UNCERTAINTY SETS FOR IMAGE CLASSIFIERS USING CONFORMAL PREDICTION

Anastasios N. Angelopoulos∗†, Stephen Bates∗, Jitendra Malik, & Michael I. Jordan
Department of Electrical Engineering and Computer Sciences
Department of Statistics
University of California, Berkeley
Berkeley, CA 94704, USA
{angelopoulos,stephenbates,malik,jordan}@cs.berkeley.edu

ABSTRACT

Convolutional image classifiers can achieve high predictive accuracy, but quantifying their uncertainty remains an unresolved challenge, hindering their deployment in consequential settings. Existing uncertainty quantification techniques, such as Platt scaling, attempt to calibrate the network’s probability estimates, but they do not have formal guarantees. We present an algorithm that modifies any classifier to output a predictive set containing the true label with a user-specified probability, such as 90%. The algorithm is simple and fast like Platt scaling, but provides a formal finite-sample coverage guarantee for every model and dataset. Our method modifies an existing conformal prediction algorithm to give more stable predictive sets by regularizing the small scores of unlikely classes after Platt scaling. In experiments on both Imagenet and Imagenet-V2 with ResNet-152 and other classifiers, our scheme outperforms existing approaches, achieving coverage with sets that are often factors of 5 to 10 smaller than a stand-alone Platt scaling baseline.

1 INTRODUCTION

Imagine you are a doctor making a high-stakes medical decision based on diagnostic information from a computer vision classifier. What would you want the classifier to output in order to make the best decision? This is not a casual hypothetical; such classifiers are already used in medical settings (e.g., Razzak et al., 2018; Lundervold & Lundervold, 2019; Li et al., 2014). A maximum-likelihood diagnosis with an accompanying probability may not be the most essential piece of information. To ensure the health of the patient, you must also rule in or rule out harmful diagnoses. In other words, even if the most likely diagnosis is a stomach ache, it is equally or more important to rule out stomach cancer. Therefore, you would want the classifier to give you—in addition to an estimate of the most likely outcome—actionable uncertainty quantification, such as a set of predictions that provably covers the true diagnosis with a high probability (e.g., 90%). This is called a prediction set (see Figure 1). Our paper describes a method for constructing prediction sets from any pre-trained image classifier that are formally guaranteed to contain the true class with the desired predictor, relatively small, and practical to implement. Our method modifies a conformal predictor (Vovk et al., 2005) given in Romano et al. (2020) for the purpose of modern image classification in order to make it more stable in the presence of noisy small probability estimates. Just as importantly, we provide extensive evaluations and code for conformal prediction in computer vision.

Formally, for a discrete response $Y \in \mathcal{Y} = \{1, \ldots, K\}$ and a feature vector $X \in \mathbb{R}^d$, we desire an uncertainty set function, $C(X)$, mapping a feature vector to a subset of $\{1, \ldots, K\}$ such that

$$P(Y \in C(X)) \geq 1 - \alpha,$$

(1)

for a pre-specified confidence level $\alpha$ such as 10%. Conformal predictors like our method can modify any black-box classifier to output predictive sets that are rigorously guaranteed to satisfy the
desired coverage property shown in Eq. (1). For evaluations, we focus on Imagenet classification using convolutional neural networks (CNNs) as the base classifiers, since this is a particularly challenging testbed. In this setting, X would be the image and Y would be the class label. Note that the guarantee in Eq. (1) is marginal over X and Y—it holds on average, not for a particular image X.

A first approach toward this goal might be to assemble the set by including classes from highest to lowest probability (e.g., after Platt scaling and a softmax function; see Platt et al., 1999; Guo et al., 2017) until their sum just exceeds the threshold $1 - \alpha$. We call this strategy naive and formulate it precisely in Algorithm 1. There are two problems with naive: first, the probabilities output by CNNs are known to be incorrect (Nixon et al., 2019), so the sets from naive do not achieve coverage. Second, image classification models’ tail probabilities are often badly miscalibrated, leading to large sets that do not faithfully articulate the uncertainty of the model; see Section 2.3. Moreover, smaller sets that achieve the same coverage level can be generated with other methods.

The coverage problem can be solved by picking a new threshold using holdout samples. For example, with $\alpha = 10\%$, if choosing sets that contain 93% estimated probability achieves 90% coverage on the holdout set, we use the 93% cutoff instead. We refer to this algorithm, introduced in Romano et al. (2020), as Adaptive Prediction Sets (APS). The APS procedure provides coverage but still produces large sets. To fix this, we introduce a regularization technique that tempers the influence of these noisy estimates, leading to smaller, more stable sets. We describe our proposed algorithm, Regularized Adaptive Prediction Sets (RAPS), in Algorithms 2 and 3 (with APS as a special case). As we will see in Section 2, both APS and RAPS are always guaranteed to satisfy Eq. (1)—regardless of model and dataset. Furthermore, we show that RAPS is guaranteed to have better performance than choosing a fixed-size set. Both methods impose negligible computational requirements in both training and evaluation, and output useful estimates of the model’s uncertainty on a new image given, say, 1000 held-out examples.

In Section 3 we conduct the most extensive evaluation of conformal prediction in deep learning to date on Imagenet and Imagenet-V2. We find that RAPS sets always have smaller average size than naive and APS sets. For example, using a ResNeXt-101, naive does not achieve coverage, while APS and RAPS achieve it almost exactly. However, APS sets have an average size of 19, while RAPS sets have an average size of 2 at $\alpha = 10\%$ (Figure 2 and Table 1). We will provide an accompanying codebase that implements our method as a wrapper for any PyTorch classifier, along with code to exactly reproduce all of our experiments.

1.1 RELATED WORK

Reliably estimating predictive uncertainty for neural networks is an unsolved problem. Historically, the standard approach has been to train a Bayesian neural network to learn a distribution over network weights (Quinonero-Candela et al., 2005; MacKay, 1992; Neal, 2012; Kuleshov et al., 2018; Gal, 2016). This approach requires computational and algorithmic modifications; other approaches avoid these via ensembles (Lakshminarayanan et al., 2017; Jiang et al., 2018) or approximations of Bayesian inference (Riquelme et al., 2018; Sensoy et al., 2018). These methods also have major practical limitations; for example, ensembling requires training many copies of a neural network adversarially. Therefore, the most widely used strategy is ad-hoc traditional calibration of the softmax scores with Platt scaling (Platt et al., 1999; Guo et al., 2017; Nixon et al., 2019).
This work develops a method for uncertainty quantification based on conformal prediction. Originating in the online learning literature, conformal prediction is an approach for generating predictive sets that satisfy the coverage property in Eq. (1) (Vovk et al., 1999; 2005). We use a convenient data-splitting version known as split conformal prediction that enables conformal prediction methods to be deployed for essentially any predictor (Papadopoulos et al., 2002; Lei et al., 2018). While mechanically very different from traditional calibration as discussed above, we will refer to our approach as conformal calibration to highlight that the two methodologies have overlapping but different goals.

