_i, \bar{w_j})}{\sum_{y=1}^{K} p(y | x_i, \bar{\theta}) \prod_{j \in \mathcal{W}_i} p(z_{ij} | y, \bar{w_j})}\end{aligned} \tag{4}$$

At every step, they construct B HITs by randomly sampling from a set of unfinished examples, $\mathcal{U} = \{x_i | \mathcal{R}(\bar{y_i}) \geq C\}$ *i.e.* images with risk greater than a threshold. To compare with online labeling without any learning model, we adopt

[3] Previous work [2] used a linear SVM as the classifier head, but we found that using 2 layer MLP sufficient to achieve comparable performance with far less time.

(a) Commodity (b) Dog (c) Insect + Fungus

Figure 2: **Example images from datasets** (Tab. 3). *Commodity* dataset consists of data with coarse-labels, while the *Dog* and *Insect+Fungus* datasets are more fine-grained and difficult to annotate, which is reflected in the lower avg. worker accuracy.

Dataset	# Images	# Classes	Worker Acc.	Fine-Grain.
Commodity	20140	16	0.76	
Vertebrate	23220	18	0.72	
Insect + Fungus	16770	13	0.65	✓
Dog	22704	19	0.43	✓
Dog + Vertebrate	45924	37	0.59	✓
ImageNet100	125689	100	0.70	✓

Figure 3: **ImageNet100 sub-tasks.** We separate ImageNet100 into different difficulty levels.

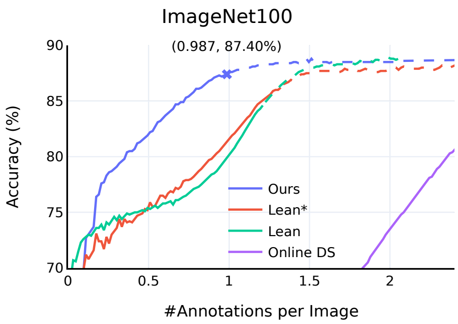

Figure 4: **Results on ImageNet100.** We compare our full framework with [2] and the online DS model on our ImageNet100 dataset (125k images). Our framework achieves 80% top-1 label accuracy with 0.35 annotation per image, a 2.7x reduction from [2], and 6.7x compared to the online DS model.

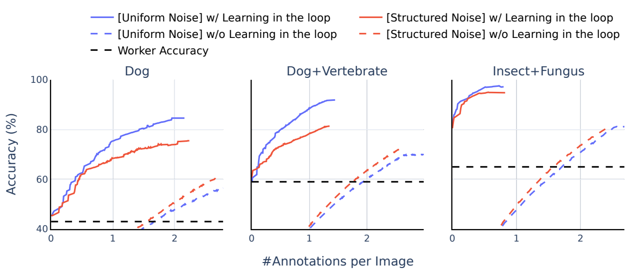

Figure 5: **Over-optimistic results from workers with uniform noise.** Human workers tend to make "structured" mistakes. Simulated workers with uniform label noise (blue) can result in over-optimistic annotation performance. Experiments under workers with structured noise reflect real-life performance better.

this sampling scheme and remove the model learning, referred to as "online DS" in the following sections.

In the following sections, we propose improvements to this online labeling framework, both in how learnt models are used and in practical design choices. Our proposed improvements are validated on multiple subsets of varying difficulty from the ImageNet dataset, using realistically simulated labelers, both of which we also introduce next.

4. Improving Annotation Efficiency

In this section, we explore modifications to the online-labeling algorithm proposed in Lean Crowdsourcing [2], focused on improving efficiency of learning in the loop and improving practical implementation choices. We present these as proposals followed by their resulting impact on annotation efficiency and label accuracy.

We first introduce datasets in Sec. 4.1 constructed by taking subsets of ImageNet [12] which we use for experiments. Experiments and ablations at scale with human labelers are prohibitively expensive. Hence, we also propose a more realistic worker simulation in Sec. 4.2.

