Test-time augmentation improves efficiency in conformal prediction

Anonymous Authors[1]

Abstract

The goal in conformal classification is to output a *set* of predicted classes, accompanied by a probabilistic guarantee that the set includes the true class. The utility of a conformal predictor depends upon its ability to achieve a strong guarantee without generating an excessively large set. In practice, the utility of conformal prediction has often been limited by a tendency to yield large prediction sets. We study this phenomenon and provide insights into why large set sizes persist, even for conformal methods designed to produce small sets. Using these insights, we propose a method to reduce prediction set size while maintaining coverage. We use test-time augmentation–a technique that introduces inductive biases during inference–to replace a classifier's predicted probabilities with probabilities aggregated over a set of augmentations. Our approach is flexible, computationally efficient, and effective. It can be combined with any conformal score, requires no model retraining, and reduces prediction set sizes by up to 30%. We conduct an evaluation of the approach spanning three datasets, three models, two established conformal scoring methods, and multiple coverage values to show when and why test-time augmentation is a useful addition to the conformal pipeline.

1. Introduction

Machine learning classifiers excel at providing the most likely category for a particular input; where they often fall short is in providing accurate notions of *uncertainty* (Guo et al., 2017). Conformal prediction has emerged as a promising framework to provide existing classifiers with statistically valid uncertainty estimates. It does this by replacing the prediction of the most likely class with an *uncertainty set*–a set of classes accompanied by a probabilistic guarantee that the true class appears in the set (Vladmir Vovk, 2005). These properties have led to the application of conformal prediction in multiple high-stakes domains, including healthcare (Papadopoulos et al., 2009; Lu et al., 2022a) and finance (Wisniewski et al., 2020).

Unfortunately, achieving a suitably strong guarantee often leads to prediction sets that are uninformatively large (Babbar et al., 2022). For example, nearly every class in the iNaturalist 2021 dataset (Van Horn et al., 2021) has an average prediction set size of more than 100 species at a coverage of 99%–even when using an algorithm designed to yield small sets (Angelopoulos et al., 2022).

To build a conformal classifier one starts with a model that, given an example, outputs a probability for each possible class, and a desired *coverage* (the probability that the set returned by the conformal predictor contains the correct class). One then uses a calibration set of samples to derive a *conformal threshold*, used to generate prediction sets that contain the correct class at the pre-specified coverage level. Overly large predictions sets can be generated when the underlying classifier's prediction for the true class is low. This leads to the inclusion of many classes to meet the coverage guarantee. In this work, we show that 1) introducing inductive biases during inference, in the form of test-time augmentation, can increase the predicted probability of the true class, and 2) doing so leads to smaller prediction sets.

Test-time augmentation generates an ensemble of predictions by perturbing the input with label-preserving transformations. In this work, we learn a test-time augmentation policy of label-preserving transformations using a small set of labeled data that is *distinct* from the labeled examples used to identify the conformal threshold. In doing so, we preserve the assumption of exchangeability, and thereby the coverage guarantee. We demonstrate that the proposed approach reduces set sizes for the classes with the *largest* prediction set sizes by up to 30% with no loss of coverage. We also show that test-time augmentation can bridge gaps between classifiers of different sizes (e.g. test-time augmentation combined with ResNet-50 produces smaller set sizes than ResNet-101 alone).

Contributions The main contributions of this work are threefold. 1) It is the first work to propose combining test-time augmentation and conformal prediction. 2) We present a method that reduces the prediction set sizes of existing conformal predictors by using automatically learned test-time augmentations. 3) We demonstrate, in an extensive set of experiments, that our approach to combining conformal prediction and test-time augmentation leads to dramatically smaller prediction sets.

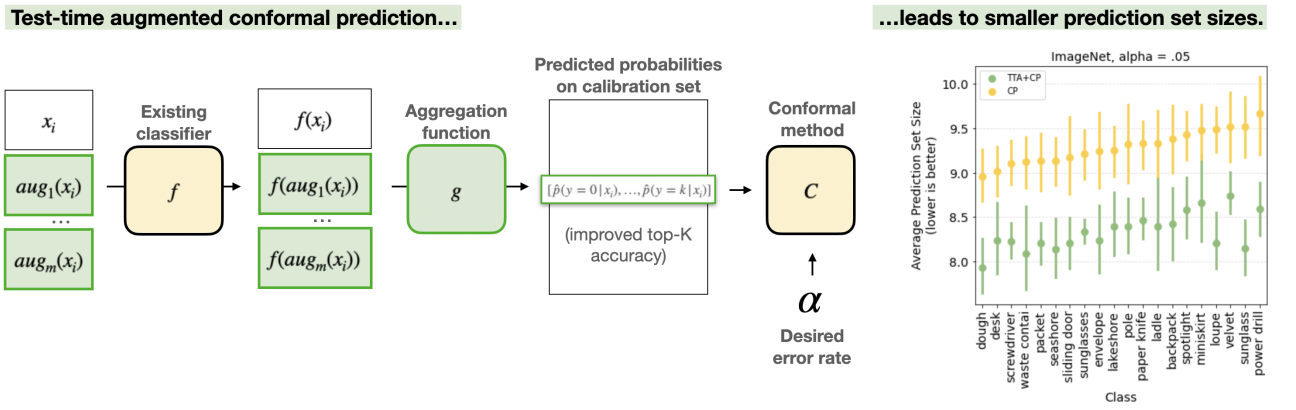

Figure 1: We illustrate the addition of test-time augmentation to conformal calibration in green (left) and provide a snapshot of the improvements it can confer (right). We show results on Imagenet, with a desired coverage of 95%, for the 20 classes with the largest predicted set sizes on average (computed over 10 calibration/test splits).

2. Related work

In recent years, conformal prediction has become a popular approach to uncertainty quantification in machine learning (Barber et al., 2023). It was first introduced by Gammerman et al. (1998), and further developed by Saunders & Holloway (1999) and Vladmir Vovk (2005). We review efforts to ensemble conformal predictors and efforts to reduce prediction set sizes below.

Ensembles in conformal prediction Several methods that generate ensembles of conformal predictors are known to improve efficiency. These methods include cross-conformal prediction (Vovk, 2012), bootstrap conformal prediction (Vovk, 2015), aggregated conformal prediction (Carlsson et al., 2014; Linusson et al., 2017), and out-of-bag conformal prediction (Linusson et al., 2020). The approaches primarily differ in how data is sampled to create the training dataset for the classifier and the calibration set for the conformal predictor. The estimated thresholds are typically averaged over the estimated conformal predictors. However, all require training multiple base classifiers or conformal predictors. Our approach is distinct: we propose a technique to generate an ensemble from a *single* model by perturbing the input, which requires no additional base models and no additional conformal predictors.

Efficiency in conformal prediction There are two ways to improve efficiency in split conformal prediction: adjustments to the conformal score or improvements to the underlying model. Many works have proposed new procedures to estimate and apply thresholds on conformal scores (Tibshirani et al., 2019; Bellotti, 2021; Angelopoulos et al., 2022; Prinster et al., 2022; Ding et al., 2023). Romano et al. (2020) proposed APS, a conformal score based on the cumulative probability required to include the correct class in a prediction set. Angelopoulos et al. (2022) built on this work to propose RAPS, which modifies APS by penalizing the inclusion of low-probability classes. Comparatively little work has focused on improvements to the underlying model. Jensen et al. (2022) ensemble a set of base classifiers, where the classifiers are created by training models on subsets of the training data. Stutz et al. (2022) provide a new way to train the base classifier and conformal wrapper jointly through a conformal training loss. In contrast, our work focuses on improving the underlying model *without* retraining, and can be easily combined with any of the above procedures; indeed, we see that the smallest prediction set sizes are achieved by combining TTA *and* RAPS.

Test-Time Augmentation Test-time augmentation (TTA) is a popular technique to improve the accuracy, robustness, and calibration of an existing classifier by aggregating predictions over a set of input transformations (Shanmugam et al., 2021; Perez et al., 2021; Zhang et al., 2022; Enomoto et al., 2023; Ayhan & Berens, 2018; Conde et al., 2023; Hekler et al., 2023). TTA has been applied to a diverse range of predictive tasks across domains ranging from healthcare (Cohen et al., 2021) to content moderation (Lu et al., 2022b). Consequently, many have proposed new ways to perform TTA—for example, learning when to apply TTA (Mocerino et al., 2021), which augmentations to use (Kim et al., 2020; Lyzhov et al., 2020; Chun et al., 2022), and how to aggregate the resulting predictions (Shanmugam et al., 2021; Chun et al., 2022; Conde et al., 2023). Existing work typically focuses on test-time augmentation's impact on highest predicted probability. Here, we analyze how test-time augmentation increases the predicted probability assigned to the true class when it appears *outside* the top few classes, and how that change is consequential in conformal prediction.

3. Problem setting

We operate within the split conformal prediction framework. In this setting, a conformal classifier $\mathcal{C}(X_i) \subset \{1, \ldots, K\}$ maps input X_i to a subset of K possible classes and requires three inputs:

- **Calibration set** $D^{(cal)} = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$, containing N labeled examples.
- **Classifier** $\hat{f} : \mathcal{X} \mapsto \Delta^K$, mapping input domain $\mathcal{X}$ to a probability distribution over K classes.
- **Desired upper bound on error rate** $\alpha \in [0, 1]$, where $(1 - \alpha)$ represents the probability the set contains the true class.

We study the introduction of two variables drawn from the test-time augmentation literature:

- **Augmentation policy** $\mathcal{A} = \{a_0, \ldots, a_m\}$, consisting of $m + 1$ augmentation functions, where a_0 is the identity transform. Policy $A(x_i)$ maps image x_i to a set of inputs consisting of the original image and m augmentations of the original image.
- **Aggregation function** $\hat{g}$, which aggregates a set of predictions to produce one prediction.

Each variable translates to a critical choice in test-time augmentation: what augmentations to apply ($\mathcal{A}$) and how to aggregate the resulting probabilities ($\hat{g}$).