Conformal prediction is a general framework, not a specific algorithm—important design decisions must be made to achieve the best performance for each context. To this end, Romano et al. (2020) and Cauchois et al. (2020) introduce techniques aimed at achieving coverage that is similar across regions of feature space, whereas Vovk et al. (2003); Hechtlinger et al. (2018) and Guan & Tibshirani (2019) introduce techniques aimed at achieving equal coverage for each class. While these methods have conceptual appeal, thus far there has been limited empirical evaluation of this general approach for state-of-the-art CNNs. Concretely, the only works that we are aware of that include some evaluation of conformal methods on ImageNet—the gold standard for benchmarking computer vision methods—are Hechtlinger et al. (2018), Park et al. (2019), Cauchois et al. (2020), and Messoudi et al. (2020), although in all four cases further experiments are needed to more fully evaluate their operating characteristics for practical deployment. At the heart of conformal prediction is the conformal score: a measure of similarity between labeled examples which is used to compare a new point to among those in a hold out set. Our theoretical contribution can be summarized as a modification of the conformal score from Romano et al. (2020) to have smaller, more stable sets. Lastly, there are alternative approaches to returning prediction sets not based on conformal prediction (Pearce et al., 2018; Zhang et al., 2018). These methods can be used as input to a conformal procedure to potentially improve performance, but they do not have finite-sample coverage guarantees when used alone.

**Algorithm 1 Naive Prediction Sets**

**Input:** $\alpha$, sorted scores $s$, associated permutation of classes $I$, boolean $\text{rand}$

1: procedure $\text{NAIVE}(\alpha, s, I, \text{rand})$
2: $L \leftarrow 1$
3: while $\sum_{i=1}^{L} s_i < 1 - \alpha$ do
4:     $L \leftarrow L + 1$
5:     if $\text{rand}$ then
6:         $U \leftarrow \text{Unif}(0, 1)$
7:         $V \leftarrow (\sum_{i=1}^{L} s_i - (1 - \alpha))/s_L$
8:     if $U \leq V$ then
9:         $L \leftarrow L - 1$
10: return $\{I_1, \ldots, I_L\}$

**Output:** The $1 - \alpha$ prediction set, $\{I_1, \ldots, I_L\}$
In developing uncertainty set methods to improve upon naive, we are guided by three desiderata. First and most importantly, the **coverage desideratum** says the sets must provide $1 - \alpha$ coverage, as discussed above. Secondly, the **size desideratum** says we want sets of small size, since these convey more detailed information and may be more useful in practice. Lastly, the **adaptiveness desideratum** says we want the sets to communicate instance-wise uncertainty: they should be smaller for easy test-time examples than for hard ones; see Figure 1 for an illustration. Coverage and size are obviously competing objectives, but size and adaptiveness are also often in tension. The size desideratum seeks small sets, while the adaptiveness desideratum seeks larger sets when the classifier is uncertain. For example, always predicting a set of size five could achieve coverage, but it is not adaptive. As noted above, both APS and RAPS achieve correct coverage, and we will show that RAPS improves upon APS according to the other two desiderata.

We now turn to the specifics of our proposed method. We begin in Subsection 2.1 by describing an abstract data-splitting procedure called conformal calibration that enables the near-automatic construction of valid predictive sets (that is, sets satisfying Eq. (1)). Subsequently, in Subsection 2.2, we provide a detailed presentation of our procedure, with commentary in Section 2.3. In Subsection 2.4 we discuss the optimality of our procedure, proving that it is at least as good as the procedure that returns sets of a fixed size, unlike alternative approaches.

### 2.1 Conformal Calibration

We first review a general technique for producing valid prediction sets, following the articulation in Gupta et al. (2019). Consider a procedure that outputs a predictive set for each observation, and further suppose that this procedure has a tuning parameter $\tau$ that controls the size of the sets. (In RAPS, $\tau$ is the cumulative sum of the sorted, penalized classifier scores.) We take a small independent conformal calibration set of data, and then choose the tuning parameter $\tau$ such that the predictive sets are large enough to achieve $1 - \alpha$ coverage on this set. See Figure 3 for an illustration. This calibration step yields a choice of $\tau$, and the resulting set is formally guaranteed to have coverage $1 - \alpha$ on a future test point from the same distribution; see Theorem 1 below.

Formally, let $(X_i, Y_i)_{i=1,...,n}$ be an independent and identically distributed (i.i.d.) set of variables that was not used for model training. Further, let $C(x, u, \tau) : \mathbb{R}^d \times [0,1] \times \mathbb{R} \to 2^Y$ be a set-valued function that takes a feature vector $x$ to a subset of the possible labels. The second argument $u$ is included to allow for randomized procedures; let $U_1, \ldots, U_n$ be i.i.d. uniform $[0,1]$ random variables that will serve as the second argument for each data point. Suppose that the sets are indexed by $\tau$ such that they are nested, meaning larger values of $\tau$ lead to larger sets:

$$C(x, u, \tau_1) \subseteq C(x, u, \tau_2) \quad \text{if} \quad \tau_1 \leq \tau_2. \quad (2)$$

To find a function that will achieve $1 - \alpha$ coverage on test data, we select the smallest $\tau$ that gives at least $1 - \alpha$ coverage on the conformal calibration set, with a slight correction to account for the finite sample size:

$$\hat{\tau}_{ccal} = \inf\left\{ \tau : \frac{|\{i : Y_i \in C(X_i, U_i, \tau)\}|}{n} \geq \frac{(n + 1)(1 - \alpha)}{n} \right\}. \quad (3)$$

2 Methods

Figure 3: **Visualizations of conformal calibration and RAPS sets.** In the left panel, the y-axis shows the empirical coverage on the conformal calibration set, and $1 - \alpha' = \frac{1}{(n + 1)(1 - \alpha)}$. In the right panel, the printed numbers indicate the cumulative probability plus penalty mass. For the indicated value $\hat{\tau}_{ccal}$, the RAPS prediction set is $\{c, d, f, b\}$. 
The set function $C(x, u, \tau)$ with this data-driven choice of $\tau$ is guaranteed to have correct finite-sample coverage on a fresh test observation, as stated formally next.

**Theorem 1** (Conformal calibration coverage guarantee). Suppose $(X_i, Y_i, U_i)_{i=1,\ldots,n}$ and $(X_{n+1}, Y_{n+1}, U_{n+1})$ are i.i.d. and let $C(x, u, \tau)$ be a set-valued function satisfying the nesting property in Eq. (2). Suppose further that the sets $C(x, u, \tau)$ grow to include all labels for large enough $\tau$: for all $x \in \mathbb{R}^d$, $C(x, u, \tau) = \mathcal{Y}$ for some $\tau$. Then for $\hat{\tau}_{\text{ical}}$ defined as in Eq. (3), we have the following coverage guarantee:

$$P\left(Y_{n+1} \in C(X_{n+1}, U_{n+1}, \hat{\tau}_{\text{ical}})\right) \geq 1 - \alpha.$$ 

This is the same coverage property as Eq. (1) in the introduction, written in a more explicit manner. The result is not new—a special case of this result leveraging sample-splitting first appears in the regression setting in Papadopoulos et al. (2002), and the core idea of conformal prediction was introduced even earlier; see Vovk et al. (2005).

As a technical remark, the theorem also holds if the observations to satisfy the weaker condition of exchangeability; see Vovk et al. (2005). In addition, for most families of set-valued functions $C(x, u, \tau)$ there is a matching upper bound:

$$P\left(Y_{n+1} \in C(X_{n+1}, U_{n+1}, \hat{\tau}_{\text{ical}})\right) \leq 1 - \alpha + \frac{1}{n + 1}.$$ 

Roughly speaking, this will hold whenever the sets grow smoothly in $\tau$. See Lei et al. (2018) for a formal statement of the required conditions.