In Sec. 4.3, we investigate whether self-supervised learning can be used to replace the feature extractor ϕ in the algorithm. Next, in Sec. 4.4 we cast the learning problem in online-labeling as a semi-supervised problem. We identify that semi-supervised learning during online-labeling can cause a negative feedback loop in the algorithm, which we mitigate. In Sec. 4.5, we ablate several key design choices and provide good practices and guidelines to encourage future adoption. Finally, in Sec. 4.6, we apply all of our proposed methods and best practices on our ImageNet100 dataset. These results are shown in Fig. 4, where we observe that we can achieve 80% top-1 label accuracy with 0.35 annotations per image, which is a 2.7x improvement over [2] and 6.7x improvement over the online DS model. We achieve 87.4% top-1 label accuracy with 0.98 annotations per image at convergence, compared to 79.5% and 46.9% in label accuracy for the competing methods. To validate our proposed approach in experiments with human annotations, we conduct a small-scale experiment in Sec. 4.7. We show that our approach achieves 91% top-1 accuracy with 2x fewer annotations compared to the previous work [2].

4.1. ImageNet100 Sandbox

Evaluating and ablating multi-class label annotation efficiency at scale requires large datasets with diverse and relatively clean labels. We construct multiple subsets of the ImageNet dataset [12] for our experiments. We use 100 classes sampled from the ImageNet label set [40] and construct smaller sub-tasks of varying difficulty using the label hierarchy. These tasks range from 20k to 125k images in size. We use the average accuracy of human workers on these datasets (Sec. 4.2) as a proxy for their difficulty, with the average human accuracy ranging from 43% to 76%. Tab. 3 details the different sub-tasks, the number of images, classes, and average human accuracy. Fig. 2 shows example images from these tasks. For each class, we use 10 images

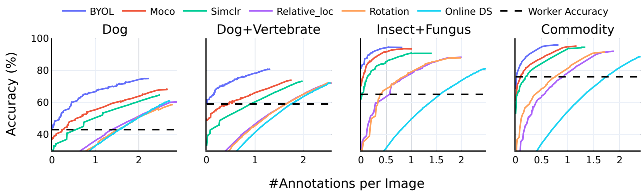

Figure 6: **Self-supervised features advance online labeling.** We compare different self-supervised features, showing that improvements in self-supervised learning translate into improvements in online labeling. For the *Dog*, using BYOL [17] helps reach the same accuracy achieved without learning in the loop (Online DS) with 5x fewer annotations. (Sec. 4.3)

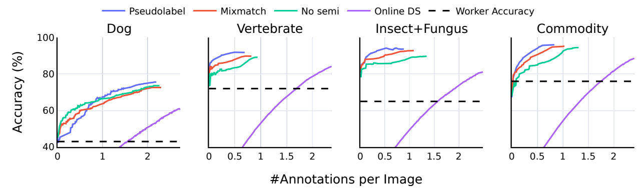

Figure 7: **Semi-supervised learning advances online labeling.** The learning problem in online-labeling can be seen as a semi-supervised problem (Sec. 4.4). We compare two semi-supervised techniques: Pseudo-Labeling [27] and MixMatch [1]. Both improve annotation efficiency on all subsets, particularly for fine-grained datasets. Surprisingly, Pseudolabels performs slightly better than a modified version of MixMatch (Sec. 4.4).

as prototype images to be provided by the requestor. They ground concepts for human annotators and also help with learning and model selection.

4.2. Simulating Realistic Workers

Prior work [30, 21] simulated workers as confusion matrices. Class confusion was modeled with symmetric uniform noise, which can result in over-optimistic performance estimates. Human annotators exhibit *asymmetric* and *structured* confusion *i.e.*, classes get confused with each other differently. Fig. 5 compares the number of annotations per image in simulation using uniform label noise vs. structured label noise that we crowdsource. We see large gaps between the two. This arises particularly when using learnt models in the loop, due to sensitivity to noisy labels coming from structured confusion in the workers.

To efficiently crowdsource a diverse set of worker confusion matrices for ImageNet100, we split the dataset into 6 disjoint subsets using the ImageNet label hierarchy and crowdsource annotations (using HITs on Amazon Mechanical Turk) for each subset. The assumption is that these sets contain most of the confusion internally and are rarely confused with each other, which we also verify. We collect 40 annotations per worker, which gives us a noisy estimated confusion matrix. To simulate a worker, we sample a confusion matrix per subset and smooth it using an affine combination with the average confusion matrix.