4. Approach

Preliminaries Our goal is to learn – given an augmentation policy $\mathcal{A}$ – an aggregation function $\hat{g}$ to maximize the accuracy of the underlying classifier, and ultimately reduce the sizes of the prediction sets generated from the classifier's predicted probabilities. We will briefly outline the conformal approach, and then detail the mechanics of our method (illustrated in Figure 1).

Conformal predictors accept three inputs: a probabilistic classifier f, a calibration set $\mathcal{D}^{(cal)}$, and a pre-specified error rate α. Using the these inputs, one can construct a conformal predictor in three steps:

1. Define a score function $c(x, y)$, which produces a *conformal score* representing the uncertainty of the input example and label pair.
2. Produce a distribution of conformal scores across the calibration set by computing $c(x_i, y_i)$ for all $(x_i, y_i) \in \mathcal{D}^{(cal)}$.
3. Compute threshold $\hat{q}$ as the $\lceil (n+1)(1-\alpha) \rceil / n$ quantile of the distribution of conformal scores over n examples in the calibration set.

For a new example x, we compute $c(x, y)$ for all $y \in \{1, \ldots, K\}$, and include all y_j for which $c(x, y_j) < \hat{q}$. We adopt the conformal score proposed by Romano et al. (2020), which equates to the cumulative probability required to include the correct class:

$$\hat{\pi}_x(y') = \hat{p}(y = y'|x) = f(x)_{y'} \tag{1}$$

$$\rho_x(y) = \sum_{y'=1}^{K} \hat{\pi}_x(y') \mathbb{I}[\hat{\pi}_x(y') > \hat{\pi}_x(y)] \tag{2}$$

$$c(x, y) = \rho_x(y) + \hat{\pi}_x(y) \tag{3}$$

where $\rho_x(y)$ is the cumulative probability of all classes predicted with higher probability than y and $\hat{\pi}_x(y')$ corresponds to the predicted probability of class y' given x. Conformal score $c(x, y)$ is thus composed of this cumulative probability and the predicted probability of class y.

Proposal Our approach differs from prior work in that the conformal score is derived by transforming the probabilities output by f using test-time augmentation. Concretely, this replaces Equation 1 with the following, parametrized by augmentation policy $\mathcal{A}$ and augmentation weights θ.

$$\hat{\pi}_x(y') = \hat{p}(y = y'|x) = f_{tta}(x_i; f, \mathcal{A}, \Theta) \tag{4}$$

A key idea is to learn the aggregation weights θ using a portion of the validation set, $D^{(TTA)}$, distinct from calibration set used to identify the conformal threshold ($D^{(cal)}$). In contrast to traditional approaches, where all labeled data is used to estimate the conformal threshold, we instead reserve a portion to improve the underlying classifier. We learn a set of weights which maximize classification accuracy on $D^{(TTA)}$ by minimizing the cross-entropy loss[1] computed between the predicted probabilities and true labels. More formally, f_{tta} applies θ and $\mathcal{A}$ as follows:

$$f_{tta}(x_i; f, \mathcal{A}, \Theta) \quad = \quad \Theta^T \mathbf{A}(f, \mathcal{A}, x_i) \tag{5}$$

where $\mathbf{A}$ uses f to map input x_i to a $M \times K$ matrix of predicted probabilities where M is the number of augmentation transforms and K is the number of classes. Θ is a $1 \times m$ vector corresponding to augmentation-specific weights. Each row in $\mathbf{A}(f, \mathcal{A}, x_i)$ represents the pre-trained classifier's predicted probabilities on augmentation a_m of x_i or $f(a_m(x_i))$.

[1] We found no significant difference between the use of cross-entropy loss and alternate losses considered in the conformal prediction literature (e.g. focal or conformal training loss). See Table 3 in the Appendix.

TTA-Learned refers to TTA combined with learned augmentation weights, while TTA-Avg refers to a simple average over the augmentations.

We refer to the fraction of the validation set allotted to $D^{(TTA)}$ as β. Figure 8 shows that performance is not sensitive to the choice of β; as a result, all experiments use $\beta = .2$ (see Section A.10 for further discussion). This does reduce the amount of data available to identify the appropriate threshold, but we find that the benefits TTA confers outweigh the cost to threshold estimation. Computational cost scales linearly with the size of $\mathcal{A}$; each additional augmentation translates to a forward pass of the base classifier. One can use the learned weights to save computation by identifying which test-time augmentations to generate.

Preserving exchangeability The validity of conformal prediction depends upon the assumption of exchangeability: that all orderings of examples are equally likely (in effect, meaning that the distribution of examples in the calibration set is indistinguishable from the distribution of unseen examples). The use of distinct examples to learn the test-time augmentation policy preserves this guarantee. If we were to instead use the *same* examples to learn the test-time augmentation policy and the conformal threshold, exchangeability could be broken. For example, if the test-time augmentation policy is overfit to the calibration set, the distribution of conformal scores during calibration will differ from the distribution of scores over unseen examples.

5. Experimental Set-Up

Datasets We show results on the test splits of three datasets: ImageNet (Deng et al., 2009) (50,000 natural images across 1,000 classes), iNaturalist (Van Horn et al., 2021) (100,000 images spanning 10,000 species), and CUB-Birds (Wah et al., 2011) (5,794 images representing 200 categories of birds). Images are distributed evenly over classes in ImageNet and iNaturalist, while CUB-Birds has between 11 and 30 images per class.

Models The default model architecture, across all datasets, is ResNet-50 (He et al., 2016). The accuracies of the base classifiers are 76.1% (ImageNet), 76.4% (iNaturalist), and 80.5% (CUB-Birds). To study the relationship between model complexity and performance, we also provide results using ResNet-101 and ResNet-152 on ImageNet. For ImageNet, we make use of the pretrained models made available by PyTorch (Paszke et al., 2019). For iNaturalist, we use a model made public by Niers, Tom (2021). For CUB-Birds, we train a network by finetuning the final layer of a ResNet-50 model initialized with ImageNet's pretrained weights.

Augmentations We consider two augmentation policies. The first (the *simple* augmentation policy) consists of a random-crop and a horizontal-flip; to produce a random crop, we pad the original image with 4 pixels and take a 256x256 crop of the expanded image (thereby preserving the original image resolution). The simple augmentation policy is widely used because these augmentations are likely to be label-preserving. The second, which we will term the *expanded* augmentation policy, consists of 12 augmentations: increase-sharpness, decrease-sharpness, autocontrast, invert, blur, posterize, shear, translate, color-jitter, random_crop, horizontal-flip, and random-rotation. The supplement contains a description of each augmentation (Sec. A.1). These augmentations are not always label preserving, but, as we show, can improve performance when weights are learned.

Baselines We benchmark results using two conformal scores (translating to different definitions of $c(x, y)$ in Equation 3). The first score is APS (Romano et al., 2020) (described in Eqn. 3), which represents the cumulative probability required to include the correct class, and the second is RAPS (Angelopoulos et al., 2022), which modifies APS by adding a term to penalize large set sizes. For all experiments, we perform randomization of conformal scores during calibration and do not allow sets of size 0. We implement RAPS and APS using code provided by Angelopoulos et al. (2022), and automatically select hyperparameters k_{reg} and λ to minimize set size. We also compare against conformal prediction using a simple average over the test-time augmentations (TTA-Avg). In the supplement, we also compare against non-conformal Top-1 and Top-5 prediction sets.

Evaluation We evaluate results using the three metrics commonly used in the conformal prediction literature: efficiency, coverage, and adaptivity. We quantify efficiency using two both average prediction set size (measured across all examples) and class-conditional prediction set size (measured across all examples in a class). Coverage is the percentage of sets containing the true label. We define adaptivity as the size-stratified coverage violation (SSCV), introduced by Angelopoulos et al. (2022). We first partition examples based upon the size of the prediction set. We create bins for set sizes of $[0, 1], [2, 3], [4, 10]$, and $[101,]$. We then compute the empirical coverage within each bin, and compute adaptivity as the maximum difference between theoretical coverage and empirical coverage across bins. The closer this value is to 0, the better the adaptivity.

For each dataset, we report results across 10 randomly generated splits into validation and test sets. For all experiments (save for the validation set size experiment), the validation set and test set are the same size. We allot 20% of examples from the validation set to $D^{(TTA)}$ (used to learn TTA policy), and allot the remaining examples to the calibration

set. For the experiment studying validation set size, we downsample the validation set. We compute statistical significance using a paired t-test, with a Bonferroni correction (Weisstein, 2004) for multiple hypothesis testing. Code to reproduce all experiments will be made publicly available.

6. Results

We first provide statistics on large prediction sets in Sec. 6.1 and present results on the improvements TTA confers across multiple datasets, coverage values, and augmentation policies. We compare against RAPS in the main text since it outperforms baselines in every comparison, and provide results comparing our method to APS and the Top-K baselines in the supplement (Sec. A.6 and Sec. A.5 respectively). Replicates of each experiments across multiple α and datasets can be found in Section A.8. We then examine the dependence of these results on dataset, base model, and class. We conclude by providing some intuition about why test-time augmentation improves the efficiency of conformal predictors.

6.1. Large prediction set sizes.

For the datasets studied here, conventional conformal predictors often produce large prediction sets which consist of many low-probability classes.

Consider the coverage (vs) prediction set size tradeoff made by RAPS(Angelopoulos et al., 2022) (Table 1), a widely used conformal prediction framework designed to produce small set sizes. For a coverage level of 99%, RAPS produces large prediction sets: 10% of examples receive a set size larger than 100 for Imagenet, 193 for iNaturalist, and 44 for CUB-Birds. Looking at the classes included in the prediction sets across all examples, we can see that a large percentage are associated with predicted probabilities lower than $1/(\#\text{ of classes})$: 47% for ImageNet, 62% for iNaturalist, and 45% for CUB-Birds. Relaxing the coverage to 95%, we can still observe large prediction sets: 10% of examples still receive large set sizes (ImageNet: ≥ 10, iNaturalist: ≥ 14, CUB-Birds: ≥ 6).