### 2.2 Our Method

Conformal calibration is a powerful general idea, allowing one to achieve the coverage desideratum for any choice of sets $C(x, u, \tau)$. Nonetheless, this is not yet a full solution, since the quality of the resulting prediction sets can vary dramatically depending on the design of $C(x, u, \tau)$. In particular, we recall the size and adaptiveness desiderata from Section 1—we want our uncertainty sets to be as small as possible while faithfully articulating the instance-wise uncertainty of each test point. In this section, we explicitly give our algorithm, which can be viewed as a special case of conformal calibration with the uncertainty sets $\hat{C}$ designed to extract information from CNNs.

Our algorithm has three main ingredients. First, for a feature vector $x$, the base model computes class probabilities $\hat{\pi}_x \in \mathbb{R}^k$, and we order the classes from most probable to least probable. Then, we add a regularization term to promote small predictive sets. Finally, we conformally calibrate the penalized prediction sets to guarantee coverage on future test points.

Formally, let $\rho_x(y) = \sum_{y' = 1}^K \hat{\pi}_x(y') I_{\{\hat{\pi}_x(y') > \hat{\pi}_x(y)\}}$ be the total probability mass of the set of labels that are more likely than $y$. These are all the labels that will be included before $y$ is included. In addition, let $o_x(y) = |\{y' \in \mathcal{Y}: \hat{\pi}_x(y') \geq \hat{\pi}_x(y)\}|$ be the ranking of $y$ among the labels based on the probabilities $\hat{\pi}$. For example, if $y$ is the third most likely label, then $o_x(y) = 3$.$^1$ We take

$$C^*(x, u, \tau) := \left\{ y : \rho_x(y) + \hat{\pi}_x(y) \cdot u + \lambda \cdot \left(o_x(y) - k_{\text{reg}}\right)^+ \leq \tau \right\} , \tag{4}$$

where $(z)^+$ denotes the positive part of $z$ and $\lambda, k_{\text{reg}} \geq 0$ are regularization hyperparameters that are introduced to encourage small set sizes. See Figure 3 for a visualization of a RAPS predictive set and Appendix E for a discussion of how to select $k_{\text{reg}}$ and $\lambda$.

Since this is the heart of our proposal, we carefully parse each term. First, the $\rho_x(y)$ term increases as $y$ ranges from the most probable to least probable label, so our sets will prefer to include the $y$ that are predicted to be the most probable. The second term, $\hat{\pi}_x(y) \cdot u$, is a randomized term to handle the fact that the value will jump discretely with the inclusion of each new $y$. The randomization term can never impact more than one value of $y$: there is at most one value of $y$ such that $y \in C(x, 0, \tau)$ but $y \notin C(x, 1, \tau)$. These first two terms can be viewed as the CDF transform after arranging the classes from most likely to least likely, randomized in the usual way to result in a continuous uniform random variable (cf. Romano et al., 2020). We discuss randomization further in Appendix B.

$^1$For ease of notation, we assume distinct probabilities. Else, label-ordering ties should be broken randomly.
Algorithm 2 RAPS Conformal Calibration

Input: $\alpha, s, I \in \{0, 1\}^{n \times K}, I \in \{1, \ldots, K\}^{n \times K}$, and $y \in \{0, 1, \ldots, K\}^n$ corresponding respectively to the sorted scores, the associated permutation of indexes, and ground-truth labels for each of $n$ examples in the calibration set; $k_{\text{reg}}, \lambda$; boolean $\text{rand}$

1: procedure $\text{RAPSC}(\alpha, s, I, y, \lambda)$
2: for $i \in \{1, \ldots, n\}$ do
3: $L_i \leftarrow j$ such that $I_{i,j} = y_i$
4: $E_i \leftarrow \Sigma_{j=1}^{L_i} s_{i,j} + \lambda(L_i - k_{\text{reg}})^+$
5: if $\text{rand}$ then
6: $U \sim \text{Unif}(0, 1)$
7: $E_i \leftarrow E_i - U \cdot s_{i,L_i}$
8: $\hat{\tau}_{\text{ccal}} \leftarrow \text{the } [(1 - \alpha)/(1 + n)] \text{ largest value in } \{E_i\}_{i=1}^n$
9: return $\hat{\tau}_{\text{ccal}}$

Output: The generalized quantile, $\hat{\tau}_{\text{ccal}}$ \hspace{3cm} $\triangleright$ The value in Eq. (3)

Algorithm 3 RAPS Prediction Sets

Input: $\alpha$, sorted scores $s$ and the associated permutation of classes $I$ for a test-time example, $\hat{\tau}_{\text{ccal}}$ from Algorithm 2, $k_{\text{reg}}, \lambda$, boolean $\text{rand}$

1: procedure $\text{RAPS}(\alpha, s, I, \hat{\tau}_{\text{ccal}}, k_{\text{reg}}, \lambda, \text{rand})$
2: $L \leftarrow \{j \in Y : \Sigma_{i=1}^{j} s_i + \lambda(j - k_{\text{reg}})^+ \leq \hat{\tau}_{\text{ccal}} \}$ \hspace{1cm} $\triangleright$ The $L$ most likely classes
3: if $\text{rand}$ then
4: $U \sim \text{Unif}(0, 1)$
5: $L \leftarrow L - 1 \{\Sigma_{i=1}^{L} s_i + \lambda(L - k_{\text{reg}})^+ - \hat{\tau}_{\text{ccal}})/(s_L + \lambda L > k_{\text{reg}}) \leq U\}$
6: return $C = \{I_1, \ldots, I_L\}$

Lastly, the regularization promotes small set sizes: for values of $y$ that occur farther down the ordered list of classes, the term $\lambda \cdot (\alpha \cdot (y - k_{\text{reg}})^+$ makes that value of $y$ require a higher value of $\tau$ before it is included in the predictive set. For example, if $k_{\text{reg}} = 5$, then the sixth most likely value of $y$ has an extra penalty of size $\lambda$, so it will never be included until $\tau$ exceeds $\rho_x(y) + \tilde{\pi}_x(y) \cdot u + \lambda$, whereas it enters when $\tau$ exceeds $\rho_x(y) + \tilde{\pi}_x(y) \cdot u$ in the nonregularized version. Our method has the following coverage property:

Proposition 1 (RAPS coverage guarantee). Suppose $(X_i, Y_i, U_i)_{i=1,\ldots,n}$ and $(X_{n+1}, Y_{n+1}, U_{n+1})$ are i.i.d. and let $C^*(x, u, \tau)$ be defined as in Eq. (4). Suppose further that $\tilde{\pi}_x(y) > 0$ for all $x$ and $y$. Then for $\tau_{\text{ccal}}$ defined as in Eq. (3), we have the following coverage guarantee:

$$1 - \alpha \leq P\left(Y_{n+1} \in C^*(X_{n+1}, U_{n+1}, \hat{\tau}_{\text{ccal}}) \right) \leq 1 - \alpha + \frac{1}{n + 1}.$$ 

Note that the first inequality is a corollary of Theorem 1, and the second inequality is a special case of the remark in Section 2.1. The restriction that $\tilde{\pi}_x(y) > 0$ is not necessary for the first inequality.

2.3 Why Regularize?

In our experiments, the sets from APS are larger than necessary, because APS is sensitive to the noisy probability estimates far down the list of classes. This noise leads to a permutation problem of unlikely classes, where ordering of the classes with small probability estimates is determined mostly by random chance. If 5% of the true classes from the calibration set are deep in the tail due to the permutation problem, APS will choose large 95% predictive sets; see Figure 2. The inclusion of the RAPS regularization causes the algorithm to avoid using the unreliable probabilities in the tail; see Figure 4. We discuss how RAPS improves the adaptiveness of APS in Section 4 and Appendix E.

