Low-Shot Validation: Active Importance Sampling for Estimating Classifier Performance on Rare Categories Supplement

1. Training Details

As mentioned in Section 4.1, we train the SwAV and BYOL models on the 1% split of ImageNet as these pre-trained versions of these models are not provided by the authors. We also trained a SwAV model on the 10% split of iNaturalist that we created to evaluate on *iNat100K*. We will release the 10% split of iNaturalist with the rest of the code for the paper. Note that we attempt to train semi-supervised models that perform close to what is reported by the authors, but, for our experiments, it is more important that the models are representative of semi-supervised models trained for the target dataset (that is, they does not necessarily need to be state-of-the-art).

For SimCLRv2, we use the pretrained model provided by the authors trained on the 1% split of ImageNet [1].

For SwAV, we produced a semi-supervised model by fine-tuning the provided self-supervised weights on the 1% split of ImageNet using the author's official code for semi-supervised learning [2]. For evaluating on *iNat100K*, we modified the author's semi-supervised training code in the following manner: (1) we modified the data loader to read the 10% iNaturalist training split, and (2) we changed the input image size to 299 from 224.

For BYOL, we modified the linear fine-tuning code provided by the authors [3] to (1) fine-tune the whole network by unfreezing the backbone (propagating gradient updates to all layers of the model during training), (2) train for 50 epochs instead of 80, and (3) to train on the 1% split of ImageNet.

2. Estimator Robustness to Classifier Thresholds

Figure 9 from the main paper illustrates that labeled datasets sampled for a given model can be reused to compute F1 for different models that are performing the same classification task. This suggests that labeled datasets sampled by our algorithm can be reused effectively across model families. Here, we run a similar experiment on a model family constructed through varying the classifier threshold (the model score above which we label samples as positives).

We construct binary classifiers by 1) taking the 1000-way outputs from the models described in Section 4.1 of the paper, 2) constructing probabilities per class by applying a softmax transformation to the logits, 3) converting the probabilities to positive and negative class probabilities through one-vs-all classification, then 4) calculating predicted labels by checking whether the positive class probability is greater than a threshold t.

We use ACIS to estimate the F-Score for binary classifiers using a SwAV model on *ImageNet50K* using the threshold 0.1, producing a labeled dataset. We reuse this labeled dataset across a family of binary classifiers constructed using thresholds ranging from 0.02 to 0.5.

In Figure 1, the labeled datasets curated for a threshold of 0.1 effectively estimate F1 across the range of thresholds tested, though performance is slightly worse for higher classifier thresholds. Along with the results from Section 4.4 of the paper, these results suggest that it is possible to curate an actively labeled validation set that can effectively estimate F-score across a family of models.

3. Platt scaling vs isotonic regression

Figure 2 shows the result of replacing isotonic regression in our method with Platt scaling. As seen in this figure, Platt scaling performs worse than isotonic regression in our experiments. Unlike Guo et al. [1], who evaluated under fairly

[1] Model code: https://github.com/google-research/simclr. Training weights: https://console.cloud.google.com/storage/browser/simclr-checkpoints/simclrv2/finetuned_1pct

[2] https://github.com/facebookresearch/swav#evaluate-models-semi-supervised-learning-on-imagenet

[3] https://github.com/deepmind/deepmind-research/tree/master/byol#linear-evaluation

