A Derivation of Proposition 2

Proposition. *If the label prior $p(y_i)$ is a uniform distribution, $\mathbb{I}(z_{ij}; y_i|\theta, Z_t, \hat{W})$ can be expressed as follow:*

$$\begin{aligned}&\mathbb{I}(z_{ij}; y_i|\theta, Z_t, \hat{W})\\&= \mathbb{E}_{z_{ij}}[\sum_{y_i} p(y_i|z_{ij}, \theta, Z_t, \hat{W}) \log p(z_{ij}|y_i, \hat{w}_j) - \frac{1}{K} \log p(z_{ij}|y_i, \hat{w}_j)]\end{aligned}$$

Proof.

$$\begin{aligned}&\mathbb{I}(z_{ij}; y_i|\theta, Z_t, \hat{W})\\&= \mathbb{E}_{z_{ij}}[\mathcal{D}_{\text{KL}}(p(y_i|z_{ij}, \theta, Z_t, \hat{W})||p(y_i|\theta, Z_t, \hat{W}))]\\&= \mathbb{E}_{z_{ij}}[\sum_{y_i} p(y_i|z_{ij}, \theta, Z_t, \hat{W}) \log \frac{p(y_i|z_{ij}, \theta, Z_t, \hat{W})}{p(y_i|\theta, Z_t, \hat{W})}]\\&= \mathbb{E}_{z_{ij}}[\sum_{y_i} p(y_i|z_{ij}, \theta, Z_t, \hat{W}) \log \frac{\frac{p(y_i|\theta)p(Z_t|y_i,\hat{W})p(z_{ij}|y_i,\hat{w}_j)}{p(Z_t|\hat{W})p(z_{ij}|\hat{w}_j)}}{\frac{p(y_i|\theta)p(Z_t|y_i,\hat{W})}{p(Z_t|\hat{W})}}]\\&= \mathbb{E}_{z_{ij}}[\sum_{y_i} p(y_i|z_{ij}, \theta, Z_t, \hat{W}) \log \frac{p(z_{ij}|y_i, \hat{w}_j)}{p(z_{ij}|\hat{w}_j)}]\\&= \mathbb{E}_{z_{ij}}[-\log p(z_{ij}|\hat{w}_j) + \sum_{y_i} p(y_i|z_{ij}, \theta, Z_t, \hat{W}) \log p(z_{ij}|y_i, \hat{w}_j)]\\&= \mathbb{E}_{z_{ij}}[-\sum_{y_i} p(y_i) \log p(z_{ij}|y_i, \hat{w}_j) + \sum_{y_i} p(y_i|z_{ij}, \theta, Z_t, \hat{W}) \log p(z_{ij}|y_i, \hat{w}_j)]\\&= \mathbb{E}_{z_{ij}}[\sum_{y_i} (p(y_i|z_{ij}, \theta, Z_t, \hat{W}) - \frac{1}{K}) \log p(z_{ij}|y_i, \hat{w}_j)]\\&= \mathbb{E}_{z_{ij}}[\sum_{y_i} p(y_i|z_{ij}, \theta, Z_t, \hat{W}) \log p(z_{ij}|y_i, \hat{w}_j) - \frac{1}{K} \log p(z_{ij}|y_i, \hat{w}_j)]\end{aligned}$$

The first equality comes from the relation between mutual information and Kullback-Leibler divergence $\mathbb{I}(X;Y) = \mathbb{E}_Y[\mathcal{D}_{\text{KL}}(p_{X|Y}||p_X]$. The 7th equality leverages the assumption that the label prior $p(y_i)$ is a uniform distribution. □