2.4 Optimality Considerations

To complement these experimental results, we now formally prove that RAPS with the correct regularization parameters will always dominate the simple procedure that returns a fixed set size. (Section 3.5 shows the parameters are easy to select and RAPS is not sensitive to their values). For a feature vector $x$, let $\hat{y}_{(j)}(x)$ be the label with the $j$th highest predicted probability. We define the top-$k$ predictive sets to be $\{\hat{y}_{(1)}(x), \ldots, \hat{y}_{(k)}(x)\}$.
### Table 1: Results on Imagenet-Val

We report coverage and size of the optimal, randomized fixed sets, naive, APS, and RAPS sets for nine different Imagenet classifiers. The median-of-means for each column is reported over 100 different trials. See Section 3.1 for full details.

Proposition 2 (RAPS dominates top-k sets). Suppose \((X_i, Y_i, U_i)_{i=1,...,n}\) and \((X_{n+1}, Y_{n+1}, U_{n+1})\) are i.i.d. draws. Let \(k^*\) be the smallest \(k\) such that the top-\(k\) predictive sets have coverage at least \(\left\lfloor (n+1)(1-\alpha) \right\rfloor/n\) on the conformal calibration points \((X_i, Y_i)_{i=1,...,n}\). Take \(C^*(x, u, \tau)\) as in Eq. (4) with any \(k_{reg} \leq k^*\) and \(\lambda = 1\). Then with \(\tau_{cal}\) chosen as in Eq. (3), we have

\[
C^*(X_{n+1}, U_{n+1}, \hat{\tau}_{cal}) \subseteq \{\hat{y}(1)(x), \ldots, \hat{y}(k^*)(x)\}.
\]

In words, the RAPS procedure with heavy regularization will be at least as good as the top-\(k\) procedure in the sense that it has smaller or same average set size while maintaining the desired coverage level. This is not true of either the naive baseline or the APS procedure; Table 2 shows that these two procedures usually return predictive sets with size much larger than \(k^*\).

### 3 Experiments

In this section we report on experiments that study the performance of the predictive sets from naive, APS, and RAPS, evaluating each based on the three desiderata above. We begin with a brief preview of the experiments. In **Experiment 1**, we evaluate naive, APS, and RAPS on Imagenet-Val. Both APS and RAPS provided almost exact coverage, while naive sets had coverage slightly below the specified level. APS has larger sets on average than naive and RAPS. RAPS has a much smaller average set size than APS and naive. In **Experiment 2**, we repeat Experiment 1 on Imagenet-V2, and the conclusions still hold. In **Experiment 3**, we produce histograms of set sizes for naive, APS, and RAPS for several different values of \(\lambda\), illustrating a simple tradeoff between set size and adaptiveness. In **Experiment 4**, we compute histograms of RAPS sets stratified by image difficulty, showing that RAPS sets are smaller for easier images than for difficult ones. In **Experiment 5**, we report the performance of RAPS with many values of the tuning parameters.

In our experiments, we use nine standard, pretrained Imagenet classifiers from the torchvision repository (Paszke et al., 2019) with standard normalization, resize, and crop parameters. Before applying naive, APS, or RAPS, we calibrated the classifiers using the standard temperature scaling/Platt scaling procedure as in Guo et al. (2017) on the calibration set. Thereafter, naive, APS, and RAPS were applied, with RAPS using a data-driven choice of parameters described in Appendix E. We use the randomized versions of these algorithms—see Appendix B for a discussion.

#### 3.1 Experiment 1: Coverage vs Set Size on Imagenet

In this experiment, we calculated the coverage and mean set size of each procedure for two different choices of \(\alpha\). Over 100 trials, we randomly sampled two subsets of Imagenet-Val: one conformal calibration subset of size 20K and one evaluation subset of size 20K. The median-of-means over trials for both coverage and set size are reported in Table 1. Figure 2 illustrates the performances of naive, APS, and RAPS; RAPS has much smaller sets than both naive and APS, while achieving coverage. We also report results from a conformalized fixed-k procedure, which finds the smallest fixed set size achieving coverage on the holdout set, \(k^*\), then predicts sets of size \(k^* - 1\) or \(k^*\) on new examples in order to achieve exact coverage; see Algorithm 4 in Appendix E.
<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Coverage</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-1</td>
<td>Top-5</td>
<td>Top K</td>
</tr>
<tr>
<td>ResNeXt101</td>
<td>0.678</td>
<td>0.874</td>
<td>0.900</td>
</tr>
<tr>
<td>ResNet152</td>
<td>0.67</td>
<td>0.876</td>
<td>0.900</td>
</tr>
<tr>
<td>ResNet101</td>
<td>0.657</td>
<td>0.859</td>
<td>0.901</td>
</tr>
<tr>
<td>ResNet50</td>
<td>0.634</td>
<td>0.847</td>
<td>0.901</td>
</tr>
<tr>
<td>ResNet18</td>
<td>0.572</td>
<td>0.802</td>
<td>0.902</td>
</tr>
<tr>
<td>DenseNet161</td>
<td>0.653</td>
<td>0.862</td>
<td>0.902</td>
</tr>
<tr>
<td>VGG16</td>
<td>0.588</td>
<td>0.817</td>
<td>0.900</td>
</tr>
<tr>
<td>Inception</td>
<td>0.573</td>
<td>0.797</td>
<td>0.900</td>
</tr>
<tr>
<td>ShuffleNet</td>
<td>0.559</td>
<td>0.781</td>
<td>0.899</td>
</tr>
</tbody>
</table>

Table 2: Results on Imagenet-V2. We report coverage and size of the optimal, randomized fixed sets, naive, APS, and RAPS sets for nine different Imagenet classifiers. The median-of-means for each column is reported over 100 different trials at the 10% level. See Section 3.2 for full details.

3.2 EXPERIMENT 2: COVERAGE VS SET SIZE ON IMAGENET-V2

The same procedure as Experiment 1 was repeated on Imagenet-V2, with exactly the same normalization, resize, and crop parameters. The size of the calibration and evaluation sets was 5K, since Imagenet-V2 is a smaller dataset. The result shows that our method can still provide coverage even for models trained on different distributions, as long as the conformal calibration set comes from the new distribution. The variance of the coverage is higher due to having less data.

3.3 EXPERIMENT 3: SET SIZES OF NAIVE, APS, AND RAPS ON IMAGENET

We investigate the effect of regularization in more detail. For three values of $\lambda$, we collected the set sizes produced by each of naive, APS, and RAPS and report their histograms in Figure 4.

3.4 EXPERIMENT 4: ADAPTIVENESS OF RAPS ON IMAGENET

We now show that RAPS sets are smaller for easy images than hard ones, addressing the adaptiveness desideratum.

Table 4 reports the size-stratified coverages of RAPS at the 90% level with $k_{reg} = 5$ and different choices of $\lambda$. When $\lambda$ is small, RAPS allows sets to be large. But when $\lambda = 1$, RAPS clips sets to be a maximum of size 5. Table 8 (in the Appendix) stratifies by image difficulty, showing that RAPS sets are small for easy examples and large for hard ones. Experiments 3 and 4 together illustrate the tradeoff between adaptiveness and size: as the average set size decreases, the RAPS procedure truncates sets larger than the smallest fixed set that provides coverage, taming the heavy tail of the APS procedure. Since RAPS with large $\lambda$ undercovers hard examples, it must compensate by taking larger sets for easy examples to ensure the $1 - \alpha$ marginal coverage guarantee. However, the size only increases slightly since easy images are more common than hard ones, and the total probability mass can often exceed $\hat{\tau}_{ccal}$ by including only one more class. If this behavior is not desired, we can instead automatically pick $\lambda$ to optimize the adaptiveness of RAPS; see Section 4.

