A. Understanding Selection Frequency

Understanding Selection Frequencies In Figure 8 we randomly sample images while varying selection frequency. Here, the straightforwardness of identifying images correlates with increasing selection frequency (e.g. all the 36/36 selection frequency images clearly identify with their corresponding class, while some of the 0/36 selection frequency images appear to be mislabeled).

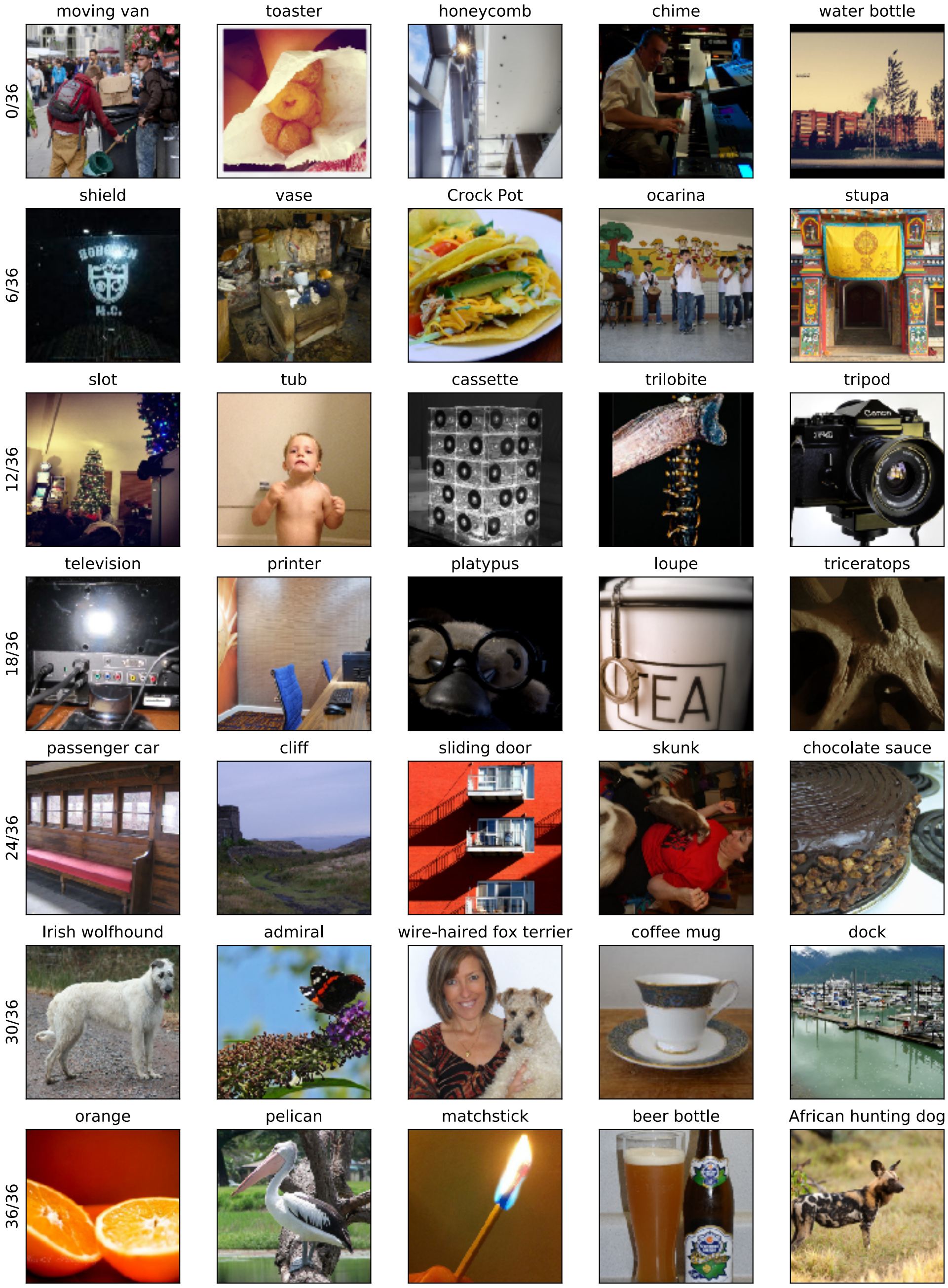


Figure 8. Randomly drawn images from `v1`, varying selection frequency.

B. Experimental Setup

Here, we provide more detail of the experimental setup. We first lay out the setup of our Mechanical Turk experiments for remeasuring selection frequency (B.1), and highlight the subtle differences between our setup and that of Recht et al. (2019b) (B.2). In Appendix D we discuss our analysis of the original data and algorithm of Recht et al. (2019b) showing the existence of bias in that setting.

B.1. Selection frequency remeasuring experiment

In Section 3, we replicate the ImageNet experiment to remeasure selection frequencies for the ImageNet v1 and v2 datasets. We present annotators with a grids of 48 images along with an ImageNet class. The annotators are also provided with the WordNet synsets for the ImageNet class being queried, along with a Wikipedia link and asked to selected all images containing instances of that object (ignoring clutter as per the original dataset creation process). The 48 images in each grid consist of: (a) 10 ImageNet v1 validation set images from that class, (b) 10 ImageNet v2 validation set images from that class, (c) 22 related Flickr images scraped from Flickr (using the exact script and queries described in Recht et al. (2019b)) and (c) 6 negative control images (three corresponding to randomly chosen labels, and 3 corresponding to the "nearest" label to the true label in terms of WordNet path similarity).

We implemented our setup by modifying the code made publicly available by Recht et al. (2019b) [10]. A screenshot of our interface appears in Figure 10. Each such grid of images is shown to 40 annotators. For each image-class pair, we can then compute the "selection frequency" based on how often it was selected by the annotators.

Deployment Details. There are a number of deployment details that could cause variations in results. We compensated MTurk workers with \$0.23 per assignment (i.e., each completed grid), which we calibrated to pay a rate of at least \$9/hr for most workers. To collect 40 separate MTurk annotations for each submitted grid of images, we obtained 10 annotations on 4 different dates and times, all within the span of a single week. We placed qualification requirements on the workers allowed to complete assignments. Specifically, we filtered for workers that (a) agree to view adult content (as some ImageNet images have content like nudity or gore) and (b) have a larger than 95% assignment approval rate (as to ensure the quality of the results).

Controls. All of the results presented in this work were run on a "clean" and "raw" version of our data, i.e., without and with data cleaning respectively. We find that the inclusion of data cleaning makes the observed gap between `v1` and `v2` slightly larger but otherwise does not have a significant effect on results.

Our data cleaning process is as follows: a given batch is "flagged" if: (a) there are less than 6 selected images out of the total 48, or (b) more than one of the negative controls was selected. We only omit data, however, from workers whose batches were rejected at a rate of 30% or higher (e.g., if an annotator completed 30 batches, but more than 10 of them are flagged to be low-quality, then all of the annotator's data is omitted). Finally, to make computing of the statistics easier, we evened out the number of annotators per image to equal the minimum number of remaining assignments per image, which was 36 (compared to 40 originally) by randomly discarding annotations. In total, the entire process corresponds to discarding 10% of the annotations.