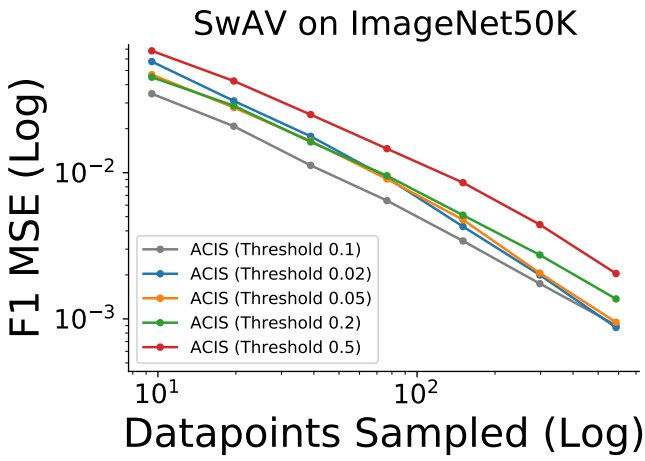

Figure 1: Validation sets curated for SwAV for a model with a given classifier threshold can be used to efficiently estimate F1 score of the model family generated by varying the classifier threshold. The validation set curated for a binary classifier on *ImageNet50K* using a threshold of 0.1 accurately estimates the F1 of the models generated by using thresholds ranging from 0.02 to 0.5. The labeled datasets transfer slightly better to models with lower classifier thresholds than to models with higher classifier thresholds.

balanced categories and calibrated using thousands of labeled examples, our work calibrates binary classifiers with severe class imbalance using only tens of labeled examples.

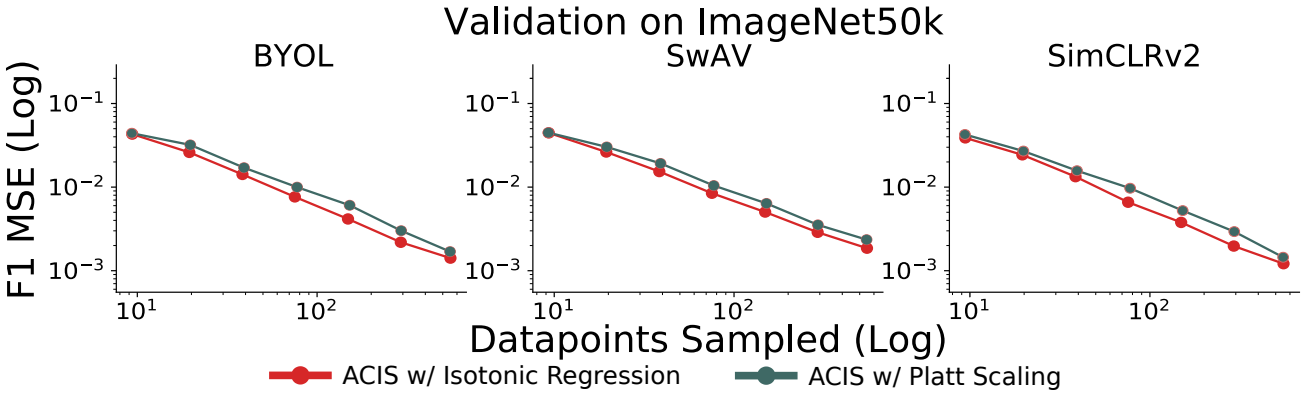

Figure 2: ACIS performs better with isotonic regression than with Platt scaling.

4. Variance Estimation

4.1. Averaging Across Iterations

In Section 3.4, we note that when combining the estimated $\hat{G}_j$'s, we estimate the variance of the weighted average $\hat{G}$ by taking a weighted average of the sampling variances of each iteration. While this assumes that there is no covariance between the $\hat{G}_j$'s, we find that this is a reasonable assumption in practice, and yields better estimates of the variance of $\hat{G}$ than the worst-case estimator (which assumes a covariance of 1). As seen in Figure 3, the worst-case estimator significantly overestimates the empirical variance, whereas taking a weighted average of the sampling variances across iterations yields an accurate estimator of the empirical variance.