3.5 EXPERIMENT 5: CHOICE OF TUNING PARAMETERS

While any value of the tuning parameters $\lambda$ and $k_{reg}$ lead to coverage (Proposition 1), some values will lead to smaller sets. In Experiments 1 and 2, we chose $k_{reg}$ and $\lambda$ adaptively from data (see Appendix E), achieving strong results for all models and choices of the coverage level. Table 3 gives the performance of RAPS with many choices of $k_{reg}$ and $\lambda$ for ResNet-152.
In Table 4, we report on the coverage of \( \lambda \) and a ResNet-152, and coverage level 90%.

### 4 ADAPTIVENESS AND CONDITIONAL COVERAGE

In this section, we point to a definition of adaptiveness that is more natural for the image classification setting than the existing notion of conditional coverage. We show that \( \text{APS} \) does not satisfy conditional coverage, and that \( \text{RAPS} \) with small \( \lambda \) outperforms it in terms of adaptiveness.

#### 4.1 A METRIC OF CONDITIONAL COVERAGE

We say that a set-valued predictor \( C : \mathbb{R}^d \to 2^Y \) satisfies exact conditional coverage if \( P(Y \in C(X) \mid X = x) = 1 - \alpha \) for each \( x \). Distribution-free guarantees on conditional coverage are impossible (Vovk, 2012; Lei & Wasserman, 2014), but many algorithms try to satisfy it approximately (Romano et al., 2019; 2020; Cauchois et al., 2020). In a similar spirit, Tibshirani et al. (2019) suggest a notion of local conditional coverage, where one asks for coverage in a neighborhood of each point, weighted according to a chosen kernel. Cauchois et al. (2020) introduce the worst-case slab metric for measuring violations of the conditional coverage property. We present a different way of measuring violations of conditional coverage based on the following result:

**Proposition 3.** Suppose \( P(Y \in C(X) \mid X = x) = 1 - \alpha \) for each \( x \in \mathbb{R}^d \). Then,

\[
P(Y \in C(X) \mid \{C(X) \in A\}) = 1 - \alpha \quad \text{for any } A \subset \{0, 1, 2, \ldots\}.
\]

In words, if conditional coverage holds, then coverage holds after stratifying by set size.

In Table 4, we report on the coverage of \( \text{APS} \) and \( \text{RAPS} \), stratified by the size of the prediction set. Turning our attention to the \( \lambda = 0 \) column, we see that when \( \text{APS} \) outputs a set of size 101–1000, \( \text{APS} \) has coverage 97%, substantially higher than 90% nominal rate. By Proposition 3, we conclude that \( \text{APS} \) is not achieving exact conditional coverage, because the scores are far from the oracle probabilities. The \( \text{APS} \) procedure still achieves marginal coverage by overcovering hard examples and undercovering easy ones, an undesirable behavior. Alternatively, \( \text{RAPS} \) can be used to regularize the set sizes—for \( \lambda = .001 \) to \( \lambda = .01 \) the coverage stratified by set size is more balanced. In summary, even purely based on the adaptiveness desideratum, \( \text{RAPS} \) with light regularization is preferable to \( \text{APS} \). Note that as the size of the training data increases, as long as \( \hat{\pi} \) is consistent, naive and \( \text{APS} \) will become more stable, and so we expect less regularization will be needed.

We now describe a particular manifestation of our adaptiveness criterion that we will use to optimize \( \lambda \). Consider disjoint set-size strata \( \{S_i\}_{i=1}^{\infty} \), where \( \bigcup_{j=1}^{\infty} S_j = \{1, \ldots, |Y|\} \). Then define the indexes of examples stratified by the prediction set size of each example from algorithm \( C \) as \( J_j = \{i : |C(X_{i},Y_{i},U_{i})| \in S_j\} \). Then we can define the size-stratified coverage violation of an algorithm \( C \) on strata \( S \) as

\[
\text{SSCV}(C, \{S\}_{i=1}^{\infty}) = \sup_{j} \left| \frac{|\{i : Y_i \in C(X_{i},Y_{i},U_{i}), i \in J_j\}|}{|J_j|} - (1 - \alpha) \right|.
\]

Table 3: Set sizes of \( \text{RAPS} \) with parameters \( k_{\text{reg}} \) and \( \lambda \), a ResNet-152, and coverage level 90%.

<table>
<thead>
<tr>
<th>( k_{\text{reg}} )</th>
<th>( \lambda = 0 )</th>
<th>( \lambda = 0.001 )</th>
<th>( \lambda = 0.01 )</th>
<th>( \lambda = 0.1 )</th>
<th>( \lambda = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.97</td>
<td>0.93</td>
<td>0.92</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>2</td>
<td>0.96</td>
<td>0.93</td>
<td>0.92</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>5</td>
<td>0.96</td>
<td>0.93</td>
<td>0.92</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>10</td>
<td>0.96</td>
<td>0.93</td>
<td>0.92</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>50</td>
<td>0.96</td>
<td>0.93</td>
<td>0.92</td>
<td>0.93</td>
<td>0.93</td>
</tr>
</tbody>
</table>

Table 4: Coverage conditional on set size. We report average coverage of images stratified by the size of the set output by \( \text{RAPS} \) using a ResNet-152 for varying \( \lambda \). The marginal coverage rate is 90%.
Table 5: Adaptiveness results after automatically tuning $\lambda$. We report the median size-stratified coverage violations of APS and RAPS over 10 trials. See Section 4.2 for experimental details.

<table>
<thead>
<tr>
<th>Model</th>
<th>APS at $\alpha = 10%$</th>
<th>Violation at $\alpha = 5%$</th>
<th>APS at $\alpha = 5%$</th>
<th>Violation at $\alpha = 5%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNeXt101</td>
<td>0.088</td>
<td>0.048</td>
<td>0.046</td>
<td>0.020</td>
</tr>
<tr>
<td>ResNet152</td>
<td>0.070</td>
<td>0.041</td>
<td>0.039</td>
<td>0.020</td>
</tr>
<tr>
<td>ResNet101</td>
<td>0.072</td>
<td>0.059</td>
<td>0.039</td>
<td>0.015</td>
</tr>
<tr>
<td>ResNet50</td>
<td>0.070</td>
<td>0.030</td>
<td>0.038</td>
<td>0.017</td>
</tr>
<tr>
<td>ResNet18</td>
<td>0.047</td>
<td>0.034</td>
<td>0.032</td>
<td>0.023</td>
</tr>
<tr>
<td>DenseNet161</td>
<td>0.076</td>
<td>0.040</td>
<td>0.039</td>
<td>0.017</td>
</tr>
<tr>
<td>VGG16</td>
<td>0.051</td>
<td>0.024</td>
<td>0.030</td>
<td>0.015</td>
</tr>
<tr>
<td>Inception</td>
<td>0.084</td>
<td>0.042</td>
<td>0.043</td>
<td>0.025</td>
</tr>
<tr>
<td>ShuffleNet</td>
<td>0.059</td>
<td>0.034</td>
<td>0.035</td>
<td>0.020</td>
</tr>
</tbody>
</table>

4.2 Optimizing Adaptiveness with RAPS

Next, we show empirically that RAPS with an automatically chosen set of $k_{reg}$ and $\lambda$ improves the adaptiveness of APS, as measured by the SSCV criterion. After picking $k_{reg}$ as described in Appendix E, we can choose $\lambda$ using the same tuning data to optimize this notion of adaptiveness.