B.2. Comparison to the original setup

Recht et al. (2019b) measure the average selection frequency of `v1` to be 0.71, whereas our experiment measures the average `v1` selection frequency to be 0.85. While our experiments were modeled closely after that of Recht et al. (2019b) (and in fact use the same core codebase to minimize discrepancies in task presentation/inferace), we made a few changes to the setup to ensure high data quality. We hypothesize that these changes, discussed below, are what result in the discrepancy between the measured average selection frequencies. However, since these changes are applied at the task level and annotators are not told which dataset each image is sourced from, we find it unlikely that these changes would affect annotations for one dataset more than the other. Furthermore, in Appendix D, we demonstrate that the bias identified in this paper can be found even using the original data collected by Recht et al. (2019b).

Worker pay and qualifications. In our experiment, we paid annotators 20 cents per set of 48 images completed—this was informed by the average time taken to complete a batch, and was calibrated so that the task paid approximately 12 dollars per hour. Conversely, the original experiment of Recht et al. (2019b) pay 10 cents per batch. Although worker pay usually

[10] `https://github.com/modestyachts/ImageNetV2`

has only a mild impact on worker reliability on MTurk (Mason & Watts, 2009; Buhrmester et al., 2011), higher worker pay has been recognized as a tool to boost participation rates (Buhrmester et al., 2011) and requester reputation for future experiments (Paolacci et al., 2010).

Perhaps the most important modification made was our inclusion of worker qualifications, which only allow annotators who have had 95% or more of previous tasks accepted to participate in our task. Prior work has shown that without these worker qualifications, crowdsourced data tends to be of significantly worse quality. For example, Peer et al. (2013) report that 2.6% of workers with the "95% accepted" qualification failed an "attention-check test," compared to 33.9% of workers without qualifications[11]. We should therefore expect a significant increase in annotation reliability (and so in turn some discrepancy) from using worker qualifications.

Makeup of each batch. Another difference between the two experiments is that in our experiment, each batch of images contains 10 images from ImageNet-v1, and 10 images from ImageNet-v2, in order to ensure that we could obtain 40 annotations for each `v1` and `v2` image while keeping to a reasonable budget constraint. The experiment of Recht et al. (2019b) uses only three images from ImageNet-v1 per batch (and a variable number of ImageNet-v2 images, since the dataset was not yet realized). Thus, the grids presented in our experiment contain images that are on average more likely (*a priori*) to be selected. This could in part contribute to the higher average selection frequency that we observe (though again, we would expect this effect to apply to both datasets and thus preserve the observed selection frequency gap).

Randomization. Response-order bias is a well-documented phenomenon in literature on crowdsourcing (e.g. (Schuman & Presser, 1981)), although its effects in the domain of image selection are not well-understood yet. In our experiment, we randomize the order of the images per batch per worker (i.e., we used JavaScript to randomize the image order on page load) to mitigate the potential effects of this bias. In the prior experiment, however, the image order is deterministic, and thus the study may have response-order effects.

Worker duplicates. Due to the mechanism by which images were distributed to assignments in the study of Recht et al. (2019b), 5.6% of the annotations are duplicated (i.e., 5.6% of the [worker, image] pairs collected are non-unique), with approximately 3% of the annotations being redundant (unlike the preceding number, this fraction does not count the "original copy" of each non-unique pair). A histogram of the number of times a single worker labeled a single image is shown in Figure 9. Since duplicate workers violate sample indpendence and can skew measured selection frequencies for some images, in our study we ensure that no worker labels the same image more than once.

Data cleaning and controls. Our study also differs in having built-in mechanisms for data cleaning (as discussed in the last section), allowing us to run all of our experiment on the "cleaned" and "raw" versions of our data. These results tend to not be substantially different (for the cleaned data, the selection frequency gap we measure between `v1` and `v2` slightly increases from 4.5% to 4.6%). Possible reasons for this similarity between cleaned and raw results include any of the quality control protocols outlined in this section.

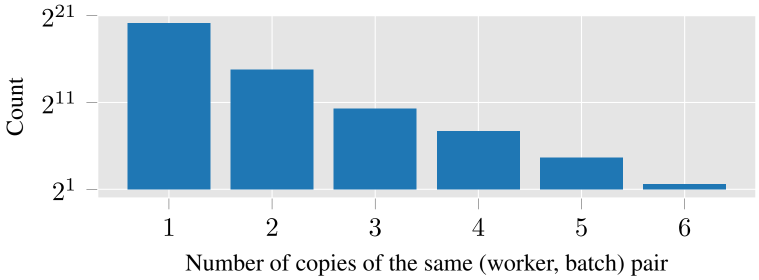
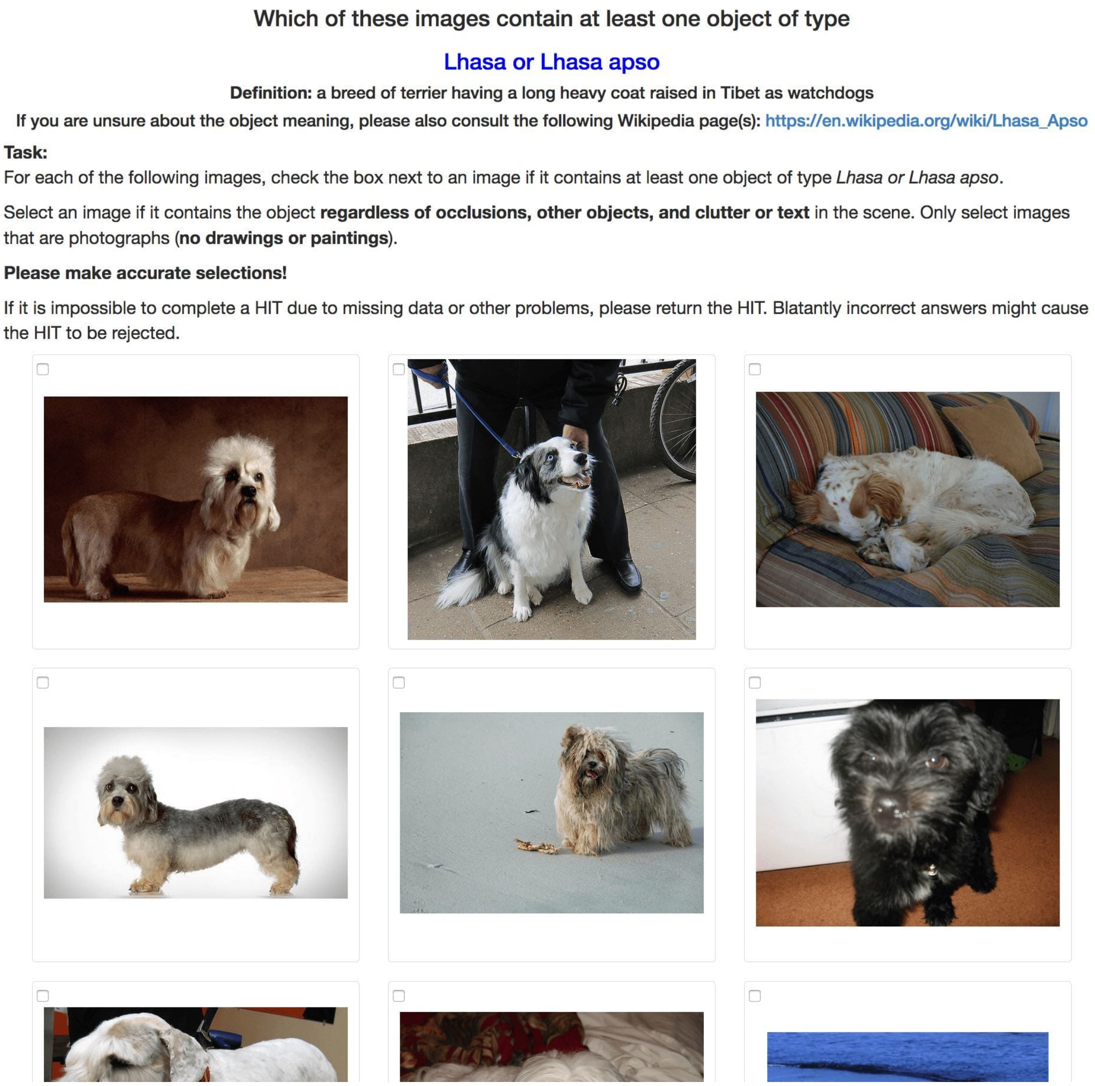


Figure 9. A histogram showing the existence of duplicate (worker, batch) pairs present in the original collected data. Each point in the histogram is a unique (worker, batch) pair, and the x axis corresponds to the number of times that pair is observed in the dataset.