B Additional Analysis

B.1 Local and Global Influence w.r.t. Time

The acquisition function in LA-BALD consists of two terms, global influence $\mathbb{I}(z_{ij}; \theta)$ and local influence $\mathbb{I}(z_{ij}; y_i|\theta)$. We are interested in how the importance of global/local influence changes through time. Since we use the ranking of the LA-BALD instead of the exact values, directly comparing the values only tells part of the story. Instead, we compute the top 50 image-worker pairs with $S_{\text{LA-BALD}}$, $\mathbb{I}(z_{ij}; \theta)$ and $\mathbb{I}(z_{ij}; y_i|\theta)$ and treat them as sets to compute the Jaccard similarity $J(.)$. We denote them as $D_{\text{LA-BALD}}$, D_{global} and D_{local}. Since $\mathbb{I}(z_{ij}; \theta)$ governs the global influence, $J(D_{\text{LA-BALD}}, D_{\text{global}})$ shows how much LA-BALD dedicates the budgets to the global influence. On the other hand, since $\mathbb{I}(z_{ij}; y_i|\theta)$ represents the local influence, $J(D_{\text{LA-BALD}}, D_{\text{local}})$ shows how much LA-BALD favors individual improvements.

Fig. 10 visualizes $J(D_{\text{LA-BALD}}, D_{\text{global}})$ (blue) and $J(D_{\text{LA-BALD}}, D_{\text{local}})$ (green) in *Insect+Fungus* and *Commodity*. Local influence is more important than global influence. One of the reasons is that the difference of $\mathbb{I}(z_{ij}; \theta)$ across different images is small. Since the model is initialized with unlabeled images via self-supervised learning, the values of each example become lower and similar. The active learning commiunity Chan et al. (2021); Bengar et al. (2021) have similar observations.

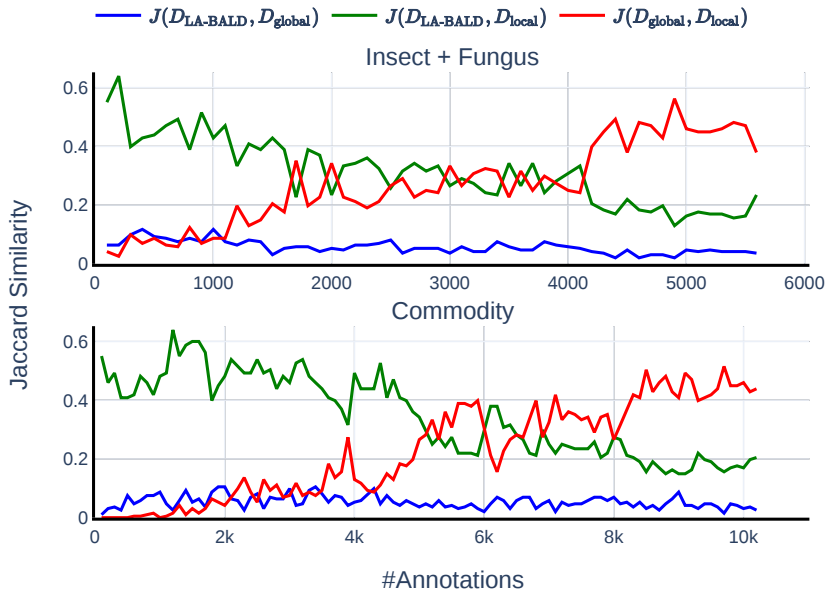

Figure 10: **Importance of Global Influence $\mathbb{I}(z_{ij};\theta)$ and Local Influence $\mathbb{I}(z_{ij};y_i|\theta)$ through time**.

Another observation is that $J(D_{\text{global}}, D_{\text{local}})$ (red) grows through time. As we explain in Sec. 4.3 and Sec. 4.4, both scores are highly dependent on the estimated worker skills and are maximized when the uncertainty of w_j becomes small. The worker uncertainty is expected to decrease as we obtain more annotations. Having high $J(D_{\text{global}}, D_{\text{local}})$ means that $S_{\text{LA-BALD}}$ becomes more similar to $S_{\text{J-BALD}}$ in the later course of the labelling process. To summarize, LA-BALD favors the local influence at the beginning. Still, as we obtain more annotations, the local and global influences tend to favor similar image-worker pairs.