To maximize adaptiveness, we’d like to choose $\lambda$ to minimize the size-stratified coverage violation of RAPS. Write $C_{\lambda}$ to mean the RAPS procedure for a fixed choice of $k_{reg}$ and $\lambda$. Then we would like to pick

$$\lambda = \arg\min_{\lambda'} \text{SSCV}(C_{\lambda'}, \{S\}_{j=s}^{z_j}).$$  

6

In our experiments, we choose a relatively coarse partitioning of the possible set sizes: 0-1, 2-3, 4-10, 11-100, and 101-1000. Then, we chose the $\lambda \in \{0.00001, 0.0001, 0.0008, 0.001, 0.0015, 0.002\}$ which minimized the size-stratified coverage violation on the tuning set. The results in Table 5 show RAPS always outperforms the adaptiveness of APS on the test set, even with this coarse, automated choice of parameters. The table reports the median size-stratified coverage violation over 10 independent trials of APS and RAPS with automated parameter tuning.

4.3 Alternatives to Conditional Coverage

Lastly, we argue that conditional coverage is a poor notion of adaptiveness when the best possible model (i.e., one fit on infinite data) has high accuracy. Given such a model, the oracle procedure from Romano et al. (2020) would return the correct label with probability $1 - \alpha$ and the empty set with probability $\alpha$. That is, having correct conditional coverage for high-signal problems where $Y$ is perfectly determined by $X$ requires a perfect classifier. In our experiments on ImageNet, APS does not approximate this behavior. Therefore, conditional coverage isn’t the right goal for prediction sets with realistic sample sizes. Proposition 3 and the SSCV criterion suggests a relaxation; we could require that we have the right coverage, no matter the size of the prediction set: $P(Y \in C(x) \mid \{|C(x)| \in A\}) \geq 1 - \alpha$ for any $A \subset \{0, 1, 2, \ldots\}$. We view this as a promising way to reason about adaptiveness in high-signal problems such as image classification.

5 Discussion

For classification tasks with many possible labels, our method enables a researcher to take any base classifier and return predictive sets guaranteed to achieve a pre-specified error level, such as 90%, while retaining small average size. It is simple to deploy, so it is an attractive, automatic way to quantify the uncertainty of image classifiers—an essential task in such settings as medical diagnostics, self-driving vehicles, and flagging dangerous internet content. Predictive sets in computer vision (from RAPS and other conformal methods) have many further uses, since they systematically identify hard test-time examples. Finding such examples is useful in active learning where one only
has resources to label a small number of points. In a different direction, one can improve efficiency of a classifier by using a cheap classifier outputting a prediction set first, and an expensive one only when the cheap classifier outputs a large set (a cascade; see, e.g., Li et al. 2015), and Fisch et al. (2021) for an implementation of conformal prediction in this setting. One can also use predictive sets during model development to identify failure cases and outliers and suggest strategies for improving its performance. Prediction sets are most useful for problems with many classes; returning to our initial medical motivation, we envision RAPS could be used by a doctor to automatically screen for a large number of diseases (e.g. via a blood sample) and refer the patient to relevant specialists.

ACKNOWLEDGEMENTS

A. A. was partially supported by the National Science Foundation Graduate Research Fellowship Program and a Berkeley Fellowship. We wish to thank John Cherian, Sasha Sax, Amit Kohli, Tete Xiao, Todd Chapman, Mariel Werner, Ilija Radosavovic, Shiry Ginosar, Tia Miceli, and Tiago Miguel Saldanha Salvador for giving feedback on an early version of this manuscript. We used Hydra (Yadan, 2019) and PyTorch (Paszke et al., 2019) in our development. This material is based upon work supported in part by the Army Research Office under contract/grant number W911NF-16-1-0368.

REFERENCES


A Proofs

Theorem 1. Let $s(x, u, y) = \inf \{y \in C(x, u, \tau)\}$, and let $s_i = s(X_i, U_i, Y_i)$ for $i = 1, \ldots, n$. Then
$$\{y : s(x, u, y) \leq \tau\} = \{y : y \in C(x, u, \tau)\}$$
because $C(x, u, \tau)$ is a finite set growing in $\tau$ by the assumption in Eq. (2). Thus,
$$\{\tau : |\{i : s_i \leq \tau\}| \geq \left\lceil (1 - \alpha)(n + 1) \right\rceil\} = \left\{\tau : \frac{|\{i : Y_i \in C(X_i, U_i, \tau)\}|}{n} \geq \frac{\left\lceil (n + 1)(1 - \alpha) \right\rceil}{n}\right\}.$$ Considering the left expression, the infimum over $\tau$ of the set on the left hand side is the
$$\left\lceil (1 - \alpha)(n + 1) \right\rceil$$ smallest value of the $s_i$, so this is the value of $\hat{\tau}_{\text{cal}}$. Since $s_1, \ldots, s_n, s(X_{n+1}, U_{n+1}, Y_{n+1})$ are exchangeable random variables, $|\{i : s(X_{n+1}, U_{n+1}, Y_{n+1}) > s_i\}|$ is stochastically dominated by the discrete uniform distribution on $\{0, 1, \ldots, n\}$. We thus have that
$$P(Y_{n+1} \notin C(X_{n+1}, U_{n+1}, \hat{\tau}_{\text{cal}})) = P(s(X_{n+1}, U_{n+1}, Y_{n+1}) > \hat{\tau}_{\text{cal}})$$
$$= P(|\{i : s(X_{n+1}, U_{n+1}, Y_{n+1}) > s_i\}| \geq \left\lceil (n + 1)(1 - \alpha) \right\rceil)$$
$$= P\left(\frac{|\{i : s(X_{n+1}, U_{n+1}, Y_{n+1}) > s_i\}|}{n + 1} \geq \frac{(n + 1)(1 - \alpha)}{n + 1}\right) \leq \alpha.$$

Proposition 1. The lower bound follows from Theorem 1. To prove the upper bound, using the result from Theorem 2.2 of Lei et al. (2018) it suffices to show that the variables $s(X_i, U_i, Y_i) = \inf \{\tau : Y_i \in C(X_i, U_i, \tau)\}$ are almost surely distinct. To this end, note that that
$$s(X_i, U_i, Y_i) = \rho_{X_i}(Y_i) + \hat{\pi}_{X_i}(Y_i) \cdot U_i + \lambda(\sigma_{X_i}(Y_i) - k_{\text{reg}})^+, $$
and due to the middle term of the sum, these values are distinct almost surely provided $\hat{\pi}_{X_i}(Y_i) > 0$.

Proposition 2. We first show that $\hat{\tau}_{\text{cal}} \leq 1 + k^* - k_{\text{reg}}$. Note that since at least $\left\lceil (1 - \alpha)(n + 1) \right\rceil$ of the conformal calibration points are covered by a set of size $k^*$, at least $\left\lceil (1 - \alpha)(n + 1) \right\rceil$ of the $E_i$ in Algorithm 2 are less than or equal to $1 + k^* - k_{\text{reg}}$. Thus, by the definition of $\hat{\tau}_{\text{cal}}$, we have
### Table 6: Results on Imagenet-Val.