[11] Attention-check tests are a series of three attention-check questions (ACQs). ACQs are questions with right/wrong answers unrelated to the task meant to gauge an annotator's attentiveness, e.g. "Have you ever had a fatal heart attack?". In the Peer et al. (2013) study, 16.4% of unqualified workers reported that they had suffered a *fatal* heart attack, compared to 0.4% of qualified workers.



Which of these images contain at least one object of type

Lhasa or Lhasa apso

Definition: a breed of terrier having a long heavy coat raised in Tibet as watchdogs

If you are unsure about the object meaning, please also consult the following Wikipedia page(s): https://en.wikipedia.org/wiki/Lhasa_Apso

Task:

For each of the following images, check the box next to an image if it contains at least one object of type *Lhasa or Lhasa apso*.

Select an image if it contains the object **regardless of occlusions, other objects, and clutter or text** in the scene. Only select images that are photographs (**no drawings or paintings**).

Please make accurate selections!

If it is impossible to complete a HIT due to missing data or other problems, please return the HIT. Blatantly incorrect answers might cause the HIT to be rejected.

Figure 10. Screenshot of a sample grid to measure selection frequencies (Section 3). Annotators are given a grid of 48 images for a specific label and asked to select all images that contain that label. The interface and instructions are based on Recht et al. (2019b).

C. Theoretical Results

In this section we show the series of calculations used in Section 2 to attain the result in Equation 1, i.e. the bias incurred by a matching procedure in the toy model.

Recall that in our setup we have $p_{flickr}(s)$ and $p_1(s)$ given by Beta(α, β) and Beta$(\alpha+1, \beta)$ respectively, and that $\hat{s}(x)$ is given by first sampling $s \sim p_i(s)$ then sampling n Bernoulli trials with success probability s. We noted in Section 2 that the distribution of $s(x)$ induced by matching $p_{flickr}(s)$ and $p_1(s)$ based on samples of $\hat{s}(x)$ is given by:

$$\begin{aligned} p_{flickr}(s(x)) \cdot \mathbb{P}(\text{x is accepted}|s(x)) &= p_{flickr}(s(x)) \cdot \int_{\hat{s}} p(\hat{s}|s) \mathbb{P}(\text{x is accepted}|\hat{s}(x)) \\ &= p_{flickr}(s(x)) \cdot \int_{\hat{s}} p(\hat{s}|s) \frac{p_1(\hat{s}(x))}{p_{flickr}(\hat{s}(x))} \end{aligned}$$

Now, note that by construction, $\hat{s}(x)$ is distributed according to the beta-binomial distribution[12], and thus (a) has support $\{0, \ldots, n\}$; and (b) induces the following closed-form probability mass function for $p_{flickr}(\hat{s})$ ($p_1(\hat{s})$ can be written analogously, with $\alpha+1$ replacing α) :

$$p_{flickr}(\hat{s}(x) = k) = \binom{n}{k} \frac{B(k+\alpha, n-k+\beta)}{B(\alpha, \beta)},$$

where

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)},$$

and Γ is the Gamma function—for simplicity we will assume that $\alpha, \beta \in \mathbb{N}$ and so $\Gamma(x) = (x-1)!$. Thus, returning to the full induced density:

$$\begin{aligned} & p_{flickr}(s(x)) \cdot \int_{\hat{s}} p(\hat{s}|s) \frac{p_1(\hat{s}(x))}{p_{flickr}(\hat{s}(x))} \\ &= p_{flickr}(s(x)) \cdot \sum_{k=0}^{n} p(\hat{s}|s) \frac{p_1(\hat{s}(x))}{p_{flickr}(\hat{s}(x))} \\ &= p_{flickr}(s(x)) \cdot \sum_{k=0}^{n} \left[\binom{n}{k} s^k (1-s)^{n-k} \right] \frac{\left[\frac{\binom{n}{k} B(k+\alpha+1, n-k+\beta)}{B(\alpha+1, \beta)} \right]}{\left[\frac{\binom{n}{k} B(k+\alpha, n-k+\beta)}{B(\alpha, \beta)} \right]} \\ &= \left[\frac{s^{\alpha-1} \cdot (1-s)^{\beta-1}}{B(\alpha, \beta)} \right] \cdot \sum_{k=0}^{n} \left[\binom{n}{k} s^k (1-s)^{n-k} \right] \frac{B(k+\alpha+1, n-k+\beta)}{B(k+\alpha, n-k+\beta)} \cdot \frac{B(\alpha, \beta)}{B(\alpha+1, \beta)} \\ &= \left[\frac{s^{\alpha-1} \cdot (1-s)^{\beta-1}}{B(\alpha+1, \beta)} \right] \cdot \sum_{k=0}^{n} \left[\binom{n}{k} s^k (1-s)^{n-k} \right] \frac{B(k+\alpha+1, n-k+\beta)}{B(k+\alpha, n-k+\beta)} \end{aligned}$$

Now in general, note that

$$\frac{B(x+1, y)}{B(x, y)} = \frac{\frac{\Gamma(x+1)\Gamma(y)}{\Gamma(x+y+1)}}{\frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}} = \frac{\Gamma(x+1)}{\Gamma(x)} \cdot \frac{\Gamma(x+y)}{\Gamma(x+y+1)} = \frac{x}{x+y} \qquad \text{for } x, y \in \mathbb{N}.$$