B.2 Time Complexity

Joint optimization over images and workers results in higher time complexity than SS. Here, we analyze the time complexity to compute LA-BALD. Computing $\mathbb{I}(z_{ij};\theta)$ requires us to perform β MC sampling, and we need to apply the worker transformation, in the form of confusion matrix, to get $p(z_{ij}|\theta)$. To compute $\mathbb{I}(z_{ij};y_i|\theta)$, we need to compute $p(y_i|z_{ij},\theta)$, which requires us to perform β MC samples and β annotation sampling. Ultimately, we need to perform the aggregation over K classes as in Eq. 2. To sum up, for a pair of image-worker (i,j), the time complexity to compute $\mathbb{I}(z_{ij};\theta)$ is $\mathcal{O}(\beta K^2)$, and the time complexity to compute $\mathbb{I}(z_{ij};y_i|\theta)$ is $\mathcal{O}(\beta^2 K)$.

B.3 Robustness to imbalanced datasets

Our approach relies on the ML model learning in the loop. An imbalanced data pool would cause a biased ML model. To test the robustness of imbalanced datasets, we follow the experiments setting in Fig. 5 but vary the number of images of each class. We set the ratio between the majority and minority classes to be 10 $\frac{\max_{k\in K} n_k}{\min_{k\in K} n_k} = 10$ and find that LA-BALD reduces the number of annotations by 20% and 6 % compared to J-BALD.

C Bayesian Active Learning via MC dropout

We provide background to perform Bayesian active learning via MC dropout. The idea of prior work Gal et al. (2017) is that bayesian neural networks (BNNs) provide better uncertainty measures. In BNNs, the parameters of the model are sampled from a distribution. During inference, we need to

integrate all possible sets of parameters. However, it is intractable to integrate all possible parameters in the distribution, so we need to take the Monte Carlo samples for approximation.

Monte Carlo dropout Gal et al. (2017) leverages the idea that every parameter θ_i has a Bernoulli prior, multiplied by m_i. With MC dropout, running a network with dropout is equivalent to running a BNN sampled from a Bernoulli parameter prior. Combined with MC sampling, we can perform BNN inference as follows:

$$\begin{aligned} p(y|x) &= \int p(y|x,\theta)p(\theta)d\theta \\ &\approx \frac{1}{N}\sum_{n} p(y|x,\theta_i), \theta_i \sim p(\theta) \end{aligned} \tag{6}$$

where x, y, θ are input, output, and parameters. $p(\theta)$ is the learned Bernoulli priors, consisting of a Bernoulli prior multiplied by a scalar m. With MC dropout, we approximate the entropy $\mathbb{H}(y_i|\theta)$ in Sec. 3.3. For more derivation details, please refer to Gal et al. (2017).

D Limitations

With consistent improvements, LA-BALD stills have some limitations, which we discuss in this section.

Time Complexity. Computing the scores of every image-worker pair is time-consuming. From the time complexity analysis in Sec. B.2, it takes $\mathcal{O}(NM\beta^2K)$ to compute all possible pairs. A possible workaround is to cluster the unlabeled images and sample over the cluster center Nguyen & Smeulders (2004) so that we can reduce to $\mathcal{O}(\log(N)M\beta^2K)$.

Semi-supervised Active Learning. Recent works Gao et al. (2020); Huang et al. (2021) have shown promising improvements in performing active learning to the semi-supervised learner. However, prior art Chan et al. (2021) shows an inconsistent conclusion that unlabeled data saturates the improvements brought by different active learning algorithms. Therefore, we leave the efforts to measure the annotation influences w.r.t. a semi-supervised learner out of the scope of this paper.

Dynamic Worker skills. In practice, human workers get more familiar when exposed to more tasks Nakayama et al. (2021). In this work, we ignore the dynamics of worker skills since there is no off-the-shelf realistic worker simulation considering this factor.

E Computational Resources

The computational part of this work can be separated into three parts: worker simulations, online-learned model training, and calculating BALD scores. Worker simulation contains only CPUs computations and is easily parallelized. We perform the worker simulations on an 8-CPU instance. Our online-learned model consists of a fixed feature extractor and a feedforward neural network. We only need to update the weights in the feedforward neural network. We use one Titan XP for the network training. We show the time complexity of calculating BALD in sec B.2. Though the time complexity is high, it's easily parallelized. We use the same 8-CPU instance to compute each image-worker pair's BALD scores. The most computationally expensive experiment is in Fig. 7, which takes around 30 hours.