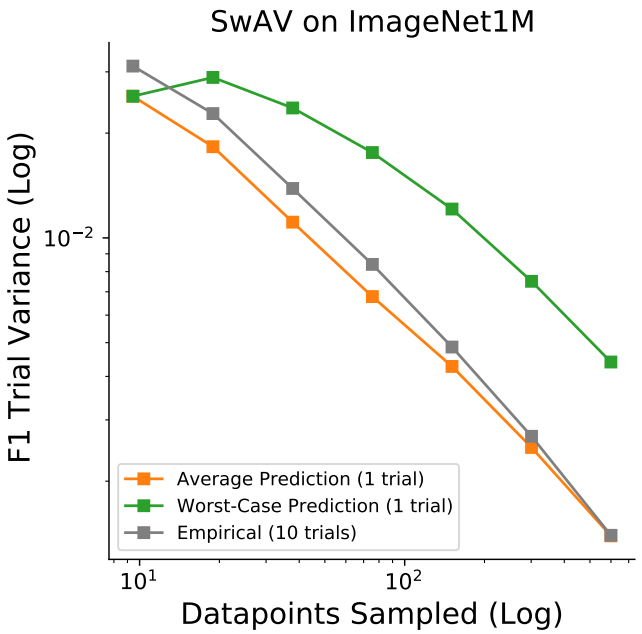

Figure 3: When estimating the F-score of a SwAV model trained on *ImageNet1M*, a weighted average of variance estimates across iterations for ACIS produces a variance estimate which is very similar to the observed empirical variance across trials. On the other hand, assuming a covariance of 1 across trials and estimating a worst-case variance for ACIS significantly overestimates the observed empirical variance.

4.2. Derivation of Estimator

In this section, we motivate the definition of our estimator

$$S^2_{n,q} = \frac{C^{-1} * \sum_{j=1}^{n} \left(w(x_j, y_j, M(x_j))^2 \left(\ell(M(x_j), y_j) - \hat{G}_{n,q} \right)^2 \right)}{\frac{1}{n} \left(\sum_{j=1}^{n} w(x_j, y_j, M(x_j)) \right)^2}$$

of the sampling variance introduced in Section 3.4. The derivation below largely follows that of Lemma 2 of Sawade et al. [2], correcting an error in their variance derivation. Let $q(x, y)$ be any distribution such that $q(x, y)$ can be factorized as $q_x(x)p(y|x)$ for some $q_x(x)$. As in the main paper, we use the notation $q(x)$ to denote $q_x(x)$ and similarly use $p(x)$ to denote the marginal distribution of x under p.

Define $\hat{G}_{n,q} = \frac{\sum_{j=1}^{n} \ell(x_j, y_j, \hat{y}_j) w(x_j, y_j, \hat{y}_j)}{\sum_{j=1}^{n} w(x_j, y_j, \hat{y}_j)}$, where $\{(x_j, y_j)\}_{j=1}^{n}$ are drawn IID according to $q(x, y)$ and $\hat{y}_j = M(x_j)$ [where M is the model]. We first claim that the asymptotic variance of $\hat{G}_{n,q}$ is

$$\frac{\int\int w(x, y, M(x))^2 \left(\ell(M(x), y) - G \right)^2 q(x, y) \, dy \, dx}{\left(\iint w(x, y, M(x)) q(x, y) \, dy \, dx \right)^2}$$

Proof: Define

$$\begin{aligned} v_j &= v(x_j, y_j, \hat{y}_j) = v(x_j, y_j, M(x_j)), \\ w_j &= \tfrac{p(x_j)}{q(x_j)} \cdot v_j, \\ \ell_j &= \ell(\hat{y}_j, y_j), \\ \hat{G}^0_{n,q} &= \sum_{j=1}^{n} \ell_j w_j, \\ W_n &= \sum_{j=1}^{n} w_j \ . \end{aligned}$$