[12] https://en.wikipedia.org/wiki/Beta-binomial_distribution

Applying this identity to the above and continuing to simplify:

$$
\begin{aligned}
&= \left[\frac{s^{\alpha-1} \cdot (1-s)^{\beta-1}}{B(\alpha+1,\beta)}\right] \cdot \sum_{k=0}^{n} \left[\binom{n}{k} s^k (1-s)^{n-k}\right] \frac{k+\alpha}{n+\alpha+\beta} \\
&= \left[\frac{s^{\alpha-1} \cdot (1-s)^{\beta-1}}{B(\alpha+1,\beta)}\right] \cdot \mathbb{E}_{k\sim \text{Binomial}(n,s)} \left[\frac{k+\alpha}{n+\alpha+\beta}\right] \\
&= \left[\frac{s^{\alpha-1} \cdot (1-s)^{\beta-1}}{B(\alpha+1,\beta)}\right] \cdot \frac{n \cdot s+\alpha}{n+\alpha+\beta} \\
&= \frac{n}{n+\alpha+\beta} \cdot \frac{s^{(\alpha+1)-1} \cdot (1-s)^{\beta-1}}{B(\alpha+1,\beta)} + \frac{\alpha}{n+\alpha+\beta} \cdot \frac{s^{\alpha-1} \cdot (1-s)^{\beta-1}}{B(\alpha+1,\beta)} \\
&= \frac{n}{n+\alpha+\beta} \cdot \frac{s^{(\alpha+1)-1} \cdot (1-s)^{\beta-1}}{B(\alpha+1,\beta)} + \frac{\alpha}{n+\alpha+\beta} \cdot \left(\frac{\alpha+\beta}{\alpha} \cdot \frac{B(\alpha+1,\beta)}{B(\alpha,\beta)}\right) \cdot \frac{s^{\alpha-1} \cdot (1-s)^{\beta-1}}{B(\alpha+1,\beta)} \\
&= \frac{n}{n+\alpha+\beta} \cdot \text{Beta}(\alpha+1,\beta) + \frac{\alpha+\beta}{n+\alpha+\beta} \cdot \text{Beta}(\alpha,\beta)
\end{aligned}
$$

which matches precisely the result shown in Section 2.

D. Analysis of Original Data

In Section 3, we remeasured selection frequencies using a new Mechanical Turk experiment. Here we set out to verify the existence of the hypothesized bias in the original collected data.

We reimplemented the sampling component of the algorithm exactly as described in (Recht et al., 2019b) and the corresponding code release, using the `pandas` Python package. The source code is available in our code release, along with a serialized version of the data collected by Recht et al. (2019b)[13]. As a sanity check, we verified that all of the results hold using the exact code published by Recht et al. (2019b)[14]

D.1. Sampled dataset accuracy increases with more workers

We begin by showing that the accuracy we observe on ImageNet-v2 depends on the number of workers used to sample the dataset. We gradually decrease the number of workers n used in computing observed selection frequencies to study the effect of noise on statistic matching. We find that model accuracy on the resulting replicated dataset degrades as n decreases. For example, the accuracy gap from `v1` to the replication increases from 12% when $n = 10$, to 14% when $n = 5$. This is consistent with our model of statistic matching bias: fewer annotators means noisier observed selection frequencies $\hat{s}_n(x)$, which in turn amplifies the effect of the bias, driving down model accuracies.

Methodology. Specifically, we use the frequency-adjusted accuracy introduced in Section 5.1, to estimate model accuracy on a version of the candidate pool reweighted to have the same selection frequency distribution as ImageNet-v1:

$$\hat{\mathcal{A}}^n_{\mathcal{D}_{flickr}|s_1} = \sum_{k=0}^{n} \mathbb{E}_{x_{flickr}\sim\mathcal{D}_{flickr}} \left[f(x_{flickr}) \middle| \hat{s}_n(x_{flickr}) = \frac{k}{n} \right] \cdot p_1 \left(\hat{s}_n(x_1) = \frac{k}{n} \right). \tag{6}$$

This estimator is analogous to the ImageNet-v2 selection process of Recht et al. (2019b), but operates by reweighting the candidate pool rather than filtering it.

We plot this estimator in Figure 11, varying n from 5 to 10. We find that the gap between the adjusted accuracy and ImageNet accuracy shrinks as n grows, until shrinking to (and not plateauing at) 12.3% at $n = 10$. This behavior is predicted by statistic matching bias, and suggests that in the infinite-annotator limit the ImageNet-v2 accuracy is higher. (Ideally, we could estimate the infinite-annotator limit using the data of Recht et al. (2019b), but 10 annotators is too few to get a reliable estimate.)

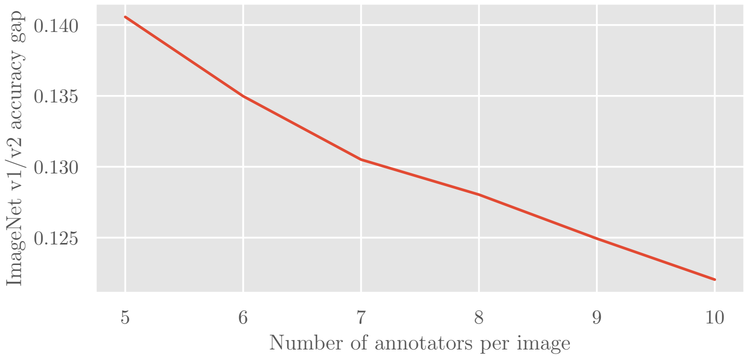


Figure 11. The frequency-adjusted accuracy gap between ImageNet-v1 and ImageNet-v2, using a varying number of annotators to estimate selection frequencies. The gap continually decreases, and does not plateau at 10 annotators. Bootstrapped 95% confidence intervals are shown (shaded).

[13] https://github.com/MadryLab/dataset-replication-analysis

[14] Since we only study the sampling component, we opt to rewrite a specialized script that has the benefit of being significantly shorter, simpler, and faster.

D.2. Measuring the selection frequency gap using held-out workers

Recall that most images in Recht et al. (2019b) have (at least) 10 annotations per image, and that to get an unbiased estimate of the selection frequency we must "hold out" some annotations per image (e.g. we should not use the same annotations for both matching and measuring selection frequencies). To that end, we split the annotations for each candidate image into a "train" set and a "test" set. We then mimic the original `v2` creation process using only the train set of annotations, and use the remaining test annotations for each image to obtain a held-out measurement of selection frequency.

Since Recht et al. (2019b) collect 10 annotations for most candidate images, and since the original buckets used in the matching process are split by boundaries $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$, we use 5 annotations[15] for each image in the train set and reserve the rest for independently measuring the held-out frequency.

The results of this experiment are given in Table 1. We repeat the experiment using both 5 and 10 annotators to estimate ImageNet-v1 selection frequencies. In both cases, the average selection frequency of the in-sample images overestimates the heldout (true) selection frequency by 2-3%, and the resulting replicated dataset has lower selection frequency than ImageNet-v1.

Table 1. Average in-sample and heldout selection frequencies for the experiment described in D.2—the top (bottom) table presents the result of using 10 (5) annotators per image to estimate ImageNet-v1 selection frequencies. We use five annotators per image to estimate selection frqeuencies, then use the filtering algorithm of Recht et al. (2019b) to get a replicated dataset meant to match the selection frequency distribution of ImageNet-v1. The results show that (a) bias results in the average selection frequency of the new sample being *lower* than that of ImageNet-v1, and that (b) the bias is undetectable without heldout samples.

	ImageNet-v1	**Sampled replication**
Average selection frequency	0.71	0.71
Heldout selection frequency	0.71	0.69

	ImageNet-v1	**Sampled replication**
Average selection frequency	0.71	0.73
Heldout selection frequency	0.71	0.70

Effect size. The difference in held-out mean selection frequencies between `v1` and `v2` here is smaller than the one we observe in the newly collected data. However, as discussed in Section 3, the size of the average gap is not necessarily predictive of the size of the accuracy correction. The latter depends on the distributional difference between true `v1` and `v2` selection frequencies, rather than on just their first moments. In particular, observing different mean selection frequencies for `v1` and `v2` is a sufficient but not necessary condition for there to be an accuracy gap[16].

Unfortunately, getting an accurate estimate of the gap on the original data seems impossible: first, there are insufficient workers to reliably apply any of our techniques from Section 5. Furthermore, with k held-out workers, we can only estimate the first k moments of the true selection frequency distributions $p_i(s)$, even if we had infinitely many images. So the problem seems largely underdetermined.