The existence of large prediction sets is not a criticism of RAPS; it corresponds to a limitation of the underlying probabilistic classifier. There are two possible remedies: improvements to the conformal score, as many prior works have explored (Tibshirani et al., 2019; Angelopoulos & Bates, 2022; Guan, 2023), or improvements to the underlying classifier. As the next section will illustrate, test-time augmentation is a viable approach to improving the underlying classifier, and thereby the performance of conformal predictors.

6.2. TTA produces consistent and significant reductions in prediction set size.

We begin with results in the context of the expanded augmentation policy.

Learned test-time augmentation policies produce meaningfully significant reductions in prediction set size (RAPS+TTA-Learned in Table 1 and APS+TTA-Learned in Table 6). TTA-Learned reduces prediction set sizes significantly in 16 of the 18 cases, and performs comparably in the remaining 2. Across all cases, the combination of RAPS, TTA-Learned, and the expanded augmentation policy produces the smallest average set sizes.

Comparing learned augmentation weights (TTA-Learned) to a fixed average (TTA-Avg) on the expanded augmentation policy, we find that **TTA-Learned performs comparably or better than TTA-Avg** in all comparisons. When we look at the weights learned for the expanded augmentation policy, we see that several augmentations (blur, decrease sharpness, and shear) are consistently assigned a weight of 0, while certain augmentations are consistently included in learned policies (autocontrast, translate).

While TTA improves both RAPS and APS, it produces improvements larger in magnitude for APS (up to 36% across datasets). This is because TTA, like RAPS, tempers the predicted probabilities. TTA lowers the maximum predicted probability on average, thereby reducing model overconfidence. Consequently, the predicted probability assigned to the remaining classes is higher. This is why the expanded augmentation policy demonstrates such strong performance compared to the simple augmentation policy: it tempers the probabilities to a greater extent.

TTA-Learned preserves coverage across all experiments, since it respects the assumption of exchangeability. In some cases, TTA significantly improves coverage, although the magnitude of this difference is small (exact values can be found in Tables 10 and 11). We next evaluate adaptivity using size stratified coverage violation (SSCV). At low alpha (α = .01, and α = .05), TTA-Learned improves efficiency at no cost to adaptivity. At higher alpha (α = .10), there are three settings in which TTA-Learned produces lower values for SSCV (significant according to a paired t-test).

6.3. Datasets, augmentation policies, and base models

Dependence on dataset TTA consistently improves prediction set sizes on ImageNet and iNaturalist, but not CUB-Birds. This may be because the validation set size for CUB-Birds (2,827 images) is an order of magnitude smaller than the validation sets for ImageNet (25,000 images) and iNaturalist (50,000 images). This is consistent with our finding that effectiveness of TTA is positively correlated with the size of the validation set (Figure A.11).

		Expanded Aug Policy			Simple Aug Policy		
Alpha	Method	ImageNet	iNaturalist	CUB-Birds	ImageNet	iNaturalist	CUB-Birds
0.01	RAPS	37.751 ± 2.334	61.437 ± 6.067	**15.293 ± 2.071**	37.751 ± 2.334	61.437 ± 6.067	**15.293 ± 2.071**
0.01	RAPS+TTA-Avg	35.600 ± 2.200	57.073 ± 5.914	**13.111 ± 2.470**	**31.681 ± 3.057**	**54.169 ± 6.319**	**14.550 ± 1.425**
0.01	RAPS+TTA-Learned	**31.248 ± 2.177**	**53.195 ± 4.884**	**14.045 ± 1.323**	**32.702 ± 2.409**	**51.391 ± 5.211**	**13.803 ± 1.734**
0.05	RAPS	5.637 ± 0.357	**7.991 ± 1.521**	3.624 ± 0.361	5.637 ± 0.357	7.991 ± 1.521	3.624 ± 0.361
0.05	RAPS+TTA-Avg	5.318 ± 0.113	**7.067 ± 0.344**	**3.116 ± 0.210**	**4.908 ± 0.099**	**6.451 ± 0.279**	**3.249 ± 0.307**
0.05	RAPS+TTA-Learned	**4.889 ± 0.168**	**6.682 ± 0.447**	**3.571 ± 0.576**	**5.040 ± 0.176**	**6.788 ± 0.496**	**3.290 ± 0.186**
0.10	RAPS	2.548 ± 0.074	2.914 ± 0.116	2.038 ± 0.153	2.548 ± 0.074	2.914 ± 0.116	2.038 ± 0.153
0.10	RAPS+TTA-Avg	2.470 ± 0.071	2.740 ± 0.026	**1.780 ± 0.139**	**2.327 ± 0.086**	**2.610 ± 0.031**	**1.881 ± 0.118**
0.10	RAPS+TTA-Learned	**2.312 ± 0.054**	**2.625 ± 0.043**	**1.893 ± 0.187**	**2.362 ± 0.065**	2.638 ± 0.026	**1.840 ± 0.106**

Table 1: Results across datasets, augmentation policies, and coverage specifications. Each entry corresponds to the average prediction set size across 10 calibration/test splits. Bolded entries represent performance that is either (a) significantly better compared to the baseline (RAPS), or (b) indistinguishable from the best approach. Table 10 reports achieved coverage. Corresponding results for APS can be found in Table 7.

		Expanded Aug Policy			Simple Aug Policy		
Alpha	Method	ResNet-50	ResNet-101	ResNet-152	ResNet-50	ResNet-101	ResNet-152
0.01	RAPS	37.751 ± 2.334	33.624 ± 1.796	29.560 ± 3.481	37.751 ± 2.334	33.624 ± 1.796	29.560 ± 3.481
0.01	RAPS+TTA-Avg	35.600 ± 2.200	30.220 ± 1.774	27.203 ± 2.526	**31.681 ± 3.057**	**27.206 ± 1.840**	**24.106 ± 2.100**
0.01	RAPS+TTA-Learned	**31.248 ± 2.177**	**25.722 ± 1.713**	**23.615 ± 1.656**	**32.702 ± 2.409**	**26.760 ± 1.974**	**24.765 ± 2.736**
0.05	RAPS	5.637 ± 0.357	4.785 ± 0.102	4.376 ± 0.078	5.637 ± 0.357	4.785 ± 0.102	4.376 ± 0.078
0.05	RAPS+TTA-Avg	5.318 ± 0.113	4.433 ± 0.137	4.163 ± 0.185	**4.908 ± 0.099**	**4.147 ± 0.122**	**3.868 ± 0.126**
0.05	RAPS+TTA-Learned	**4.889 ± 0.168**	**4.200 ± 0.200**	**3.824 ± 0.128**	**5.040 ± 0.176**	**4.194 ± 0.194**	**3.916 ± 0.356**
0.10	RAPS	2.548 ± 0.074	2.267 ± 0.024	2.109 ± 0.027	2.548 ± 0.074	2.267 ± 0.024	2.109 ± 0.027
0.10	RAPS+TTA-Avg	2.470 ± 0.071	2.164 ± 0.031	2.049 ± 0.028	**2.327 ± 0.086**	**2.093 ± 0.035**	**1.996 ± 0.018**
0.10	RAPS+TTA-Learned	**2.312 ± 0.054**	**2.099 ± 0.040**	**1.993 ± 0.026**	**2.362 ± 0.065**	**2.091 ± 0.041**	**1.988 ± 0.020**

Table 2: Results across base classifiers for ImageNet. TTA-Learned can bridge the performance gap between different classifiers (for example, outperforming ResNet-152 alone when combined with ResNet-101), and yields significant reductions in set size regardless of the pretrained classifier used. We report achieved coverage in Table 11.

Dependence on augmentation policy We find that the expanded augmentation policy produces greater reductions in set size than the simple augmentation policy. Although the introduction of many augmentations outside of the base model's train-time augmentation policy can decrease the Top-1 accuracy of a classifier, the conformal scores use the predicted probabilities for *all* classes. So, while the expanded test-time augmentation policy may not result in a significantly more accurate Top-1 classifier, it modifies the predicted probabilities for lower ranked classes. Larger augmentation policies also yield greater reductions in average prediction set size (Figure 7). That said, the simple augmentation policy does have its place; it requires fewer forward passes during inference. In the absence of a learned aggregation function, our results suggest that aggregating using an average can still improve the efficiency of conformal predictors (outperforming the original conformal score in 11 comparisons, matching performance in 3, and underperforming in 4).

Dependence on base model We test the generalizability of our results to other models by rerunning the ImageNet experiments using ResNet-101 (accuracy of 77.4%) and ResNet-152 (accuracy of 78.3%). Unsurprisingly, more accurate models result in smaller prediction set sizes (Table 2). TTA variants of conformal prediction again produce significant improvements in set size while maintaining coverage. We were surprised to find that the combination of TTA with ResNet-101 produces smaller set sizes than the more complex ResNet-152 alone. For example, when α is set to .01, RAPS+TTA-Learned and ResNet-101 produce set sizes that contain, on average, 26.5 classes, while RAPS and ResNet-152 produce an average set size of 29.6.