Through additional HITs, we verify that workers confuse classes across our subsets with a very low probability (0.03). Therefore, we uniformly spread a probability mass of 0.03 for class confusion across our subsets. Overall, we crowdsourced a total of 2680 annotations from 67 workers for these statistics. In comparison, a single experiment with 20k images with 0.5 annotation each would need 10k crowdsourced labels. More details are in the Appendix.

4.3. Self-supervised Learnt Features

With recent advances in self-supervised learning, it is feasible to learn strong image feature extractors that rival supervised learning, using pretext tasks without any labels. This allows learning in-domain feature extractors for annotation tasks, as opposed to using features pre-trained on ImageNet [2]. We compare the efficacy of using BYOL [17], SimCLR [4], MoCo [19], relative location prediction [14] and rotation prediction [16] learnt on full ImageNet raw images as the feature extractor ϕ in Eq. 3. We compare the performances in online labeling in Fig. 6. Improvements in self-supervised learning consistently improve efficiency for datasets with both fine and coarse-grained labels, with up to 5x improvement at similar accuracy compared to not using a model in the loop. For the rest of the experiments, we use BYOL [17] as our fixed feature extractor.

4.4. Learning in Online-Labeling is a Semi-Supervised Problem

During online-labeling, the goal is to infer true labels for *all* images in the dataset, making learning θ akin to transductive learning, where the test set is observed and can be used for learning. Thus, it is reasonable to expect efficiency gains if the dataset's underlying structure is exploited by putting the unlabeled data to work, using semi-supervised learning. This has also been demonstrated recently in the related field of active learning [33]. Eq. 3 accordingly can be modified to,

$$\bar{\theta} = \arg\min_{\theta} \mathbb{E}_{(x_i, y_i) \sim \{x_i, \bar{y}_i\}_{1:N}} H(\bar{y}_i, p(y_i|\phi(x_i), \theta))$$

Various methods have been proposed in the literature to improve semi-supervised learning. Since the feature extractor ϕ is fixed in our setting, we adopt the off-the-self semi-supervised learning algorithms that work directly on a feature space. Specifically, we investigate using Pseudolabels [27] and MixMatch [1].

Pseudolabeling refers to using the current model belief on unlabeled data as labels for learning. We use hard pseudo-labels (argmax of belief), and only retain those labels whose highest class probability falls above a threshold τ. We use model predictions from the previous time step to generate pseudo labels and set the threshold τ to be 0.1.

$$\bar{\theta} = \arg\min_{\theta} \mathbb{E}_{(x_i, y_i) \sim \{x_i, \bar{y}_i | p(\bar{y}_i|\mathcal{Z}_i) > 1-\tau\}} H(\bar{y}_i, p(y_i|\phi(x_i), \theta))$$

MixMatch constructs virtual training examples by mixing the labeled and unlabeled data using a modified version of MixUp [51]. We modify MixMatch for our use-case and provide details in the Appendix.

Fig. 7 shows results using PseudoLabels and MixMatch during learning. Semi-supervised learning consistently improves annotation efficiency across datasets, with more pronounced improvements shown in coarse-grained datasets. For *Dog*, while all methods fail to reach 80% accuracy due to the poor quality of worker annotations, using PseudoLabels reaches the performance of its counterpart at convergence with 30% fewer annotations.

Despite its simplicity and other published results, we surprisingly find Pseudolabeling performs better than MixMatch in our case. We remind the reader that for online-labeling, the inputs to the semi-supervised learnt models are the feature vectors $\phi(x)$ instead of raw images. Therefore, various data augmentation strategies applied at an image level cannot be applied directly, which we hypothesize to be the reason for the observed trend. In the rest of the experiments, we use Pseudolabels by default.

Semi-supervised learning during online-labeling can be unstable: Learning in the loop helps provide a better prior for label-aggregation (Eq. 1) while also improving worker skill inference (Eq. 2). We find that alternate maximation of these two objectives can lead to a negative feedback cycle once either of them converges to a bad solution, often resulting in divergence. It is known that EM optimization does not always converge to a fixed point, with the likelihood function of the iterates likely to grow unbounded [31]. Using semi-supervised learning exacerbates this issue since wrong confident intermediate labels can become more confident through subsequent time-steps. Fig. 8 shows this divergence behavior. We mitigate these with simple heuristics related to EM convergence and model selection through time. During EM, there are two popular ways to determine convergence: 1) Hard Constraint: stopping if the E-step does not change across steps, and 2) Soft Constraint: stopping if the average likelihood of the observation changes minimally across steps.