We report coverage and size of the optimal, randomized fixed sets, naive, APS, and RAPS sets for nine different Imagenet classifiers. The median-of-means for each column is reported over 100 different trials. See Section 3.1 for full details.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Coverage</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-1</td>
<td>Top-5</td>
<td>Top K</td>
</tr>
<tr>
<td>ResNeXt101</td>
<td>0.794</td>
<td>0.945</td>
<td>0.905</td>
</tr>
<tr>
<td>ResNet152</td>
<td>0.783</td>
<td>0.940</td>
<td>0.950</td>
</tr>
<tr>
<td>ResNet101</td>
<td>0.774</td>
<td>0.936</td>
<td>0.950</td>
</tr>
<tr>
<td>ResNet50</td>
<td>0.762</td>
<td>0.929</td>
<td>0.950</td>
</tr>
<tr>
<td>ResNet18</td>
<td>0.698</td>
<td>0.891</td>
<td>0.950</td>
</tr>
<tr>
<td>DenseNet161</td>
<td>0.772</td>
<td>0.936</td>
<td>0.950</td>
</tr>
<tr>
<td>VGG16</td>
<td>0.716</td>
<td>0.904</td>
<td>0.950</td>
</tr>
<tr>
<td>Inception</td>
<td>0.695</td>
<td>0.887</td>
<td>0.950</td>
</tr>
<tr>
<td>ShuffleNet</td>
<td>0.694</td>
<td>0.883</td>
<td>0.950</td>
</tr>
</tbody>
</table>

B B RANDOMIZED PREDICTORS

The reader may wonder why we choose to use a randomized procedure. The randomization is needed to achieve $1 - \alpha$ coverage exactly, which we will explain via an example. Note that the randomization is of little practical importance, since the predictive set output by the randomized procedure will differ from the that of the non-randomized procedure by at most one element.

Turning to an example, assume for a particular input image we expect a set of size $k$ to have 91% coverage, and a set of size $k - 1$ to have 89% coverage. In order to achieve our desired coverage of 90%, we randomly choose size $k$ or $k - 1$ with equal probability. In general, the probabilities will not be equal, but rather chosen so the weighted average of the two coverages is exactly 90%. If a user of our method desires deterministic sets, it is easy to turn off this randomization with a single flag, resulting in slightly conservative sets.

C IMAGENET AND IMAGENET V2 RESULTS FOR $\alpha = 5\%$

We repeated Experiments 1 and 2 with $\alpha = 5\%$. See the results in Tables 6 and 7.

D COVERAGE AND SIZE CONDITIONAL ON IMAGE DIFFICULTY

In order to probe the adaptiveness properties of APS and RAPS we stratified coverage and size by image difficulty (the position of the true label in the list of most likely to least likely classes, based...
Table 7: Results on Imagenet-V2. We report coverage and size of the optimal, randomized fixed sets, naive, APS, and RAPS sets for nine different Imagenet classifiers. The median-of-means for each column is reported over 100 different trials at the 5% level. See Section 3.2 for full details.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Coverage</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-1</td>
<td>Top-5</td>
<td>Top K</td>
</tr>
<tr>
<td>ResNet101</td>
<td>0.678</td>
<td>0.950</td>
<td>0.937</td>
</tr>
<tr>
<td>ResNet152</td>
<td>0.670</td>
<td>0.951</td>
<td>0.944</td>
</tr>
<tr>
<td>ResNet101</td>
<td>0.656</td>
<td>0.949</td>
<td>0.944</td>
</tr>
<tr>
<td>ResNet50</td>
<td>0.634</td>
<td>0.949</td>
<td>0.944</td>
</tr>
<tr>
<td>ResNet18</td>
<td>0.572</td>
<td>0.949</td>
<td>0.943</td>
</tr>
<tr>
<td>DenseNet161</td>
<td>0.653</td>
<td>0.941</td>
<td>0.943</td>
</tr>
<tr>
<td>VGG16</td>
<td>0.588</td>
<td>0.934</td>
<td>0.935</td>
</tr>
<tr>
<td>Inception</td>
<td>0.573</td>
<td>0.934</td>
<td>0.935</td>
</tr>
<tr>
<td>ShuffleNet</td>
<td>0.560</td>
<td>0.934</td>
<td>0.935</td>
</tr>
</tbody>
</table>

Table 8: Coverage and size conditional on difficulty. We report coverage and size of RAPS sets using ResNet-152 with $k_{reg} = 5$ and varying $\lambda$ (recall that $\lambda = 0$ is the APS procedure). The desired coverage level is 90%. The ‘difficulty’ is the ranking of the true class’s estimated probability.

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Count</th>
<th>$\lambda = 0$</th>
<th>$\lambda = 0.001$</th>
<th>$\lambda = 0.01$</th>
<th>$\lambda = 0.1$</th>
<th>$\lambda = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15668</td>
<td>0.95</td>
<td>0.95</td>
<td>0.96</td>
<td>0.97</td>
<td>0.98</td>
</tr>
<tr>
<td>2 to 3</td>
<td>2578</td>
<td>0.78</td>
<td>15.7</td>
<td>0.78</td>
<td>10.5</td>
<td>0.84</td>
</tr>
<tr>
<td>4 to 6</td>
<td>717</td>
<td>0.68</td>
<td>31.7</td>
<td>0.68</td>
<td>19.7</td>
<td>0.71</td>
</tr>
<tr>
<td>7 to 10</td>
<td>334</td>
<td>0.63</td>
<td>41.0</td>
<td>0.63</td>
<td>24.9</td>
<td>0.71</td>
</tr>
<tr>
<td>11 to 100</td>
<td>622</td>
<td>0.55</td>
<td>57.8</td>
<td>0.51</td>
<td>34.1</td>
<td>0.22</td>
</tr>
<tr>
<td>101 to 1000</td>
<td>81</td>
<td>0.23</td>
<td>96.7</td>
<td>0.00</td>
<td>51.6</td>
<td>0.00</td>
</tr>
</tbody>
</table>

E Choosing $k_{reg}$ and $\lambda$ to Optimize Set Size

This section describes two procedures for picking $k_{reg}$ and $\lambda$ that optimize for set size or adaptiveness, outperforming APS in both cases.

Algorithm 4 Adaptive Fixed-K

Input: $\alpha; I \in \{1, \ldots, K\}^n \times K$, and $y \in \{0, 1, \ldots, K\}$ corresponding respectively to the classes from highest to lowest estimated probability mass, and labels for each of $n$ examples in the dataset

1: procedure GET-KSTAR$(\alpha, I, y)$
2: for $i \in \{1, \ldots, n\}$ do
3: $L_i \leftarrow j$ such that $L_i, j = y_i$
4: $\hat{k}^* \leftarrow$ the $\lceil (1 - \alpha)(1 + n) \rceil$ largest value in $\{L_i\}_{i=1}^n$
5: return $\hat{k}^*$

Output: The estimate of the smallest fixed size set that achieves coverage, $\hat{k}^*$

To produce Tables 1, 6, 2, and 7, we chose $k_{reg}$ and $\lambda$ adaptively. This required an extra data splitting step, where a small amount of tuning data $\{x_i, y_i\}_{i=1}^m$ were used to estimate $\hat{k}^*$, and then $k_{reg}$ is set to $\hat{k}^*$. Taking $m \approx 1000$ was sufficient, since the algorithm is fairly insensitive to $k_{reg}$ (see Table 3). Then, $\hat{k}^*$ was calculated with Algorithm 4. We produced the Imagenet V2 tables with $m = 1000$ and the Imagenet tables with $m = 10000$.