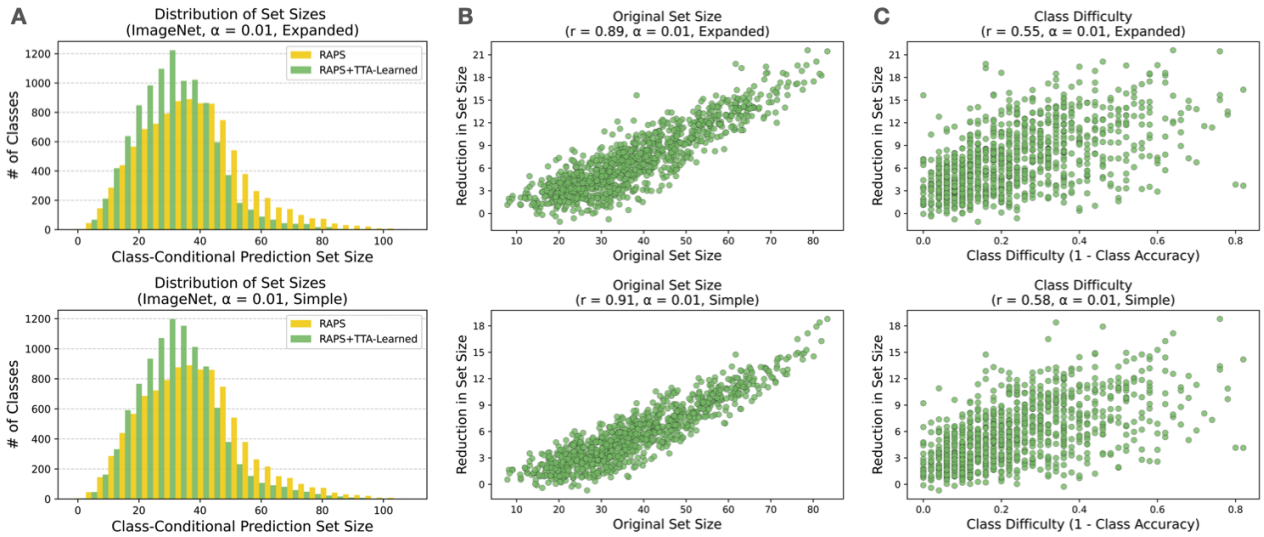

Figure 2: (A) **Class-conditional prediction set sizes.** The histogram describes the distribution of class-conditional prediction set sizes. We plot results for ImageNet with $\alpha = .01$. RAPS+TTA-Learned (green) produces a noticeable reduction in class-conditional prediction set sizes. (B, C) **Relationship between TTA improvements and original class set sizes (B) and class difficulty (C)**. Each point represents the average prediction set size for each class, across 5 splits. The reduction in average set size introduced by TTA (plotted on the y-axis) is positively correlated with original class-conditional set size ((B), expanded: r = 0.89, $p <$ 1e-10) and class difficulty ((C), expanded: r = 0.55, $p <$ 1e-10). In other words, TTA introduces the largest improvements for classes with the largest original prediction set sizes and classes on which the underlying classifier is often incorrect.

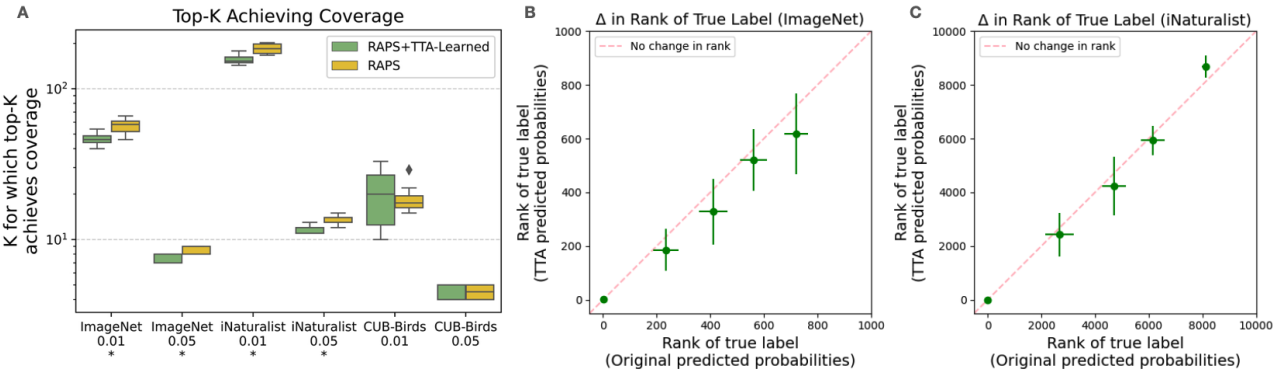

Figure 3: (A) **Effect of TTA-Learned on optimal Top-K**: TTA-Learned significantly lowers the value of k required for Top-k prediction sets to achieve coverage on ImageNet and iNaturalist, but not on CUB-Birds. (B,C) **Effect of TTA-Learned on rank of true class**: TTA-Learned improves the rank of the true class among the sorted predicted probabilities for a given example for both ImageNet (B) and iNaturalist (C). We plot the rank using the original predicted probabilities against the rank using the TTA-transformed probabilities, binning all examples in the validation set into five equal-width bins. The leftmost point in each plot describes a majority of the examples, because the classifier (ResNet-50) assigns the highest predicted probability to the true class on 76.1% of observed examples. Dots that fall below the red line indicate that TTA improves the rank of the true class. Vertical error bars represent spread in TTA-transformed ranks of the true class for examples in a given bin, while horizontal error bars represent spread in the original rank of the true class. We include the corresponding plot for CUB-Birds in the supplement.

6.4. TTA is most effective for classes with the largest prediction sets.

So far, we have established that on average TTA is a useful addition to the conformal pipeline. We now ask where this improvement come from, and what types of classes are responsible. We make two empirical observations. First, classes with larger predicted set sizes benefit most from the introduction of TTA. Figure 2 shows that a class's average prediction set size is significantly correlated with the change in set size TTA-Learned introduces (with the expanded augmentation policy and $\alpha = .01$, $r = 0.89$ and $p < 1e-10$). Second, we find that class difficulty is significantly associated with changes in set size introduced by TTA (with the expanded augmentation policy and $\alpha = .01$, r = 0.55 and $p <$ 1e-10), where prediction sets for difficult classes benefit more from TTA compared to their easier counterparts. These observations are related; harder classes receive larger set sizes, and consequently, offer larger room for improvements in efficiency.

7. Discussion

Why does the addition of test-time augmentation produce smaller prediction set sizes? In short, TTA improves top-K accuracy. We verify this claim by estimating k such that the uncertainty sets comprised of the top k predicted classes for each example achieve a marginal coverage of $(1 - \alpha)$. We see that the probabilities updated by TTA — both with a simple average and learned weights — produce significantly lower values for k compared to the original predicted probabilities for both ImageNet and iNaturalist (Figure 3A). This is *not* true for CUB-Birds, on which TTA offers little to no improvement. One could use such a procedure to determine whether TTA is worth adding to a conformal pipeline without collecting labeled data beyond the validation set.

Another way to understand the impact of TTA is to consider the effect on the *ordering* of classes. It has been observed in the test-time augmentation literature that TTA often promotes the true class from the second-highest to the highest predicted probability, thereby correcting the classification. Here, we introduce a new finding, which explains why TTA is particularly useful to conformal prediction. TTA *also* increases the predicted probability of the true class *even when it is predicted to be unlikely* (for example, promoting the true class from 200th most likely to 100th most likely). We visualize this effect in Figure 3 by plotting the change in true class rank (the index at which the true class appears in the sorted list of predicted probabilities) for all examples in the validation set, stratified into 5 equal-width bins. The lower left point captures examples which are classified correctly; here, test-time augmentation introduces little to no change. In subsequent bins, we see that TTA typically promotes the rank of the true class. We also include the standard deviation across the true class ranks in the original predicted probabilities (x-axis) and the TTA-transformed probabilities (y-axis).

Broader applications of TTA to conformal prediction There are many other ways to combine test-time augmentation and conformal prediction. One might apply test-time augmentation during calibration (when computing conformal scores on the calibration set) and *not* during inference. This leads to smaller set sizes, but unsurprisingly breaks the coverage guarantee. The converse (TTA during inference and not calibration) maintains coverage but dramatically increases the prediction set sizes (because TTA depresses the maximum predicted probability, more classes can be included in the outputted set). Finally, one could consider the value of throwing away conformal prediction (and the guarantees it comes with) altogether, and creating a set out of the predictions made on each of the augmentations; refer to Section A.13 for further discussion.

Limitations Learned test-time augmentation policies require two ingredients: labeled data and multiple forward passes. Although one can minimize costs by parallelizing computation or by using the learned weights to identify which augmentations to generate, inference will always cost more with test-time augmentation. Our results also rely on transformations typically used in image classification. We do not consider other modalities, for which appropriate transformations will substantially differ. Finally, test-time augmentation is one approach to generating ensembles in conformal prediction. Many other more computationally expensive approaches exist. Elucidating the trade-off between computation and ensemble performance is a useful avenue for future work.

8. Conclusion

We present an approach that improves the efficiency of conformal predictors by using test-time augmentation to replace a classifier's predicted probabilities with probabilities aggregated over a set of transformations. Moreover, we show that the learned inductive biases introduced by TTA improve the predicted probability assigned to the true class, even when the true class is predicted to be unlikely. Our experiments show that the approach is effective, efficient, and simple: it reduces prediction set sizes by up to 30%, requires no model re-training, and relies on a portion of labeled data already available to split conformal predictors. The performance of TTA-Learned suggests that, given a labeled dataset, there are settings in which it is wise to use a portion of the labeled data to improve the underlying model is beneficial, instead of reserving all labeled data for the calibration set. In sum, our work takes a step towards practically useful conformal predictors by improving efficiency, without sacrificing adaptivity or coverage.

9. Broader impact

Conformal prediction represents a promising step forward for uncertainty quantification in machine learning. As we grow closer to the deployment of conformal prediction in high-stakes settings, we see two socially-relevant considerations:

- The relationship between uncertainty sets and human models for decision making is not currently well-understood. Human decision-making is known to be biased and incorrect; it will be important to characterize how access to conformal predictions changes this behavior.
- Conformal predictions, as considered in this work, offer a coverage guarantee on *average*, rather than per example. There may still be subsets of the distribution of examples for which prediction sets do not meet the coverage guarantee. Ongoing work towards conformal prediction with conditional guarantees aims to address this problem, but it remains relevant to the deployment of uncertainty sets in safety-critical settings.

References

Angelopoulos, A., Bates, S., Malik, J., and Jordan, M. I. Uncertainty Sets for Image Classifiers using Conformal Prediction, September 2022. URL http://arxiv.org/abs/2009.14193. arXiv:2009.14193 [cs, math, stat].