We can, however, show that original data is plausibly consistent with the 4% accuracy gap estimated using the new data (i.e., that such a gap is not ruled out). Specifically, in D.1, the original accuracy difference between Flickr and ImageNet-v1 was 20.8%. When using five (ten) annotators per image, this gap shrunk to 14.0% (12.2%). In the newly collected data, the gaps for the original distributions, 5-annotator adjustment, and 10-annotator adjustment are 11.7%, 8.5%, and 7.3% respectively (again, by using the same reweighting scheme as Appendix D.1). Therefore, accuracy adjustments incurred by using more workers on the original data are significantly (about two times) larger than the corresponding accuracy adjustments on the newly collected data, and so we expect to see a larger total correction than our estimated 8.1% correction.

[15] We choose 5 annotations specifically since, as in the 10-annotation case, images fall into the same relative locations in each bin—other choices of annotations per image are severely affected by binning effects.

[16] As a concrete example, suppose that 50% of the annotators used were low-quality and did not complete the task (i.e., selected no images)—this would artificially shrink the mean selection frequency gap by 50%, but would not affect the accuracy adjustment.

D.3. Source selection frequencies determine sampled dataset accuracy

We next explore how the choice of source distribution impacts the resulting sampled dataset. We use a setup similar to that of the last experiment, in which we use five workers for the selection process. Then, using four hold-out samples from each image, we create a new candidate data pool called Flickr-E (Flickr-Easy) by including only the images which at least two out of the four heldout workers selected.

We then perform 5-worker statistic matching, both from Flickr and from Flickr-E to ImageNet-v1. In the absence of bias, the source distribution should not affect the accuracy of the resulting classifier. In contrast, we find that the dataset replication obtained from Flickr-v2 has comparable (within 0.2%) average selection frequency, but significantly higher accuracy (by ~3%) than the replication obtained from the unfiltered candidate pool (62%). This discrepancy further corroborates the hypothesis that ImageNet-v2 accuracies are impacted by the statistical bias that we identify in this work.

E. Non-parametric Adjusted Accuracy Estimation

In Section 5 we explore various methods of estimating the adjusted accuracy $\mathcal{A}_{\mathcal{D}_2|s_1}$ from the observable $\hat{s}$ samples. Section 5.1 presents the *naïve estimator*, computed by using $\hat{s}$ directly in place of s in the formula for $\mathcal{A}_{\mathcal{D}_2|s_1}$. In Section 5.2, we show that using the statistical jackknife, we can estimate and account for the bias in the naïve estimator to better estimate the true adjusted accuracy. Here, we first justify the application of the jackknife in Section 5.2, by proving the consistency of the estimator and that bias is roughly linear in (and in fact underestimated by) $1/n$.

E.1. Justifying the use of the statistical jackknife

Recall that in Section 5, the quantity of interest is the following *adjusted accuracy*:

$$\mathcal{A}_{\mathcal{D}_2|s_1} = \int_s \mathbb{E}_{\mathcal{D}_2}[f(x)|s(x) = s] \cdot p_1(s)\, ds. \tag{7}$$

In Section 5.1, we introduced the following naïve estimator

$$\hat{\mathcal{A}}^n_{\mathcal{D}_2|s_1} = \sum_{k=0}^{n} \mathbb{E}_{x_2\sim\mathcal{D}_2}\left[f(x_2)|\hat{s}_n(x_2) = \frac{k}{n}\right] \cdot p_1\left(\hat{s}_n(x_1) = \frac{k}{n}\right), \tag{8}$$

which evaluates to $\mathcal{A}_{\mathcal{D}_2|s_1}$ if $\hat{s}_n(x) = s(x)$, and assuming the expectations above can be computed exactly.

Consistency of the naïve estimator. In order for our application of the statistical jackknife to be valid, we must show that the naïve estimator is a *consistent* estimator of $\mathcal{A}_{\mathcal{D}_2|s_1}$—that is, as $n \to \infty$, $\hat{\mathcal{A}}^n_{\mathcal{D}_2|s_1} \to \mathcal{A}_{\mathcal{D}_2|s_1}$. Note that since we operate in the regime where the number of distinct images m greatly exceeds the number of annotators per image n, we will assume that the expectations above can be computed exactly. Note that the estimator remains consistent even if this is not the case, with the additional constraints that $m \to \infty$ and $m/n \to \infty$, but this greatly complicates the proof and we will show empirically that the estimator is robust to changes in m in the relevant regime.

Note that in the "infinite m" regime, the variance of the naïve estimator is 0. Thus, all of the error is due to bias in the estimator. In the following, we assume that $p_1(s)$, $p_2(s)$, and $p_2(s|f = 1)$ are continuous differentiable densities bounded away from zero and with bounded derivatives ($|d^r/dx^r\; p_i(x)| < C$).

$$\begin{aligned}
p_i\left(\hat{s}_n(x) = \frac{k}{n}\right) &= \int_0^1 p_i(s) \cdot \text{Binom}\left(n, n, k, s\right)\; ds && \text{(9)}\\
&= \int_0^1 p_i(s) \cdot \binom{n}{k} \cdot s^k \cdot (1-s)^{n-k}\, ds && \text{(10)}\\
&= \int_0^1 p_i(s) \cdot \frac{s^k(1-s)^{n-k}}{(n+1)\cdot B(k+1, n-k+1)}\, ds \qquad B(\cdot,\cdot) \text{ is the Euler beta function} && \text{(11)}\\
&= \frac{1}{n+1}\int_0^1 p_i(s) \cdot \text{Beta}(s; k+1, n-k+1)\, ds && \text{(12)}\\
&= \frac{1}{n+1}\mathbb{E}_{s\sim\text{Beta}(\cdot;k+1,n-k+1)}[p_i(s)] && \text{(13)}
\end{aligned}$$

Using a Taylor expansion of $p_i(s)$ around $\mathbb{E}_{s\sim\text{Beta}(\cdot;k+1,n-k+1)}[s]$, we can bound the above expression:

$$\begin{aligned}
&= \frac{1}{n+1}\mathbb{E}\left[p_i(\mathbb{E}[s]) + (s - \mathbb{E}[s])p_i'(E[s]) + \frac{(s-\mathbb{E}[s])^2}{2}p_i''(\mathbb{E}[s]) + \sum_{r=3}^{\infty}\frac{(s-\mathbb{E}[s])^r}{r!}p_i^{(r)}(\mathbb{E}[s])\right]\\
&= \frac{1}{n+1}\left(p_i(\mathbb{E}[s]) + \frac{1}{2}\text{Var}[s] \cdot p_i''(\mathbb{E}[s]) + O\left(\frac{1}{n^{7/2}}\right)\right)\\
&= \frac{1}{n+1}p_i\left(\frac{k+1}{n+2}\right) + \frac{(k+1)(n-k+1)}{(n+1)(n+2)^2(n+3)} \cdot p_i''(\mathbb{E}[s]) + O\left(\frac{1}{n^{7/2}}\right)
\end{aligned}$$