Thus, $\mathbb{E}_q[\hat{G}^0_{n,q}] = \sum_{j=1}^n \mathbb{E}_q[\ell_j w_j] = n\mathbb{E}_q[\ell_j w_j]$.
$\mathbb{E}_q[\ell_j w_j] = \mathbb{E}_q\left[\frac{p(x_j)}{q(x_j)} v_j \ell_j\right] = \mathbb{E}_p[v_j \ell_j]$. Defining $G = \dfrac{\mathbb{E}_p[v_j \ell_j]}{\mathbb{E}_p[v_j]}$, we thus have $\mathbb{E}_q[w_j \ell_j] = G\,\mathbb{E}_p[v_j]$. Also, $\mathbb{E}_q[W_n] = n\mathbb{E}_q[w_j] = n\mathbb{E}_p[v_j]$. By the Central Limit Theorem, we thus have

$$\begin{aligned} \sqrt{n}\left(\frac{1}{n}\hat{G}^0_{n,q} - G\mathbb{E}_p[v_j]\right) &\xrightarrow{D} \mathcal{N}(0, \mathrm{Var}_q(w_j \ell_j)) \\ \sqrt{n}\left(\frac{1}{n} W_n - \mathbb{E}_p[v_j]\right) &\xrightarrow{D} \mathcal{N}(0, \mathrm{Var}_q(w_j)) \end{aligned}$$

Define $f(x, y) = \frac{x}{y}$, so $\nabla f(x, y) = \left(\frac{1}{y},\ -\frac{x}{y^2}\right)^T$. So, $f(G\mathbb{E}_p[v_j], \mathbb{E}_p[v_j]) = G$ and $\nabla f(G\mathbb{E}_p[v_j], \mathbb{E}_p[v_j]) = \left(\frac{1}{\mathbb{E}_p[v_j]}, -\frac{G\mathbb{E}_p[v_j]}{\mathbb{E}_p[v_j]^2}\right)^T$ $= \frac{1}{\mathbb{E}_p[v_j]}(1, -G)^T$. Applying the Delta Method to the random vector $\frac{1}{n}(\hat{G}^0_{n,q},\ W_n)^T$ and the function f, we thus have

$$\sqrt{n}\left(\hat{G}^0_{n,q}/W_n - G\right) \xrightarrow{D} \mathcal{N}\left(0, \tfrac{1}{\mathbb{E}_p[v_j]^2}(1, -G) \cdot \Sigma \cdot (1, -G)^T\right),$$

where $\Sigma = \begin{pmatrix} \mathrm{Var}_q(w_j \ell_j) & \mathrm{Cov}_q(w_j \ell_j, w_j) \\ \mathrm{Cov}_q(w_j \ell_j, w_j) & \mathrm{Var}_q(w_j) \end{pmatrix}$.
Now, $(1, -G) \cdot \Sigma \cdot (1, -G)^T =$

$$\begin{aligned} &(1, -G) \cdot \begin{pmatrix} \mathrm{Var}_q(w_j \ell_j) - G\mathrm{Cov}_q(w_j \ell_j, w_j) \\ \mathrm{Cov}_q(w_j \ell_j, w_j) - G\mathrm{Var}_q(w_j) \end{pmatrix} = \\ &\mathrm{Var}_q(w_j \ell_j) - 2G\mathrm{Cov}_q(w_j \ell_j, w_j) + G^2 \mathrm{Var}_q(w_j) = \\ &(\mathbb{E}_q[w_j^2 \ell_j^2] - \mathbb{E}_q[w_j \ell_j]^2) - 2G(\mathbb{E}_q[w_j^2 \ell_j] - \mathbb{E}_q[w_j \ell_j]\mathbb{E}_q[w_j]) \\ &+ G^2(\mathbb{E}_q[w_j^2] - \mathbb{E}_q[w_j]^2) = \\ &(\mathbb{E}_q[w_j^2 \ell_j^2] - 2G\mathbb{E}_q[w_j^2 \ell_j] + G^2 \mathbb{E}_q[w_j^2]) - \\ &(\mathbb{E}_q[w_j \ell_j]^2 - 2G\mathbb{E}_q[w_j \ell_j]\mathbb{E}_p[w_j] + G^2 \mathbb{E}_q[w_j]^2) = \\ &\mathbb{E}_q[w_j^2(\ell_j - G)^2] - (\mathbb{E}_q[w_j \ell_j] - G\mathbb{E}_q[w_j])^2 \,. \end{aligned}$$