F Implementation Details

Reproducibility is important in the machine learning community. This section reports all the implementation details to reproduce the experiments.

Network Learning. We first perform self-supervised learning, BYOL Grill et al. (2020), on full ImageNet and treat it as a feature extractor. We train a feedforward neural network with two hidden fully connected layers and keep the feature extractor fixed for each sub-task Each layer contains 128 neurons. We use learning rate $1e-4$, weight decay $1e-3$, and batch size 1024 in training.

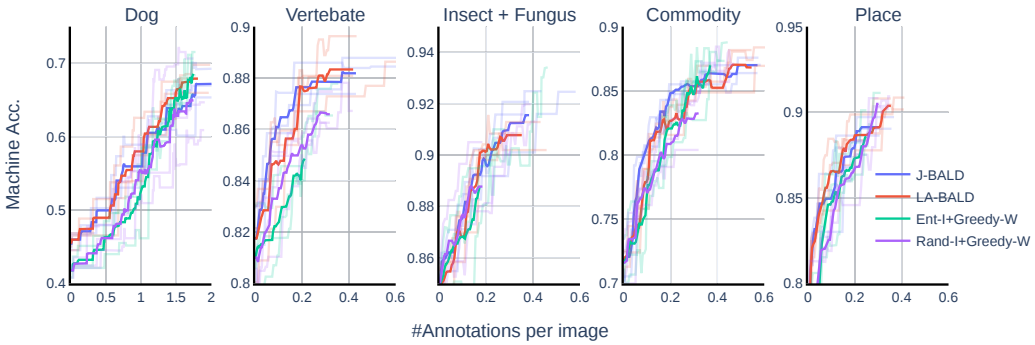

Figure 11: **Full Comparison in ML performance betwee LA-BALD and the baselines.** The ML model performances of LA-BALD and BALD are comparable and they are usually better than the ML model performances of SS.

We perform pseudo-labelling to train the feedforward neural network. At each step, we perform EM over all the annotations, with the prior being the machine predictions from the previous time step. We then use the inferred labels to train the neural networks. Since we use the probability to determine which images require more annotations, we perform temperature calibration Guo et al. (2017) on the network. We select all the confident examples as train data and use the prototypical images as validation data, equivalent to pseudo-labelling. The confidence threshold γ is set to 0.9, and we use the maximum probability over classes to measure the confidence of an example. We keep the current model's predictions if the validation error is lower than the lowest achieved. Otherwise, we discard the current model and use the previous one.

Worker Simulations. Prior work Liao et al. (2021) collects the annotations from multiple human workers for worker simulations. We use 50 crowd workers and 10 domain experts. For crowd workers, we compute the confusion matrix in three steps: (1) sum up all the annotations in the sub-tasks, (2) random sample the annotations from 3 different workers and apply the weight with 10 (3) Sum the results from 1 and 2. and normalize them. The difference between the crowd and the expert workers is that we increase the diagonal terms by 50% of the probability mass. This way results in diverse worker simulations.

Sampler. We set the confidence threshold to be 0.9 and sample size $B = 100$. We stop the annotation process when there is less than $\lfloor \frac{B}{10} \rfloor$ unconfident examples. We limit the maximum number of worker annotations of each image to 3 to avoid over-annotating ambiguous images.

EM. We use the same prior for every worker. The prior is set to be the average per-class accuracy across every worker. The prior strength is 1 since we assume to have many observations in large-scale data labelling processes.

Other details. For the MC dropout, we use $\beta = 100$ MC samples. We set the dropout rate to be 0.3 for every dropout layer to approximate the distribution of the parameters.

G Machine Learner Performance

In Fig. 11, we report the full comparison of LA-BALD with the baselines. The ML model performances of LA-BALD and J-BALD are comparable and usually better than the ML model performances of SS.