Angelopoulos, A. N. and Bates, S. A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification, December 2022. URL http://arxiv.org/abs/2107.07511. arXiv:2107.07511 [cs, math, stat].

Ayhan, M. S. and Berens, P. Test-time Data Augmentation for Estimation of Heteroscedastic Aleatoric Uncertainty in Deep Neural Networks. 2018.

Babbar, V., Bhatt, U., and Weller, A. On the Utility of Prediction Sets in Human-AI Teams, May 2022. URL http://arxiv.org/abs/2205.01411. arXiv:2205.01411 [cs].

Barber, R. F., Candes, E. J., Ramdas, A., and Tibshirani, R. J. Conformal prediction beyond exchangeability, March 2023. URL http://arxiv.org/abs/2202.13415. arXiv:2202.13415 [stat].

Bellotti, A. Optimized conformal classification using gradient descent approximation, May 2021. URL http://arxiv.org/abs/2105.11255. arXiv:2105.11255 [cs].

Carlsson, L., Eklund, M., and Norinder, U. Aggregated Conformal Prediction. In Iliadis, L., Maglogiannis, I., Papadopoulos, H., Sioutas, S., and Makris, C. (eds.), *Artificial Intelligence Applications and Innovations*, IFIP Advances in Information and Communication Technology, pp. 231–240, Berlin, Heidelberg, 2014. Springer. ISBN 978-3-662-44722-2. doi: 10.1007/978-3-662-44722-2_25.

Chun, S., Lee, J. Y., and Kim, J. Cyclic test time augmentation with entropy weight method. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 433–442. PMLR, August 2022. URL https://proceedings.mlr.press/v180/chun22a.html. ISSN: 2640-3498.

Cohen, S., Dagan, N., Cohen-Inger, N., Ofer, D., and Rokach, L. ICU Survival Prediction Incorporating Test-Time Augmentation to Improve the Accuracy of Ensemble-Based Models. *IEEE Access*, 9:91584–91592, 2021. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3091622. URL https://ieeexplore.ieee.org/abstract/document/9462159. Conference Name: IEEE Access.

Conde, P., Barros, T., Lopes, R. L., Premebida, C., and Nunes, U. J. Approaching Test Time Augmentation in the Context of Uncertainty Calibration for Deep Neural Networks, April 2023. URL http://arxiv.org/abs/2304.05104. arXiv:2304.05104 [cs].

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. AutoAugment: Learning Augmentation Policies from Data, April 2019. URL http://arxiv.org/abs/1805.09501. arXiv:1805.09501 [cs, stat].

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848. URL https://ieeexplore.ieee.org/abstract/document/5206848. ISSN: 1063-6919.

Ding, T., Angelopoulos, A. N., Bates, S., Jordan, M. I., and Tibshirani, R. J. Class-Conditional Conformal Prediction With Many Classes, June 2023. URL http://arxiv.org/abs/2306.09335. arXiv:2306.09335 [cs, stat].

Einbinder, B.-S., Romano, Y., Sesia, M., and Zhou, Y. Training Uncertainty-Aware Classifiers with Conformalized Deep Learning.

Enomoto, S., Busto, M. R., and Eda, T. Dynamic Test-Time Augmentation via Differentiable Functions, March 2023. URL http://arxiv.org/abs/2212.04681. arXiv:2212.04681 [cs].

Gammerman, A., Vovk, V., and Vapnik, V. Learning by transduction. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, UAI'98, pp. 148–155, San Francisco, CA, USA, July 1998. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-555-8.

Guan, L. Localized conformal prediction: a generalized inference framework for conformal prediction. *Biometrika*, 110(1):33–50, March 2023. ISSN 1464-3510. doi: 10.1093/biomet/asac040. URL https://doi.org/10.1093/biomet/asac040.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On Calibration of Modern Neural Networks, August 2017. URL http://arxiv.org/abs/1706.04599. arXiv:1706.04599 [cs].

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.90. URL http://ieeexplore.ieee.org/document/7780459/.

Hekler, A., Brinker, T. J., and Buettner, F. Test Time Augmentation Meets Post-hoc Calibration: Uncertainty Quantification under Real-World Conditions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(12):14856–14864, June 2023. ISSN 2374-3468. doi: 10.1609/aaai.v37i12.26735. URL https://ojs.aaai.org/index.php/AAAI/article/view/26735. Number: 12.

Jensen, V., Bianchi, F. M., and Anfinsen, S. N. Ensemble Conformalized Quantile Regression for Probabilistic Time Series Forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022. ISSN 2162-2388. doi: 10.1109/TNNLS.2022.3217694. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

Kim, I., Kim, Y., and Kim, S. Learning Loss for Test-Time Augmentation. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4163–4174. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/2ba596643cbbbc20318224181fa46b28-Abstract.html.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

Linusson, H., Norinder, U., Boström, H., Johansson, U., and Löfström, T. On the Calibration of Aggregated Conformal Predictors. In *Proceedings of the Sixth Workshop on Conformal and Probabilistic Prediction and Applications*, pp. 154–173. PMLR, May 2017. URL https://proceedings.mlr.press/v60/linusson17a.html. ISSN: 2640-3498.

Linusson, H., Johansson, U., and Boström, H. Efficient conformal predictor ensembles. *Neurocomputing*, 397:266–278, July 2020. ISSN 0925-2312. doi: 10.1016/j.neucom.2019.07.113. URL https://www.sciencedirect.com/science/article/pii/S0925231219316108.

Lu, C., Lemay, A., Chang, K., Höbel, K., and Kalpathy-Cramer, J. Fair Conformal Predictors for Applications in Medical Imaging. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12008–12016, June 2022a. ISSN 2374-3468. doi: 10.1609/aaai.v36i11.21459. URL https://ojs.aaai.org/index.php/AAAI/article/view/21459. Number: 11.

Lu, H., Shanmugam, D., Suresh, H., and Guttag, J. Improved Text Classification via Test-Time Augmentation, June 2022b. URL http://arxiv.org/abs/2206.13607. arXiv:2206.13607 [cs].

Lyzhov, A., Molchanova, Y., Ashukha, A., Molchanov, D., and Vetrov, D. Greedy Policy Search: A Simple Baseline for Learnable Test-Time Augmentation. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 1308–1317. PMLR, August 2020. URL https://proceedings.mlr.press/v124/lyzhov20a.html. ISSN: 2640-3498.

Mocerino, L., Rizzo, R. G., Peluso, V., Calimera, A., and Macii, E. Adaptive Test-Time Augmentation for Low-Power CPU, May 2021. URL http://arxiv.org/abs/2105.06183. arXiv:2105.06183 [cs, eess].

Niers, Tom. iNaturalist_competition, December 2021. URL https://github.com/EibSReM/iNaturalist_Competition. original-date: 2021-12-10T10:56:46Z.

Papadopoulos, H., Gammerman, A., and Vovk, V. Reliable diagnosis of acute abdominal pain with conformal prediction. 17(2), 2009.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner,

B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Perez, J. C., Alfarra, M., Jeanneret, G., Rueda, L., Thabet, A., Ghanem, B., and Arbelaez, P. Enhancing Adversarial Robustness via Test-time Transformation Ensembling. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 81–91, Montreal, BC, Canada, October 2021. IEEE. ISBN 978-1-66540-191-3. doi: 10.1109/ICCVW54120.2021.00015. URL https://ieeexplore.ieee.org/document/9607771/.

Prinster, D., Liu, A., and Saria, S. JAWS: Auditing Predictive Uncertainty Under Covariate Shift, November 2022. URL http://arxiv.org/abs/2207.10716. arXiv:2207.10716 [cs, stat].

Romano, Y., Sesia, M., and Candès, E. J. Classification with Valid and Adaptive Coverage, June 2020. URL http://arxiv.org/abs/2006.02544. arXiv:2006.02544 [stat].

Saunders, C. and Holloway, R. Transduction with Confidence and Credibility. 1999.

Shanmugam, D., Blalock, D., Balakrishnan, G., and Guttag, J. Better Aggregation in Test-Time Augmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1194–1203, Montreal, QC, Canada, October 2021. IEEE. ISBN 978-1-66542-812-5. doi: 10.1109/ICCV48922.2021.00125. URL https://ieeexplore.ieee.org/document/9710313/.

Stutz, D., Krishnamurthy, Dvijotham, Cemgil, A. T., and Doucet, A. Learning Optimal Conformal Classifiers, May 2022. URL http://arxiv.org/abs/2110.09192. arXiv:2110.09192 [cs, stat].

Tibshirani, R. J., Foygel Barber, R., Candes, E., and Ramdas, A. Conformal Prediction Under Covariate Shift. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/8fb21ee7a2207526da55a679f0332de2-Abstract.html.

Van Horn, G., Cole, E., Beery, S., Wilber, K., Belongie, S., and Mac Aodha, O. Benchmarking Representation Learning for Natural World Image Collections, June 2021. URL http://arxiv.org/abs/2103.16483. arXiv:2103.16483 [cs].

Vladmir Vovk. *Algorithmic Learning in a Random World*. Springer-Verlag, New York, 2005. ISBN 978-0-387-00152-4. doi: 10.1007/b106715. URL http://link.springer.com/10.1007/b106715.

Vovk, V. Cross-conformal predictors, August 2012. URL http://arxiv.org/abs/1208.0806. arXiv:1208.0806 [cs, stat].

Vovk, V. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74 (1-2):9–28, June 2015. ISSN 1012-2443, 1573-7470. doi: 10.1007/s10472-013-9368-4. URL http://link.springer.com/10.1007/s10472-013-9368-4.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Weisstein, E. W. Bonferroni correction. *https://mathworld.wolfram.com/*, 2004.

Wisniewski, W., Lindsay, D., and Lindsay, S. Application of conformal prediction interval estimations to market makers' net positions. In *Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications*, pp. 285–301. PMLR, August 2020. URL https://proceedings.mlr.press/v128/wisniewski20a.html. ISSN: 2640-3498.