Now, using the presumed boundedness of derivatives we can write:

$$\left|p_i\left(\hat{s}_n(x) = \frac{k}{n}\right) - \frac{1}{n+1}\cdot p_i\left(\frac{k+1}{n+2}\right)\right| \leq \frac{(k+1)(n-k+1)}{(n+1)(n+2)^2(n+3)} \cdot p_i''(\mathbb{E}[s]) + O\left(\frac{1}{n^{7/2}}\right) \tag{14}$$

$$
\begin{aligned}
|\mathcal{A}_{\mathcal{D}_2|s_1} - \hat{\mathcal{A}}^n_{\mathcal{D}_2|s_1}| &= p_2(f(x)=1)\left|\int_0^1 \frac{p_2(s|f(x)=1)}{p_2(s)} p_1(s)\,ds - \sum_{k=0}^{n} \frac{p_2(\hat{s}_n(x)=\frac{k}{n}|f(x)=1)}{p_2(\hat{s}_n(x)=\frac{k}{n})} p_1(\hat{s}_n(x)=\frac{k}{n})\right| \\
&\leq \left|\int_0^1 \frac{p_2(s|f(x)=1)}{p_2(s)} p_1(s)\,ds - \frac{1}{n+1}\sum_{k=0}^{n} \frac{p_2\left(\frac{k+1}{n+2}|f(x)=1\right)}{p_2\left(\frac{k+1}{n+2}\right)} p_1\left(\frac{k+1}{n+2}\right)\right| \\
&+ \left|\frac{1}{n+1}\sum_{k=0}^{n} \frac{p_2\left(\frac{k+1}{n+2}|f(x)=1\right)}{p_2\left(\frac{k+1}{n+2}\right)} p_1\left(\frac{k+1}{n+2}\right) - \frac{p_2\left(\hat{s}_n(x)=\frac{k}{n}|f(x)=1\right)}{p_2\left(\hat{s}_n(x)=\frac{k}{n}\right)} p_1\left(\hat{s}_n(x)=\frac{k}{n}\right)\right|
\end{aligned}
$$

Now, note that the first term in the above is simply the error in the Riemmann sum approximation of the integral, which vanishes as $n \to \infty$. The second term is bounded by n times the error in each individual term of the sum, which we bounded as $O(n^{-2})$ in Equation (14).

Near-linearity of bias and likely underestimation. Recall that for the statistical jackknife to yield a reliable estimate of the adjusted accuracy, the bias in the naïve estimator must be analytic in $\frac{1}{n}$, and in particular should be dominated by a $O(\frac{1}{n})$ term (as this corresponds to precisely the term accounted for by the jackknife). In Figure 12 we show that the bias estimated by our jackknife procedure is indeed roughly linear in $1/n$, but grows slightly faster than $\frac{1}{n}$, likely leading the jackknife to provide an underestimate.

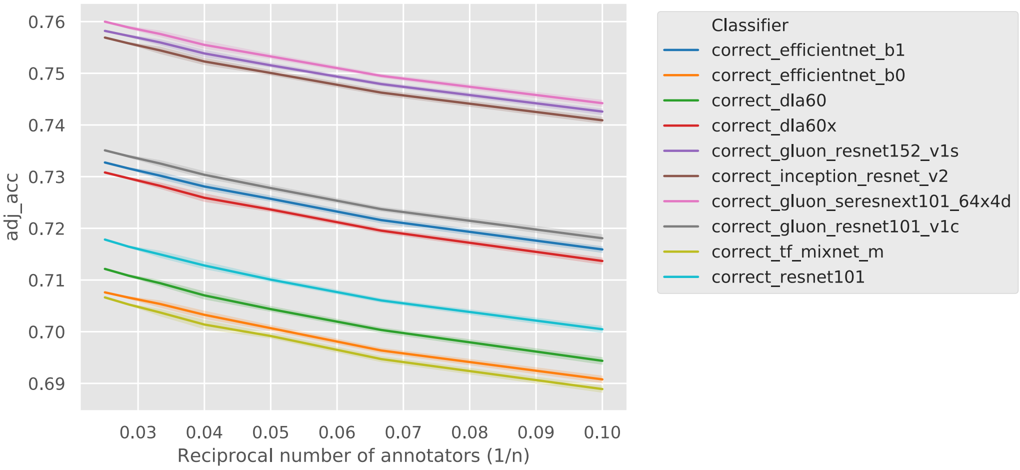


Figure 12. We plot the jackknife adjusted accuracy estimators of 10 models. On the y axis is shown the value of the n-sample jackknife estimate, with $1/n$ on the x axis. The fact that the plot is nearly linear suggests that our bias is indeed dominated by a $O(1/n)$ term, thus further justifying our use of the statistical jackknife in Section 5.2. Furthermore, the slightly accelerating slope as one moves left on the plot indicates that any error in the jackknife estimate is likely to be an underestimation of bias, rather than an overestimation.

F. Model Fitting

In this section, we describe our methods for parametric modeling.

F.1. Confidence Intervals

To construct 95% confidence intervals we perform 450 bootstrapped estimates (over the included images) of adjusted accuracy for each classifier. We then plot the 2.5% and 97.5% percentiles from the bootstrap estimates as the confidence intervals for each classifier.

F.2. Varying Annotators

We plot the results of varying the number of annotators in Figure 13.