$$\text{Hard: } \arg\max p(y_i^t|z_i^t) = \arg\max p(y_i^{t-1}|z_i^{t-1}), \forall i \quad (5)$$

$$\text{Soft: } \left|\frac{\sum_{i\in[N]}\sum_{j\in\mathcal{W}_i} p(z_i^t|y_i^t, w_j^t)}{|\mathcal{Z}^t|} - \frac{\sum_{i\in[N]}\sum_{j\in\mathcal{W}_i} p(z_i^{t-1}|y_i^{t-1}, w_j^{t-1})}{|\mathcal{Z}^{t-1}|}\right| \leq \epsilon \quad (6)$$

We find that adopting the latter mitigates divergence during EM with $\epsilon = 0.01$. Moreover, we perform model selection across time-steps *i.e.*, do not replace the model if its validation performance does not improve. We use a fixed validation set comprised of the prototype images across steps, ablated in Sec. 4.5. We show the efficacy of applying the proposed heuristics in Fig. 8.

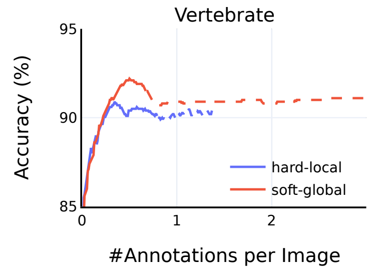

Figure 8: **Instability in model parameter learning (gradient descent) and the label aggregation (EM)** during semi-supervised learning can lead to a negative feedback loop. Using two simple heuristics, **global** model selection and **soft** convergence in EM, avoids this issue. The dashed lines show results beyond our early stopping criterion (Sec. 4.5.9)

4.5. Practical Considerations

We ablate several design choices involved in implementing such an online-labeling system, leading us to identify guidelines for future practitioners. We find these design choices can significantly affect efficiency results, discussed sequentially in the following sections.

4.5.1 Generating calibrated model likelihoods

Prior work [2] uses a modified cross-validation approach to generate model likelihoods. They ensure that the estimated prior $p(y_i|x_i, \bar{\theta})$ is produced using a model that was not trained or validated on labels from image i. We find that this can underperform when estimated labels are noisy, which pollutes validation splits and makes calibration challenging. Instead, we propose to use the clean prototype images as the validation set. We ablate the importance of having clean validation and performing cross-validation in Fig. 10. We find that having a clean validation set is more important than using cross-validation. All our experiments use temperature calibration [18] with a clean validation set comprised of prototype images to be provided by the task requestor.

4.5.2 Model update frequency

How often should one update the model vs. collect human labels? The latency of model updates vs. collection dictates varies across applications. We find that for fine-grained datasets, lower update frequencies (higher number of annotations per update) tend to overshoot in the number of annotations, while the method is robust to low update frequency on coarse-grained datasets. See more details in the Appendix.

4.5.3 Tuning Hyperparameters

Tuning hyperparameters is difficult when the task is to label data itself. We split our prototypes (10 per class) into train and validation sets to tune hyperparameters. Though there are few prototypes, we find that this works almost as well as tuning with backdoor label access. The requestor can

(a) Images with no annotations

(b) Images with 3 annotations

Figure 9: **Example images with different numbers of annotations under our full framework on ImageNet100.** Images without annotations (left) usually contain only one centered object. Images with 3 annotations (right) either have multiple objects (Bottlecap), a small object of interest (Mexican Hairless, Walking Stick), ambiguous objects (Boxer, Cocktail Shaker), and some not in their typical state (Kuvasz puppies). We also find simple examples (Milk Can) with 3 annotations, pointing to room for improvement.

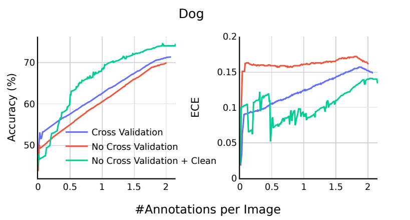

Figure 10: **Validation set selection and calibration.** We find using a clean validation set is more important than performing cross-validation.