Zhang, M., Levine, S., and Finn, C. MEMO: Test Time Robustness via Adaptation and Augmentation, October 2022. URL http://arxiv.org/abs/2110.09506. arXiv:2110.09506 [cs].

A. Appendix

A.1. Augmentations

The simple augmentation policy consists of a random crop and a horizontal flip, drawn from a widely used test-time augmentation policy in image classification (Krizhevsky et al., 2012). The random crop pads the original image by 4 pixels and takes a 256x256 crop of the resulting image. The expanded augmentation consists of 12 augmentations; certain augmentations are stochastic, while others are deterministic. We design this set based on the augmentations included in AutoAugment (Cubuk et al., 2019). We exclude certain augmentations, however, to exclude 1) redundancies among augmentations and thereby make the learned weights interpretable and 2) augmentations are unlikely to be label-preserving. In particular, we exclude CutOut (because it is clearly not label-preserving in many domains) and exclude brightness, contrast, saturation, and color for their overlap with color-jitter. We also exclude contrast, because it is already modified via autocontrast, and equalize and solarize for their overlap with autocontrast and invert. This leaves us the following augmentations:

- *Shear*: Shear an image by some number of degrees, sampled between [-10, 10] (stochastic).
- *Translate*: Samples a vertical shift (by fraction of image height) from [0, .1] (stochastic).
- *Rotate*: Samples a rotation (by degrees) from [-10, 10] (stochastic).
- *Autocontrast*: Maximizes contrast of images by remapping pixel values such that the the lowest becomes black and the highest becomes white (deterministic).
- *Invert*: Inverts the colors of an image (deterministic).
- *Blur*: Applies Gaussian blur with kernel size 5 (and default σ range of [.1, .2]) (stochastic).
- *Posterize*: Reduces the number of bits per channel to 4 (deterministic).
- *Color Jitter*: Randomly samples a brightness, contrast, and saturation adjustment parameter from the range [.9, 1.1] (stochastic).
- *Increase Sharpness*: Adjusts sharpness of image by a factor of 1.3 (deterministic).
- *Decrease Sharpness*: Adjusts sharpness of image by a factor of 0.7 (deterministic).
- *Random Crop*: Pads each image by 4 pixels, takes a 256x256 crop, and then proceeds to take a 224x224 center crop (stochastic).
- *Horizontal Flip*: Flips image horizontally (deterministic).

There are many possible expanded test-time augmentation policies; this particular policy serves as an illustrative example.

A.2. Learning aggregation function $\hat{g}$

We learn $\hat{g}$ by minimizing the cross-entropy loss with respect to the true labels on the calibration set. Specifically, we learning the weights using SGD with a learning rate of .01, momentum of .9, and weight decay of 1e-4. We train each model for 50 epochs. There are natural improvements to this optimization, but this is not the focus of our work. Instead, our goal is to highlight the surprising effectiveness of TTA-Learned *without* the introduction of hyperparameter optimization.

A.3. Results of comparison to training on focal loss

We expand Table 1 to include results for a variant of TTA-Learned which uses a focal loss in place of the cross-entropy loss. We conduct this exploration because empirically, the focal loss has been known to produce better-calibrated models. In practice, we see little difference between results when using a different loss function; RAPS+TTA-Leanred still outperforms RAPS + an average over the test-time augmentations, and RAPS alone. While this speaks to the method's flexibility to different loss functions, it is possible that the use of a loss function designed to reduce prediction set size could produce better performance.

A.4. Results of comparison to different test-time augmentation weighting schemes

One could weight each test-time augmentation by the accuracy achieved on the set of examples used to learn the test-time augmentation policy. We show results of doing so in Table 4. We include two variants of this approach: one in which each augmentations predictions are weighted by the classification accuracy of that augmented prediction (TTA-Acc-Weighted), and one in which each augmentation's predictions are inversely weighted with respect to the classification error on the labeled dataset. Unsurprisingly, this type of approach places too much weight on unhelpful augmentations. Learning the

		Expanded Aug Policy		Simple Aug Policy	
Alpha	Method	ImageNet	CUB-Birds	ImageNet	CUB-Birds
0.01	RAPS+TTA-Learned+Focal	32.612 ± 3.799	13.416 ± 1.991	31.230 ± 1.510	15.503 ± 2.364
0.01	RAPS+TTA-Learned+Conformal	32.257 ± 3.608	13.776 ± 2.198	31.716 ± 2.078	14.432 ± 2.184
0.01	RAPS+TTA-Learned+CE	31.248 ± 2.177	14.045 ± 1.323	32.702 ± 2.409	13.803 ± 1.734
0.05	RAPS+TTA-Learned+Focal	4.906 ± 0.195	3.194 ± 0.202	4.956 ± 0.239	3.313 ± 0.331
0.05	RAPS+TTA-Learned+Conformal	4.867 ± 0.122	3.302 ± 0.312	4.996 ± 0.405	3.412 ± 0.406
0.05	RAPS+TTA-Learned+CE	4.889 ± 0.168	3.571 ± 0.576	5.040 ± 0.176	3.290 ± 0.186
0.10	RAPS+TTA-Learned+Focal	2.363 ± 0.085	1.791 ± 0.102	2.308 ± 0.045	1.860 ± 0.131
0.10	RAPS+TTA-Learned+Conformal	2.308 ± 0.068	1.865 ± 0.163	2.330 ± 0.072	1.868 ± 0.122
0.10	RAPS+TTA-Learned+CE	2.312 ± 0.054	1.893 ± 0.187	2.362 ± 0.065	1.840 ± 0.106

Table 3: Results across datasets for two augmentation policies and three coverage specifications using a focal loss. We set γ to be 1, in line with prior work (Einbinder et al.). Each entry corresponds to the average prediction set size across 10 calibration/test splits. Both the focal and conformal loss do not outperform the cross-entropy loss; for simplicity, we report all results using the cross-entropy loss.

weights directly produces the best performance using the expanded augmentation policy. Learning the weights has little effect with the simple augmentation policy (a consistent result across all experiments).

A.5. Results of comparison to Top-1 and Top-5

We expand Table 1 to include the Top-1 and Top-5 baselines in Table 5. Unsurprisingly, neither outperform RAPS, and consequently none outperform the combination of RAPS, TTA-Learned, and the expanded augmentation policy.

A.6. Results using APS

TTA-Learned combined with the expanded augmentation policy produces the smallest set sizes when combined with APS, across the datasets considered (Table 6) and each base classifier (Table 8). In contrast to the results using RAPS, TTA-Learned does not significantly outperform TTA-Avg when combined with APS. The central reason is that the improvements TTA confers — namely, improved top-k accuracy — do not address the underlying sensitivity of APS to classes with low predicted probabilities. As Angelopoulos et al. (2022) discuss, APS produces large prediction sets because of noisy estimates of small probabilities, which then end up included in the prediction sets. Both TTA-Learned and TTA-Avg smooth the probabilities: they reduce the number of low-probability classes by aggregating predictions over perturbations of the image. The benefit that both TTA-Learned and TTA-Avg add to APS is thus similar to how RAPS penalizes classes with low probabilities.

A.7. Results on coverage

We provide exact values of coverage for each experiment here. In short, TTA-Learned combined with the expanded augmentation policy *never* worsens coverage, and in some cases, significantly improves it (although the improvements are small in magnitude). For those interested, we mirror each table describing average prediction set size with a table describing average coverage: coverage values for the RAPS experiment across coverage values and datasets can be found in Table 10 and coverage values for the RAPS experiment across base classifiers can be found in Table 11. Similarly, we provide coverage values for the APS experiment across datasets (Table 7) and across models (Table 8).

A.8. Replicated results with different alphas, datasets

We replicate the class-specific analysis for ImageNet at a value of $\alpha = .05$ (Figure 4), iNaturalist (Figure 5), and CUB-Birds (Figure 6). All trends are consistent with results in the main text, save for one notable exception: when TTA-Learned is applied to CUB-Birds, prediction set sizes of the classes with the *smallest* prediction set sizes and classes that are

		Expanded Aug Policy	Simple Aug Policy
Alpha	Method	ImageNet	ImageNet
0.01	RAPS+TTA-Avg	35.600 ± 2.200	**31.681 ± 3.057**
0.01	RAPS+TTA-Acc-Weighted	37.115 ± 4.112	**33.561 ± 5.174**
0.01	RAPS+TTA-Err-Weighted	36.012 ± 3.501	**33.415 ± 2.619**
0.01	RAPS+TTA-Learned	**31.723 ± 1.737**	**32.702 ± 2.409**
0.05	RAPS+TTA-Avg	5.318 ± 0.113	**4.908 ± 0.099**
0.05	RAPS+TTA-Acc-Weighted	5.258 ± 0.171	**4.942 ± 0.242**
0.05	RAPS+TTA-Err-Weighted	5.352 ± 0.366	**4.859 ± 0.139**
0.05	RAPS+TTA-Learned	**4.897 ± 0.304**	**5.040 ± 0.176**
0.10	RAPS+TTA-Avg	2.470 ± 0.071	**2.327 ± 0.086**
0.10	RAPS+TTA-Acc-Weighted	2.443 ± 0.068	**2.352 ± 0.085**
0.10	RAPS+TTA-Err-Weighted	2.416 ± 0.076	**2.348 ± 0.065**
0.10	RAPS+TTA-Learned	**2.290 ± 0.064**	**2.362 ± 0.065**

Table 4: Results comparing learned weights to no augmentation-specific weights (TTA-Avg) and weights inferred from each test-time augmentation's accuracy (TTA-Acc-Weighted) or error (inverse weighting with respect to 1 - *aug_acc*). These results show that naive methods to weight the test-time augmentations can improve upon no learned weights at all, but learning the weights directly produces the best performance.