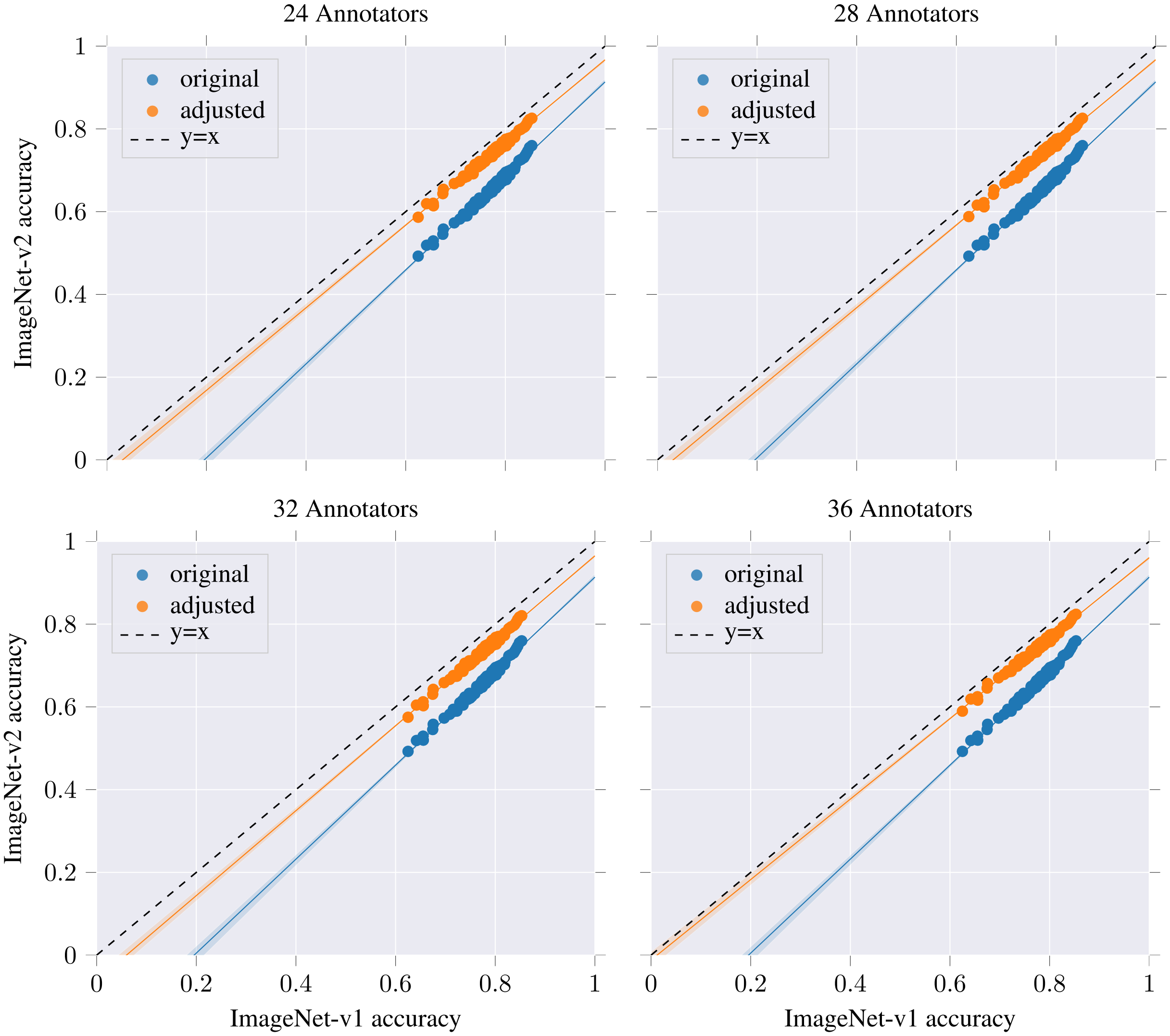


Figure 13. Replicating the `v1` vs `v2` accuracy plot using different numbers of annotators. We obtain similar results even as we decrease the number of annotators by less than half.

F.3. Varying Model Expressiveness

We plot the results of varying the number of parameters (here, by changing the number of beta distributions in our mixture) in Figure 14.

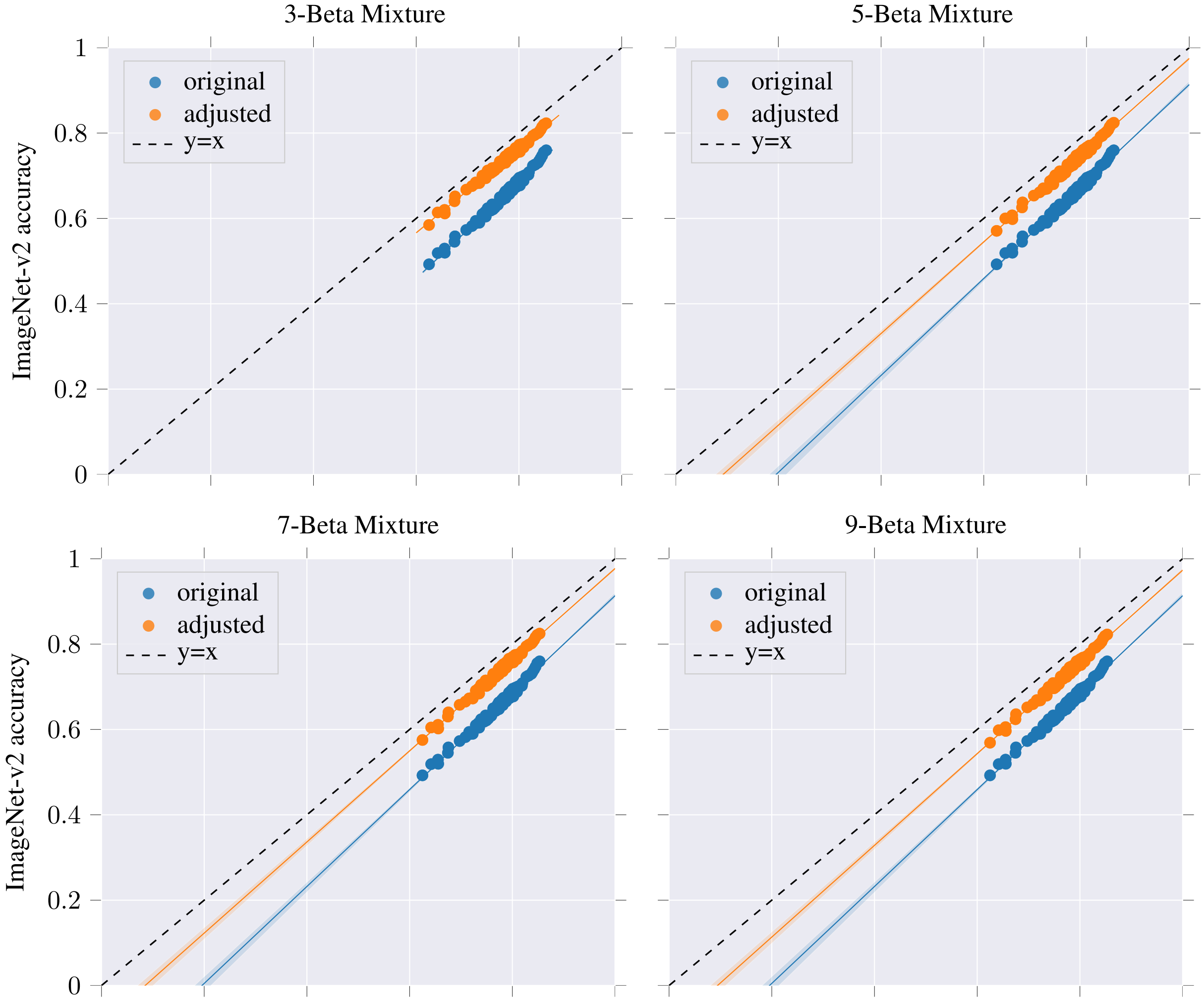


Figure 14. Replicating the `v1` vs `v2` accuracy plot using different numbers of parameters. We obtain similar results even as we increase the number of parameters by more than four-fold.

F.4. EM Algorithm for Mixture Fitting

To fit the parameters of the beta-binomial mixture model we apply the Expectation-Maximization algorithm, optimizing over mixture coefficients $\{\pi_i\}$, as well as parameters of each mixture element $\{(\alpha_i, \beta_i)\}$. Our application of the EM algorithm is rather canonical—first, we compute membership probabilities p_j^i for each example j with respect to each mixture element i, then minimize the weighted log-likelihood with respect to the mixture probabilities. Pseudocode is given in Algorithm 1.

Algorithm 1 Our instantiation of the EM algorithm

```
  Input: A set of size $n$ of empirical selection probabilities $\{\hat{s}_j\}$, the observed rate at which each image was selected, number of mixture components $k$.
  Start with random guesses for all parameters:
      $\alpha_i, \beta_i, \pi_i \leftarrow \text{random}$
  for each training iteration do
    1. Calculate membership probabilities for each observed element:
        $p_j^i = \frac{\pi_i \cdot p(\hat{s}; \alpha_i, \beta_i, 40)}{\sum_{r=1}^{k} \pi_r \cdot p(\hat{s}; \alpha_r, \beta_r, 40)} \qquad \forall\, j \in [n], i \in [k],$
    where $p(\cdot; \alpha, \beta, 40)$ is likelihood under the beta-binomial distribution with $40$ samples.
    2. As is standard in EM, update the parameters by minimizing the expected log-likelihood, weighted by the membership probabilities—i.e. update
        $\alpha_i, \beta_i = \min_{\alpha,\beta} \sum_{j=1}^{n} p_j^i \cdot \ell(\hat{s}_j; \alpha_i, \beta_i, 40),$
    where $\ell(\cdot) = -\log(p(\cdot))$ is the negative log-likelihood, and
        $\pi_i = \frac{\sum_{j=1}^{n} p_j^i}{\sum_{r=1}^{k} \sum_{j=1}^{n} p_j^r}.$
  end for
```

G. Full Model Results

In Appendix Table 2, we detail the set of models we use in our evaluation along with their corresponding Top-1 accuracies on ImageNet-v1 and -v2 validation sets. We use open-source pre-trained implementations from `https://github.com/rwightman/pytorch-image-models` for all architectures.