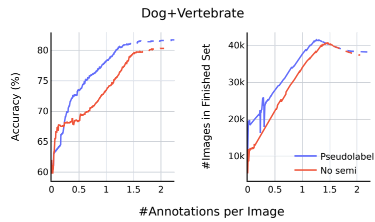

Figure 11: **Early-stop** by monitoring the size of the finished set. This avoids over-sampling for confusing images. Dashed lines represent trajectories using stopping criterion from [2].

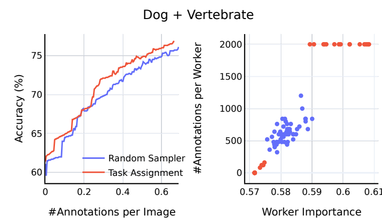

Figure 12: **Task Assignment.** We adopt a simple greedy task assignment scheme (Eq. **??**) using learnt worker skills. We show that the learnt skills help assign more tasks to "important" workers.

always add more images to the prototype set from the up-to-date annotations for improved hyperparameters search.

4.5.4 Pre-identifying Worker Skills

We explore to leverage class-dependent and worker-dependent priors. In reality, the requestor can ask gold standard questions or apply prior knowledge to design the prior. In our experiment, we find that this is especially useful for fine-grained datasets. The best explored prior improves accuracy by 15% in *Dog*, while in *Commodity*, the improvement is marginal. See more details in the Appendix.

4.5.5 Task Assignment with Inferred Skills

There are certain particularly hard classes, with only a few workers having enough expertise to annotate them correctly. We ask whether the learnt skills can be used to assign tasks better. Prior work on (optimal) task assignment tackle crowdsourcing settings with vastly different simplifying assumptions [20, 8], and designing a new task assignment scheme is out of the scope of this paper. To verify if the learnt worker skills help with task assignment, we propose a simple greedy algorithm with a cap on the maximum number of annotations α allowed per worker.

Fig. 12 shows results with task assignment with $\alpha = 2000$. The simple task assignment allows saving 13% of annotations to reach 75% label accuracy. On the right, we show the distribution between worker importance. Ideally, number worker annotations would be highly correlated with worker importance. The results show that, while not perfect, the learnt skills indeed help assign more tasks to important workers.

4.5.6 Number of Workers

We explore how annotation efficiency is affected by the different number of simulated workers. With the fixed number

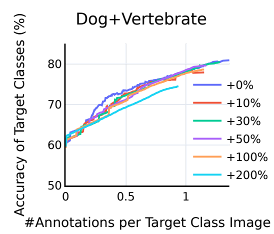

Figure 13: **Dataset contains OOD images.** Our method retains efficiency till around 100% out-of-distribuion images. (Sec. 4.5.7)

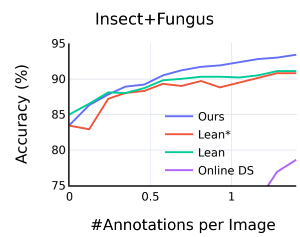

Figure 14: **Transfer to human workers.** Our method increase efficiency w.r.t. previous work [2] even when using human annotations, achieving 91% accuracy with 2x reduction. (Sec. 4.7)

of annotations, we find that having more workers hurts the performance due to the fewer observations of each worker, resulting in poor worker skill estimation. On the contrary, having fewer workers significantly helps in fine-grained datasets. Reducing the number of workers from 50 to 10 improves the accuracy by 17% in *Dog*. See more details in the Appendix.

4.5.7 Pre-Filtering Datasets

We have assumed that the requestor performs perfect filtering before annotation, *i.e.*, all the images to be annotated belong to the target classes, which does not always hold. We add an additional "None of These" class and ablate annotation efficiency in the presence of unfiltered images. We include different numbers of images from other classes and measure the mean precision with the number of annotations of the target classes. In Fig.13, we see that even with 100% more images from irrelevant classes, comparable efficiency can be retained on a fine-grained dataset.

4.5.8 Risk Threshold

The risk threshold C in online labeling determines how fast the annotation process converges and the possible final label quality. As expected, having lower risk threshold results in slower convergence, but improves final label quality. We use $C = 0.1$ for all other experiments in this paper. See more details in the Appendix.