Recall that $\mathbb{E}_q[w_j] = \mathbb{E}_p[v_j]$ and $\mathbb{E}_q[\ell_j w_j] = G\mathbb{E}_p[v_j]$, so $\mathbb{E}_q[w_j \ell_j] - G\mathbb{E}_q[w_j] = G\mathbb{E}_p[v_j] - G\mathbb{E}_p[v_j] = 0$. So, we finally obtain that the asymptotic variance is $\frac{1}{\mathbb{E}_p[v_j]^2}(1, -G) \cdot \Sigma \cdot (1, -G)^T = \frac{1}{\mathbb{E}_p[v_j]^2}\mathbb{E}_q[w_j^2(\ell_j - G)^2] =$

$$\frac{\iint w(x, y, M(x))^2 (\ell(M(x), y) - G)^2 \cdot q(x, y)\, dy\, dx}{\left(\iint p(x, y) v(x, y, M(x))\, dy\, dx\right)^2} = \frac{\iint w(x, y, M(x))^2 (\ell(M(x), y) - G)^2 \cdot q(x, y)\, dy\, dx}{\left(\iint w(x, y, M(x)) q(x, y)\, dy\, dx\right)^2},$$

since $\frac{p(x,y)}{q(x,y)} = \frac{p(x)}{q(x)}$. □

Define

$$\tilde{S}^2_{n,q} := \frac{\frac{1}{n}\sum_{j=1}^{n} w(x_j, y_j, M(x_j))^2 \left(\ell(M(x_j), y_j) - \hat{G}_{n,q}\right)^2}{\left(\frac{1}{n}\sum_{j=1}^{n} w(x_j, y_j, M(x_j))\right)^2} \tag{1}$$

By Slutsky's theorem and the preceding derivation, $\tilde{S}^2_{n,q}$ is a consistent estimator of the asymptotic variance of $\hat{G}_{n,q}$.

Now, let $C = 1 - \frac{\sum_{j=1}^{n} w(x_j, y_j, M(x_j))^2}{(\sum_{j=1}^{n} w(x_j, y_j, M(x_j)))^2}$. By Slutsky's theorem, $\frac{\frac{1}{n}\sum_{j=1}^{n} w(x_j, y_j, M(x_j))^2}{n\left(\frac{1}{n}\sum_{j=1}^{n} w(x_j, y_j, M(x_j))\right)^2}$ converges in probability to 0 as $n \to \infty$, and hence C converges in probability to 1. Thus, $S^2_{n,q} = \frac{\tilde{S}^2_{n,q}}{C}$ is also a consistent estimator of the asymptotic variance of $\hat{G}_{n,q}$, by applying Slutsky's theorem one final time.

The factor C can be interpreted analogously to the Bessel correction for sample variance. Indeed, if the w_j's are all equal to 1 (so $\hat{G}_{n,q}$ is simply the mean of the ℓ_j's), we have $C = 1 - \frac{n}{n^2} = \frac{n-1}{n}$ and $\tilde{S}^2_{n,q} = \frac{\sum_{j=1}^{n}(\ell_j - \hat{G}_{n,q})^2}{n}$, so $S^2_{n,q} = \frac{1}{n-1}\sum_{j=1}^{n}(\ell_j - \hat{G}_{n,q})^2$, which is the same as the standard sample variance calculation using Bessel's correction.

References

[1] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org, 2017. 1

[2] Christoph Sawade, Niels Landwehr, and Tobias Scheffer. Active estimation of f-measures. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, NeurIPS'10, page 2083–2091. Curran Associates Inc., 2010. 3