Model	v1	v2
tf_mobilenetv3_small_minimal_100	63.070%	48.270%
dla46_c	64.950%	51.330%
tf_mobilenetv3_small_075	65.490%	50.800%
dla46x_c	66.130%	52.200%
tf_mobilenetv3_small_100	67.500%	53.960%
dla60x_c	68.170%	55.660%
resnet18	70.420%	56.850%
gluon_resnet18_v1b	71.280%	57.610%
seresnet18	71.840%	58.200%
tf_mobilenetv3_large_minimal_100	71.910%	57.870%
hrnet_w18_small	72.860%	58.120%
tv_resnet34	73.080%	60.060%
spnasnet_100	73.760%	61.040%
tf_mobilenetv3_large_075	73.850%	59.430%
gluon_resnet34_v1b	74.470%	61.630%
mnasnet_100	74.620%	61.020%
densenet121	74.650%	61.810%
dla34	74.680%	61.510%
seresnet34	74.820%	62.330%
resnet34	74.990%	62.240%
hrnet_w18_small_v2	75.000%	61.540%
fbnetc_100	75.080%	61.240%
resnet26	75.270%	62.730%
semnasnet_100	75.690%	62.360%
tf_mobilenetv3_large_100	75.710%	61.400%
mobilenetv3_rw	75.740%	61.870%
tv_resnet50	75.820%	62.600%
dpn68	76.020%	63.000%
tf_mixnet_s	76.210%	62.040%
tf_efficientnet_b0	76.240%	63.050%
densenet169	76.370%	63.450%
hrnet_w18	76.500%	64.560%
mixnet_s	76.570%	62.840%
dla60	76.800%	64.610%
efficientnet_b0	76.820%	64.050%
seresnext26_32x4d	76.980%	64.050%
resnet26d	77.020%	63.970%
resnet101	77.090%	65.020%
tf_mixnet_m	77.120%	63.540%
tf_efficientnet_b0_ap	77.130%	64.290%
tf_efficientnet_cc_b0_4e	77.190%	64.110%
tf_efficientnet_es	77.200%	64.360%
inception_v3	77.240%	65.090%
densenet161	77.240%	64.790%
densenet201	77.280%	64.480%
res2net50_48w_2s	77.420%	64.260%
gluon_resnet50_v1b	77.530%	65.130%

adv_inception_v3	77.680%	65.380%
mixnet_m	77.710%	64.090%
gluon_resnet50_v1c	77.710%	65.180%
dpn68b	77.720%	64.830%
tf_efficientnet_cc_b0_8e	77.740%	64.410%
resnet152	77.760%	66.410%
tv_resnext50_32x4d	77.790%	65.130%
dla60_res2next	77.980%	65.820%
hrnet_w30	78.010%	65.860%
seresnet50	78.020%	65.160%
res2net50_26w_4s	78.050%	64.590%
seresnet101	78.070%	66.150%
dla60x	78.160%	66.090%
res2next50	78.180%	65.370%
tf_inception_v3	78.220%	65.480%
dla102	78.290%	65.710%
hrnet_w44	78.300%	67.130%
dla169	78.380%	66.450%
wide_resnet101_2	78.430%	65.460%
wide_resnet50_2	78.430%	65.750%
tf_efficientnet_b1	78.530%	65.620%
res2net50_14w_8s	78.540%	65.180%
hrnet_w32	78.600%	65.620%
dla60_res2net	78.610%	65.550%
tf_mixnet_l	78.610%	65.750%
hrnet_w40	78.670%	66.600%
efficientnet_b1	78.690%	66.300%
tf_efficientnet_em	78.710%	65.560%
resnext50_32x4d	78.790%	66.530%
dla102x	78.810%	66.140%
seresnet152	78.850%	66.540%
hrnet_w48	78.860%	66.320%
gluon_inception_v3	78.880%	66.110%
res2net50_26w_6s	78.890%	66.200%
mixnet_l	78.890%	66.180%
resnet50	79.000%	65.770%
hrnet_w64	79.090%	67.650%
res2net50_26w_8s	79.100%	66.710%
xception	79.110%	66.320%
gluon_resnet101_v1b	79.110%	66.300%
gluon_resnet50_v1s	79.140%	66.220%
gluon_resnet50_v1d	79.200%	66.740%
tf_efficientnet_b1_ap	79.330%	66.290%
tf_efficientnet_cc_b1_8e	79.360%	65.890%
dla102x2	79.400%	67.830%
seresnext50_32x4d	79.420%	66.790%
resnext101_32x8d	79.490%	66.660%
gluon_resnext50_32x4d	79.630%	67.610%
gluon_resnet101_v1c	79.660%	66.870%
resnext50d_32x4d	79.700%	67.700%
tf_efficientnet_b2	79.730%	67.320%
res2net101_26w_4s	79.740%	66.750%
dpn98	79.830%	67.550%
dpn107	79.950%	67.490%

dpn131	80.020%	67.580%
gluon_resnet152_v1b	80.030%	67.610%
gluon_xception65	80.070%	68.000%
ens_adv_inception_resnet_v2	80.080%	68.630%
efficientnet_b2	80.080%	67.490%
gluon_seresnext50_32x4d	80.100%	67.800%
inception_v4	80.170%	68.490%
gluon_resnet152_v1c	80.230%	67.660%
gluon_resnet101_v1s	80.320%	68.020%
mixnet_xl	80.340%	68.000%
tf_efficientnet_el	80.550%	67.190%
seresnext101_32x4d	80.570%	69.050%
gluon_resnext101_32x4d	80.580%	67.510%
dpn92	80.600%	66.750%
tf_efficientnet_b2_ap	80.680%	67.380%
inception_resnet_v2	80.840%	68.750%
gluon_resnext101_64x4d	80.860%	69.050%
gluon_resnet152_v1d	80.870%	68.730%
gluon_resnet101_v1d	81.020%	67.960%
gluon_seresnext101_32x4d	81.060%	68.890%
gluon_resnet152_v1s	81.470%	68.980%
gluon_seresnext101_64x4d	81.700%	69.040%
tf_efficientnet_b3	81.810%	69.360%
gluon_senet154	81.900%	69.930%
senet154	82.100%	70.020%
tf_efficientnet_b3_ap	82.130%	69.920%
nasnetalarge	82.780%	71.660%
pnasnet5large	83.210%	71.970%
tf_efficientnet_b4	83.350%	71.920%
tf_efficientnet_b5	84.030%	72.200%
tf_efficientnet_b4_ap	84.210%	72.130%
tf_efficientnet_b5_ap	84.260%	73.520%
tf_efficientnet_b6	84.510%	72.940%
tf_efficientnet_b6_ap	85.000%	74.570%
tf_efficientnet_b7_ap	85.460%	75.110%

Table 2: Models used in our analysis with the corresponding Top-1 on the ImageNet v1 and v2 validation sets.