4.5.9 When to Stop Annotating?

A clear criterion to stop annotation is when the unfinished set of images (images with estimated risk greater than a threshold) is empty [2, 41]. However, we observe that the annotation accuracy usually saturates and then grows slowly because of a small number of data points that are heavily confused by the pool of workers used. Therefore we suggest that the requestor 1) stop annotation at this time and separately annotate the small number of unfinished samples, possibly with expert annotators, and 2) set a maximum number of annotations per image we use 3 in our paper. However, how do we automatically decide when to stop without access to true labels? We find that performing early stopping on the size of the finished set of images is sufficient, as shown in Fig. 11. If the finished set size does not increase from its global maximum value for β consecutive steps, we stop annotation. We adopt this stopping criterion and set $\beta = 5$ for all other experiments in this paper.

4.6. Putting it Together on ImageNet100

We finally compare our full framework (Algo 1), with [2] (Lean), an adapted version with prototypes as validation set and without cross-validation (Lean*) and the online DS model in Fig. 4. Dashed lines represent results of removing the stopping criterion mentioned in Sec. 4.5.9. Our framework consistently provides higher accuracy and stable improvement over time. We achieve nearly 87% top-1 label accuracy on a 100 class subset of ImageNet with only 0.98 annotations per image and 80% top-1 label accuracy with 0.35 annotation per image, a 2.5x reduction reduction w.r.t. “Lean*”, 2.7 reduction w.r.t “Lean” and a 6.7x reduction over “Online DS”.

In Fig. 9, we visualize images with 0 and 3 annotations received, respectively, along with their ground truth ImageNet label. We find that images that got 0 annotations usually contain only one clear centered object. Images with 3 annotations sometimes have multiple objects (Bottlecap), or have a small object (Walking Stick, Mexican Hairless) or an ambiguous object (Boxer, Cocktail Shaker). We also note that there are simple images that receive 3 annotations (Milk Can), showing room for improvement.

4.7. Transfer to Human Workers

To validate if these good practices apply outside our simulation and in the real world, we collect 3088 annotations of 1878 images from *Insesct+Fungus*. Fig. 14 shows that our method works well when transferring to human workers, achieveing 91% accuracy with 2x fewer required annotations w.r.t. previous work [2]. More details in the Appendix.

Algorithm 1: Efficient Annotation

Input: Unlabeled images $\mathcal{X} = \{x_i\}_{i=1}^N$ and workers $\mathcal{W} = \{w_j\}_{j=1}^M$
Output: labels $\mathcal{Y} = \{y_i\}_{i=1}^N$

Set unfinished set $\mathcal{U} = \{i\}_{i=1}^N$, finished set $\mathcal{F} = \emptyset$, $loss^* =$ inf, and $\bar{\theta^*}$ is randomly initialized
$\bar{\theta} \leftarrow \bar{\theta^*}$
$\phi \leftarrow$ Self-supervised learning on $\mathcal{X}$
while *stopping criterion 4.5.9 is not met* **do**
  Construct B HITs from $\mathcal{U}$, sample B workers
  Obtain annotations $\mathcal{Z}$
  Initialize worker skills $\bar{\mathcal{W}}$
  while *Eq. 6 is not met* **do**
    $\bar{\mathcal{Y}} \leftarrow$ Aggregate $\mathcal{Y}$ by Eq. 1
    $\bar{\mathcal{W}} \leftarrow$ Maximize $\mathcal{W}$ by Eq.2
  $\bar{\theta} \leftarrow$ Model Parameter learning by Eq. 5
  $\bar{\theta} \leftarrow$ Calibrate with prototypes 4.5.1
  $loss \leftarrow$ Measure loss on prototypes
  if $loss \leq loss^*$ **then**
    $\bar{\theta^*} \leftarrow \bar{\theta}$; $loss^* = loss$
  else
    $\bar{\theta} \leftarrow \bar{\theta^*}$
  for $i \in \{i\}_{1:N}$ **do**
    if $\mathcal{R}(\bar{y_i}) < C$: $\mathcal{F} \leftarrow \mathcal{F} \cup i, \mathcal{U} \leftarrow \mathcal{U} \setminus i$
return $\mathcal{Y}$