After choosing $\hat{k}^*$, we chose $\lambda$ to have small set size. We used the same tuning data to pick $\hat{k}^*$ and $\lambda$ for simplicity (this does not invalidate our coverage guarantee since conformal calibration still uses...
fresh data). A coarse grid search on $\lambda$ sufficed, since small parameter variations have little impact on RAPS. For example, we chose the $\lambda \in \{0.001, 0.01, 0.1, 0.2, 0.5\}$ that achieved the smallest size on the $m$ holdout samples in order to produce Tables 1, 6, 2, and 7. We include a subroutine that automatically chooses $\hat{k}^*$ and $\lambda$ to optimize size in our GitHub codebase.

## F Comparison with Least Ambiguous Set-Valued Classifiers

In this section, we compare RAPS to the Least Ambiguous Set-valued Classifier (LAC) method introduced in Sadinle et al. (2019), an alternative conformal procedure that is designed to have small sets. The LAC method provably gives the smallest possible average set size in the case where the input probabilities are correct, with the idea that these sets should be small even when the estimated probabilities are only approximately correct. In the notation of this paper, the LAC method considers nested sets of the following form:

$$C_{LAC}(x, \tau) := \{y : \hat{\pi}_x(y) \geq 1 - \tau\},$$

which can be calibrated using as before in using $\hat{\tau}_{cal}$ from Eq. (3).

We first compare naive, APS, RAPS, and LAC in terms of power and coverage in Table 9. In this experiment, we tuned RAPS to have small set size as described in Appendix E. We see that LAC also achieves correct coverage, as expected since it is a conformal method and satisfies the guarantee from Theorem 1. We further see that it has systematically smaller sets than RAPS, although the difference is slight compared to the gap between APS and RAPS or APS and LAC.

We next compare RAPS to LAC in terms of adaptiveness, tuning RAPS as in Section 4.2. First, in Table 10, we report on the coverage of LAC for images of different difficulties, and see that LAC has dramatically worse coverage for hard images than for easy ones. Comparing this to RAPS in Table 8, we see that RAPS also has worse coverage for more difficult images, although the gap is much smaller for RAPS. Next, in Table 11, we report on the SSCV metric for adaptiveness (and conditional coverage) for APS, RAPS, and LAC. We find that APS and RAPS have much better adaptiveness than LAC, with RAPS being the overall winner. The results of all of these comparisons are expected: LAC is not targeting adaptiveness and instead trying to achieve the smallest possible set size. It succeeds at its goal, sacrificing adaptiveness to do so.
<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Coverage</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-1</td>
<td>Top-5</td>
<td>Top K Naive APS RAPS LAC Top K Naive APS RAPS LAC</td>
</tr>
<tr>
<td>ResNeXt101</td>
<td>0.793</td>
<td>0.945</td>
<td>0.900 0.889 0.900 0.900 0.900 0.900 2.42 17.2 19.9 2.01 1.65</td>
</tr>
<tr>
<td>ResNet152</td>
<td>0.783</td>
<td>0.941</td>
<td>0.900 0.894 0.900 0.900 0.900 0.900 2.64 9.68 10.4 2.09 1.76</td>
</tr>
<tr>
<td>ResNet101</td>
<td>0.774</td>
<td>0.936</td>
<td>0.900 0.895 0.900 0.900 0.900 0.900 2.83 10.0 10.8 2.25 1.87</td>
</tr>
<tr>
<td>ResNet50</td>
<td>0.761</td>
<td>0.929</td>
<td>0.899 0.896 0.900 0.900 0.900 0.900 3.13 11.7 12.3 2.55 2.05</td>
</tr>
<tr>
<td>ResNet18</td>
<td>0.698</td>
<td>0.891</td>
<td>0.900 0.895 0.900 0.900 0.900 0.900 5.74 15.3 16.1 4.38 3.64</td>
</tr>
<tr>
<td>DenseNet161</td>
<td>0.771</td>
<td>0.936</td>
<td>0.900 0.894 0.900 0.900 0.900 0.900 2.84 11.2 12.0 2.29 1.90</td>
</tr>
<tr>
<td>VGG16</td>
<td>0.716</td>
<td>0.904</td>
<td>0.900 0.896 0.901 0.900 0.900 0.900 4.75 13.4 14.1 3.54 3.00</td>
</tr>
<tr>
<td>Inception</td>
<td>0.695</td>
<td>0.886</td>
<td>0.899 0.884 0.900 0.899 0.900 0.900 6.27 74.8 88.8 5.24 4.06</td>
</tr>
<tr>
<td>ShuffleNet</td>
<td>0.694</td>
<td>0.883</td>
<td>0.900 0.892 0.900 0.899 0.900 0.900 6.45 28.8 32.1 5.01 4.14</td>
</tr>
</tbody>
</table>

Table 9: **Results on Imagenet-Val.** We report coverage and size of the optimal, randomized fixed sets, naive, APS, RAPS, and the LAC sets for nine different Imagenet classifiers. The median-of-means for each column is reported over 100 different trials at the 10% level. See Section 3.1 for full details.
<table>
<thead>
<tr>
<th>difficulty</th>
<th>count</th>
<th>cvg</th>
<th>sz</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15668</td>
<td>1.00</td>
<td>1.5</td>
</tr>
<tr>
<td>2 to 3</td>
<td>2578</td>
<td>0.81</td>
<td>2.6</td>
</tr>
<tr>
<td>4 to 6</td>
<td>717</td>
<td>0.23</td>
<td>3.0</td>
</tr>
<tr>
<td>7 to 10</td>
<td>334</td>
<td>0.00</td>
<td>2.9</td>
</tr>
<tr>
<td>11 to 100</td>
<td>622</td>
<td>0.00</td>
<td>2.7</td>
</tr>
<tr>
<td>101 to 1000</td>
<td>81</td>
<td>0.00</td>
<td>2.4</td>
</tr>
</tbody>
</table>

Table 10: **Coverage and size conditional on difficulty.** We report coverage and size of the LAC sets for ResNet-152.

<table>
<thead>
<tr>
<th>Model</th>
<th>Violation at $\alpha = 10%$</th>
<th>Violation at $\alpha = 5%$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>APS</td>
<td>RAPS</td>
</tr>
<tr>
<td>ResNeXt101</td>
<td>0.086</td>
<td><strong>0.047</strong></td>
</tr>
<tr>
<td>ResNet152</td>
<td>0.067</td>
<td><strong>0.039</strong></td>
</tr>
<tr>
<td>ResNet101</td>
<td>0.075</td>
<td><strong>0.060</strong></td>
</tr>
<tr>
<td>ResNet50</td>
<td>0.071</td>
<td><strong>0.042</strong></td>
</tr>
<tr>
<td>ResNet18</td>
<td>0.050</td>
<td><strong>0.024</strong></td>
</tr>
<tr>
<td>DenseNet161</td>
<td>0.076</td>
<td><strong>0.055</strong></td>
</tr>
<tr>
<td>VGG16</td>
<td>0.051</td>
<td><strong>0.023</strong></td>
</tr>
<tr>
<td>Inception</td>
<td>0.086</td>
<td><strong>0.042</strong></td>
</tr>
<tr>
<td>ShuffleNet</td>
<td>0.058</td>
<td><strong>0.030</strong></td>
</tr>
</tbody>
</table>

Table 11: **Adaptiveness results after automatically tuning $\lambda$.** We report the median SSCV of APS, RAPS and LAC over 10 trials. See Appendix 4.2 for experimental details.