H Over-Collecting More Annotations

Since the label aggregator combines the worker annotations and the predictions of the ML model in a probabilistic manner, we can perform early stopping when the labels of most of the images are confident as in prior work Liao et al. (2021). It is interesting to see the results when we over-collect

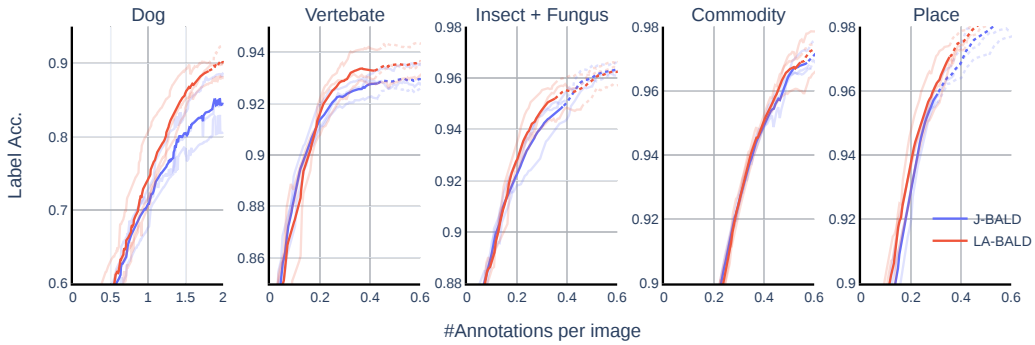

Figure 12: **LA-BALD vs. J-BALD in Over-Annotation Area.** In the over-annotation area, LA-BALD consistently outperforms J-BALD.

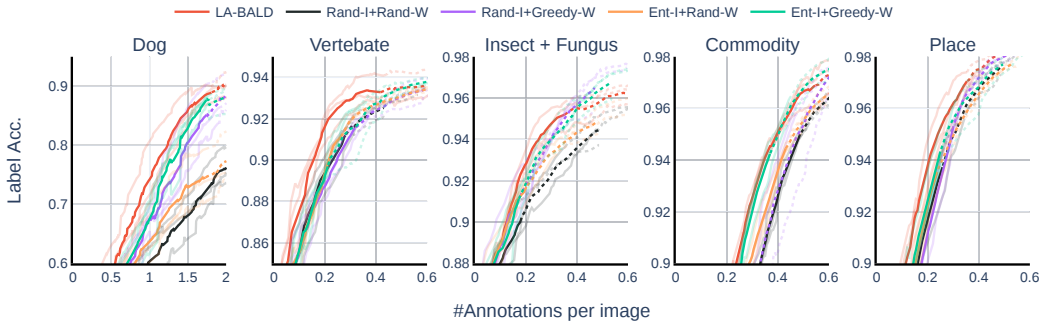

Figure 13: **LA-BALD vs. SS in Over-Annotation Area.** In the over-annotating area, LA-BALD either ends up with higher label accuracy with the same amount of annotation or reaches similar label accuracy but with a faster pace.

the annotations. In Fig. 12 and Fig. 13, we report the performances of LA-BALD and all the baselines. The solid lines represent the performances before early stopping, while the dot lines represent the performances after early stopping (over-annotation area). In the over-annotation area, LA-BALD consistently outperforms J-BALD. Compared with SS, LA-BALD either ends up with higher label accuracy with the same amount of annotation or reaches similar label accuracy but at a faster pace. The only exception is *Insect+Fungus*, where LA-BALD saturates after 0.4 number of annotations per image.

From Fig. 12 and 13, LA-BALD does not necessarily outperform all the baselines. However, when observing the performance across different datasets, LA-BAL still stands out. In Fig. 13, Rand-I+Greedy-W performs slightly better than LA-BALD in Insect + Fungus, but loses by a big margin compared to LA-BALD in Dog and Vertebrate. From the overall results, we see LA-BALD outperforms all baselines before the labelling algorithm stops (suggested by the confidence of each label). When the dataset curator over-annotates the images, LA-BALD is at least comparable, most of the time better, than other baselines.