		ImageNet		iNaturalist		CUB-Birds	
Alpha	Method	Prediction Set Size	Empirical Coverage	Prediction Set Size	Empirical Coverage	Prediction Set Size	Empirical Coverage
0.01	Top-1	1.000 ± 0.000	0.761 ± 0.002	1.000 ± 0.000	0.766 ± 0.001	1.000 ± 0.000	0.804 ± 0.008
0.01	Top-5	5.000 ± 0.000	0.928 ± 0.001	5.000 ± 0.000	0.915 ± 0.001	5.000 ± 0.000	0.959 ± 0.003
0.01	RAPS	37.751 ± 2.334	0.990 ± 0.001	61.437 ± 6.067	0.990 ± 0.001	15.293 ± 2.071	0.990 ± 0.001
0.01	RAPS+TTA-Avg	35.600 ± 2.200	0.991 ± 0.001	57.073 ± 5.914	0.990 ± 0.001	13.111 ± 2.470	0.991 ± 0.002
0.01	RAPS+TTA-Learned	31.248 ± 2.177	0.990 ± 0.001	53.195 ± 4.884	0.990 ± 0.001	14.045 ± 1.323	0.991 ± 0.002
0.05	Top-1	1.000 ± 0.000	0.761 ± 0.002	1.000 ± 0.000	0.766 ± 0.001	1.000 ± 0.000	0.804 ± 0.008
0.05	Top-5	5.000 ± 0.000	0.928 ± 0.001	5.000 ± 0.000	0.915 ± 0.001	5.000 ± 0.000	0.959 ± 0.003
0.05	RAPS	5.637 ± 0.357	0.951 ± 0.002	7.991 ± 1.521	0.954 ± 0.002	3.624 ± 0.361	0.955 ± 0.007
0.05	RAPS+TTA-Avg	5.318 ± 0.113	0.951 ± 0.001	7.067 ± 0.344	0.952 ± 0.002	3.116 ± 0.210	0.954 ± 0.007
0.05	RAPS+TTA-Learned	4.889 ± 0.168	0.952 ± 0.001	6.682 ± 0.447	0.954 ± 0.002	3.571 ± 0.576	0.957 ± 0.007
0.10	Top-1	1.000 ± 0.000	0.761 ± 0.002	1.000 ± 0.000	0.766 ± 0.001	1.000 ± 0.000	0.804 ± 0.008
0.10	Top-5	5.000 ± 0.000	0.928 ± 0.001	5.000 ± 0.000	0.915 ± 0.001	5.000 ± 0.000	0.959 ± 0.003
0.10	RAPS	2.548 ± 0.074	0.906 ± 0.004	2.914 ± 0.116	0.907 ± 0.003	2.038 ± 0.153	0.919 ± 0.014
0.10	RAPS+TTA-Avg	2.470 ± 0.071	0.905 ± 0.005	2.740 ± 0.026	0.908 ± 0.002	1.780 ± 0.139	0.912 ± 0.014
0.10	RAPS+TTA-Learned	2.312 ± 0.054	0.905 ± 0.004	2.625 ± 0.043	0.909 ± 0.003	1.893 ± 0.187	0.919 ± 0.016

Table 5: Results comparing performance against Top-K baselines. In each setting, conformal prediction produces either smaller set sizes, higher coverage, or both compared to the Top-K baselines.

		Expanded Aug Policy			Simple Aug Policy		
Alpha	Method	ImageNet	iNaturalist	CUB-Birds	ImageNet	iNaturalist	CUB-Birds
0.01	APS	98.493 ± 3.075	131.681 ± 3.515	19.436 ± 0.995	98.493 ± 3.075	**131.681 ± 3.515**	**19.436 ± 0.995**
0.01	APS+TTA-Avg	**68.714 ± 2.856**	**84.546 ± 3.655**	**17.715 ± 1.523**	**92.027 ± 4.797**	145.401 ± 4.635	**19.152 ± 1.667**
0.01	APS+TTA-Learned	**69.009 ± 2.156**	**85.093 ± 2.768**	**17.766 ± 1.608**	**90.613 ± 6.421**	144.134 ± 4.371	**18.552 ± 1.326**
0.05	APS	19.820 ± 0.482	33.481 ± 0.786	5.921 ± 0.192	19.820 ± 0.482	**33.481 ± 0.786**	**5.921 ± 0.192**
0.05	APS+TTA-Avg	14.308 ± 0.279	**26.021 ± 0.282**	**4.870 ± 0.208**	**18.862 ± 0.498**	37.370 ± 0.735	6.306 ± 0.350
0.05	APS+TTA-Learned	**14.084 ± 0.241**	**26.289 ± 0.529**	**4.913 ± 0.145**	**19.119 ± 0.479**	36.940 ± 0.632	6.361 ± 0.480
0.10	APS	8.969 ± 0.158	16.755 ± 0.394	3.455 ± 0.164	8.969 ± 0.158	**16.755 ± 0.394**	**3.455 ± 0.164**
0.10	APS+TTA-Avg	**7.193 ± 0.101**	**14.583 ± 0.333**	**3.108 ± 0.114**	**8.787 ± 0.136**	18.300 ± 0.418	**3.609 ± 0.135**
0.10	APS+TTA-Learned	**7.215 ± 0.106**	**14.538 ± 0.395**	**3.046 ± 0.073**	**8.813 ± 0.180**	18.086 ± 0.420	3.638 ± 0.146

Table 6: We replicate our experiments across coverage levels and datasets using APS, another conformal score. TTA-Learned combined with the expanded augmentation policy produces the smallest set sizes across all comparisons. Interestingly, the simple augmentation policy is not as effective in the context of iNaturalist when using APS.

		Expanded Aug Policy			Simple Aug Policy		
Alpha	Method	ImageNet	iNaturalist	CUB-Birds	ImageNet	iNaturalist	CUB-Birds
0.01	APS	0.980 ± 0.001	0.986 ± 0.000	0.985 ± 0.001	0.980 ± 0.001	**0.986 ± 0.000**	**0.985 ± 0.001**
0.01	APS+TTA-Avg	**0.985 ± 0.001**	**0.989 ± 0.001**	**0.989 ± 0.002**	**0.981 ± 0.001**	0.987 ± 0.000	**0.986 ± 0.003**
0.01	APS+TTA-Learned	**0.985 ± 0.001**	**0.989 ± 0.001**	**0.990 ± 0.002**	**0.980 ± 0.002**	0.987 ± 0.000	**0.985 ± 0.002**
0.05	APS	0.931 ± 0.002	0.952 ± 0.001	0.945 ± 0.004	0.931 ± 0.002	**0.952 ± 0.001**	**0.945 ± 0.004**
0.05	APS+TTA-Avg	0.944 ± 0.002	**0.956 ± 0.001**	**0.949 ± 0.005**	**0.937 ± 0.002**	0.960 ± 0.001	0.949 ± 0.004
0.05	APS+TTA-Learned	**0.943 ± 0.002**	**0.957 ± 0.001**	**0.950 ± 0.005**	**0.937 ± 0.002**	0.959 ± 0.001	0.950 ± 0.005
0.10	APS	0.896 ± 0.002	0.923 ± 0.001	0.915 ± 0.006	0.896 ± 0.002	**0.923 ± 0.001**	**0.915 ± 0.006**
0.10	APS+TTA-Avg	**0.903 ± 0.002**	**0.930 ± 0.001**	**0.920 ± 0.007**	**0.905 ± 0.002**	0.933 ± 0.001	**0.922 ± 0.005**
0.10	APS+TTA-Learned	**0.904 ± 0.002**	**0.930 ± 0.001**	**0.918 ± 0.006**	**0.906 ± 0.002**	0.932 ± 0.001	0.922 ± 0.004

Table 7: Coverage values associated with experiments in Table 6. TTA-Learned produces significant improvements in coverage — larger in magnitude than in conjunction with RAPS — across when using the expanded augmentation policy. TTA-Learned produces no drops in coverage when using the simple augmentation policy, a nd produces improvements at $\alpha = .01$ and $\alpha = .05$.

easier to predict benefit most from TTA. The significance of the relationship between original prediction set size and TTA improvement disappears when conducted on an example level in this setting. This could be a result of class imbalance in the dataset; it is possible that the class-average prediction set size obscures important variation in CUB-Birds.

A.9. Impact of augmentation policy size

We also analyze the impact of augmentation policy size on average prediction set size for CUB-Birds (Figure **??**), to understand if additional augmentations may produce larger reductions in set size than we observe. Larger augmentation policies appear to provide an improvement to average prediction set size at $\alpha = .05$, but offer little improvement for $\alpha = .01$.

A.10. Impact of TTA data split

Learning the test-time augmentation policy requires a set of labeled data *distinct* from those used to select the conformal threshold. This introduces a trade-off: more labeled data for test-time augmentation may result in more accurate weights, but a less accurate conformal threshold, and vice versa. We study this tradeoff empirically in the context of ImageNet and the expanded augmentation policy and show results in Figure 8. We find that, as more data is taken away from the conformal calibration set, variance in performance grows. This is in line with our intuition; we have fewer examples to approximate the

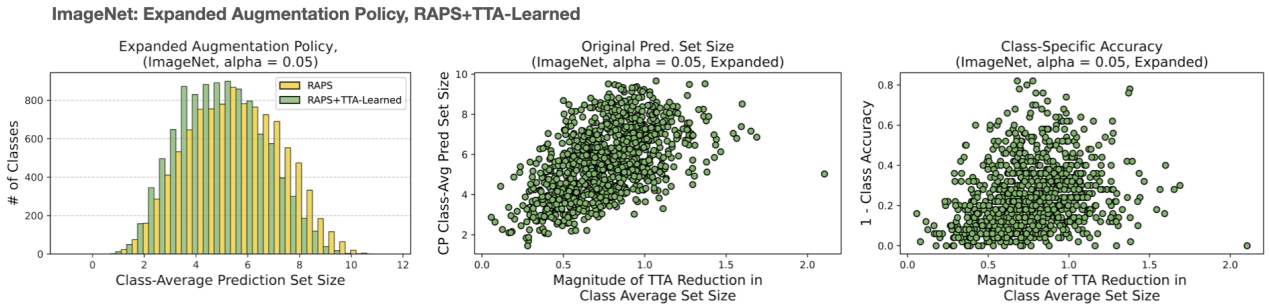

Figure 4: Class-specific performance for ImageNet, for a coverage of 95% $\alpha = .05$. Using the expanded augmentation policy RAPS+TTA-Learned produces a noticeable shift in class-average prediction set sizes to the left. There is a significant correlation between original prediction set size and improvements from TTA (middle) and between class difficulty and improvements from TTA (right).