5. Discussion and Conclusion

We presented improved online-labeling methods for large multi-class datasets. In a realistically simulated experiment with 125k images and 100 labels from ImageNet, we observe a 2.7x reduction in annotations required w.r.t. prior work to achieve 80% top-1 label accuracy. Our framework goes on to achieve 87.4% top-1 accuracy at 0.98 labels per image. Along with our improvements, we leave open questions for future research. 1) Our simulation is not perfect and does not consider individual image difficulty, instead only modeling class confusion. 2) How does one accelerate labeling beyond semantic classes, such as classifying viewing angle of a car? 3) ImageNet has a clear label hierarchy, which can be utilized to achieve orthogonal gains [41] in the worker skill estimation. 4) Going beyond classification is possible with the proposed model by appropriately modeling annotation likelihood as demonstrated in [2]. However, accelerating these with learning in the loop requires specific attention to detail per task, which is an exciting avenue for future work. 5) Finally, we discussed annotation at scale, where improvements in learning help significantly. How can these be translated to small datasets? We discuss these questions more in the Appendix, and release a codebase to facilitate further research in these directions.

Acknowledgments: This work was supported by ERA, NSERC, and DARPA XAI. SF acknowledges the Canada CIFAR AI Chair award at the Vector Institute.

References

[1] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019.

[2] Steve Branson, Grant Van Horn, and Pietro Perona. Lean crowdsourcing: Combining humans and machines in an online system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *CoRR*, abs/1903.11027, 2019.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[6] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaako Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *NeurIPS*, 2019.

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[8] Xi Chen, Qihang Lin, and Dengyong Zhou. Statistical decision making for optimal budget allocation in crowd labeling. *The Journal of Machine Learning Research*, 16(1):1–46, 2015.

[9] Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *arXiv preprint arXiv:2007.00224*, 2020.

[10] Dima Damen, Hazel Doughty, Giovanni Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines. 2020.

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

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S Bernstein, Alex Berg, and Li Fei-Fei. Scalable multi-label annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3099–3102, 2014.

[14] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[18] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*, 2017.

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[20] Chien-Ju Ho, Shahin Jabbari, and Jennifer Wortman Vaughan. Adaptive task assignment for crowdsourced classification. In *International Conference on Machine Learning*, pages 534–542, 2013.

[21] Gang Hua, Chengjiang Long, Ming Yang, and Yan Gao. Collaborative active learning of a kernel machine ensemble for recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1209–1216, 2013.

[22] Parneet Kaur, Karan Sikka, Weijun Wang, Serge Belongie, and Ajay Divakaran. Foodx-251: a dataset for fine-grained food classification. *arXiv preprint arXiv:1907.06167*, 2019.

[23] Hyun-Chul Kim and Zoubin Ghahramani. Bayesian classifier combination. In *Artificial Intelligence and Statistics*, pages 619–627, 2012.

[24] Adriana Kovashka, Olga Russakovsky, Li Fei-Fei, and Kristen Grauman. Crowdsourcing in computer vision. *arXiv preprint arXiv:1611.02145*, 2016.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[26] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018.

[27] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks.

[28] Yuan Li, Benjamin Rubinstein, and Trevor Cohn. Exploiting worker correlation for label aggregation in crowdsourcing. In *International Conference on Machine Learning*, pages 3886–3895, 2019.

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In

European conference on computer vision, pages 740–755. Springer, 2014.

[30] Chengjiang Long and Gang Hua. Multi-class multi-annotator active learning with robust gaussian process for visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2839–2847, 2015.

[31] Geoffrey J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley-Interscience, 2008.

[32] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *CoRR*, abs/2003.08934, 2020.

[33] Sudhanshu Mittal, Maxim Tatarchenko, Özgün Çiçek, and Thomas Brox. Parting with illusions about deep active learning. *arXiv preprint arXiv:1912.05361*, 2019.

[34] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[35] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[36] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

[37] Olga Russakovsky, Li-Jia Li, and Li Fei-Fei. Best of both worlds: Human-machine collaboration for object annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[38] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[39] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[40] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[41] Grant Van Horn, Steve Branson, Scott Loarie, Serge Belongie, and Pietro Perona. Lean multiclass crowdsourcing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2714–2723, 2018.

[42] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*, 2019.

[43] Fei Wang and Changshui Zhang. Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):55–67, 2007.

[44] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2097–2106, 2017.

[45] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*, pages 2424–2432, 2010.

[46] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pages 2035–2043, 2009.

[47] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[48] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

[49] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.

[50] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[51] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[52] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. Truth inference in crowdsourcing: Is the problem solved? 10(5):541–552, 2017.

[53] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004.