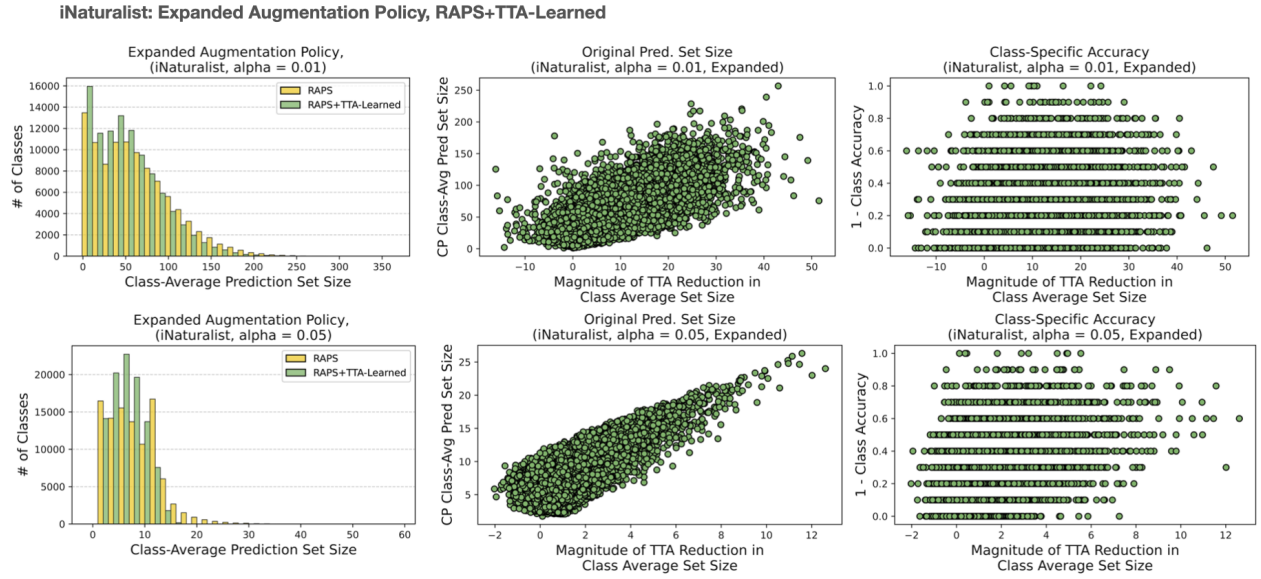

Figure 5: Class-specific performance for iNaturalist, for $\alpha = .01$ (top) and $\alpha = .05$ (bottom). We see a consistent relationship between TTA improvements and original class-average prediction set size (middle) and class difficulty (right). Estimates of class-specific accuracy on iNaturalist are quite noisy because there are 10 images per class (which produces distinct accuracy bands).

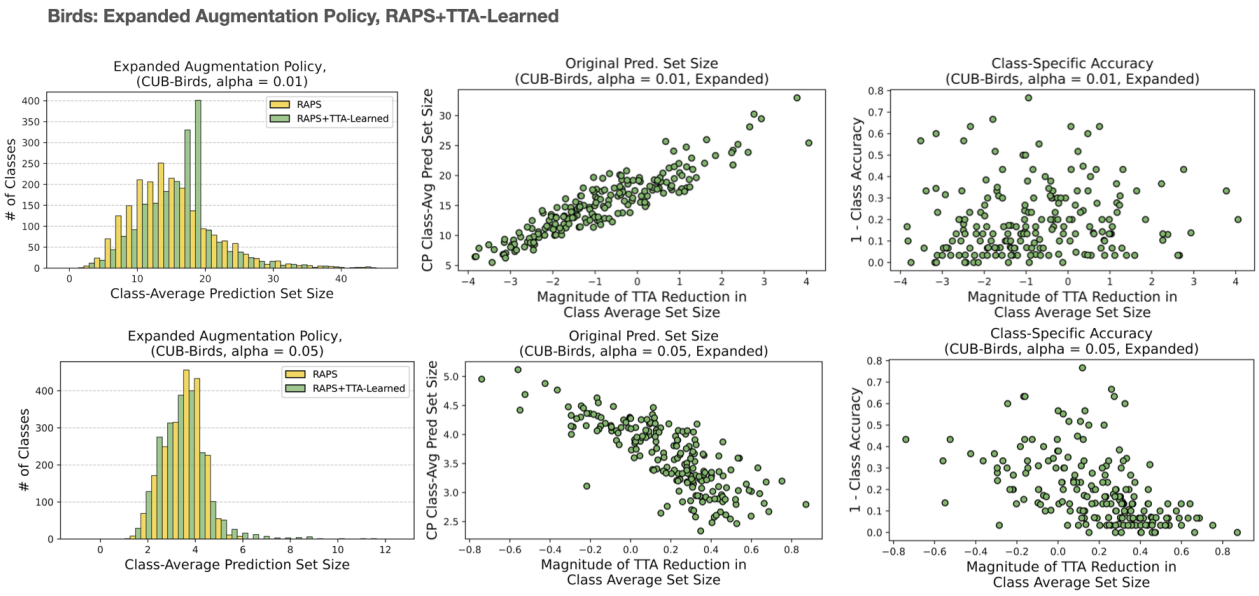

Figure 6: Class-specific performance for CUB-Birds, for $\alpha = .01$ (top) and $\alpha = .05\%$ (bottom). These graphs show an example for which TTA-Learned does *not* produce improvements in average prediction set size (computed across all examples). Interestingly, behavior on a class-specific level is different between $\alpha = .01$ and $\alpha = .05$. For $\alpha = .01$, results are consistent with other datasets: classes which originally receive large prediction set sizes and classes which are more difficult benefit most from the addition of TTA. For $\alpha = .05$, the exact opposite is true. While a majority of classes are hurt by TTA, classes that benefit from TTA are easier and receive smaller prediction set sizes.

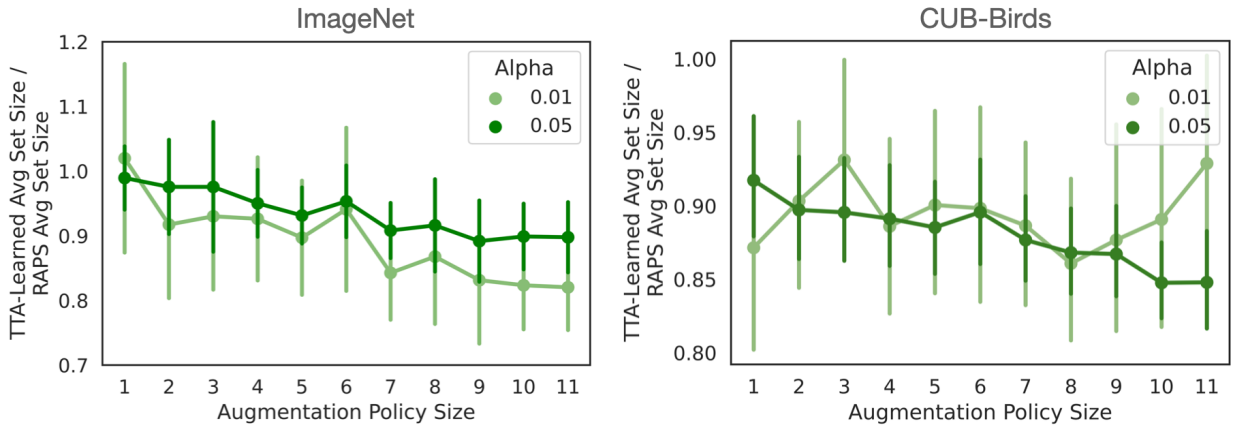

Figure 7: Impact of augmentation policy size on ImageNet (left) and CUB-Birds (right). We see that larger policy sizes translate to a greater improvement (in terms of the ratio of average prediction set sizes using RAPS+TTA-Learned to average prediction set sizes using RAPS alone) for $\alpha = .05$. For $\alpha = .01$, there is no clear trend.

		Expanded Aug Policy			Simple Aug Policy		
Alpha	Method	ResNet-50	ResNet-101	ResNet-152	ResNet-50	ResNet-101	ResNet-152
0.01	APS	98.493 ± 3.075	88.279 ± 4.121	79.231 ± 4.570	98.493 ± 3.075	88.279 ± 4.121	79.231 ± 4.570
0.01	APS+TTA-Avg	**68.714 ± 2.856**	**64.197 ± 2.336**	**62.885 ± 3.125**	**92.027 ± 4.797**	**77.344 ± 2.214**	**73.377 ± 3.600**
0.01	APS+TTA-Learned	**69.009 ± 2.156**	**64.852 ± 2.823**	**64.045 ± 3.398**	**90.613 ± 6.421**	**78.627 ± 4.101**	**74.571 ± 3.516**
0.05	APS	19.820 ± 0.482	15.830 ± 0.611	14.437 ± 0.591	19.820 ± 0.482	15.830 ± 0.611	**14.437 ± 0.591**
0.05	APS+TTA-Avg	14.308 ± 0.279	**11.085 ± 0.267**	**10.605 ± 0.373**	**18.862 ± 0.498**	**15.039 ± 0.405**	**14.206 ± 0.499**
0.05	APS+TTA-Learned	**14.084 ± 0.241**	**11.118 ± 0.209**	**10.595 ± 0.368**	**19.119 ± 0.479**	**15.011 ± 0.346**	**14.252 ± 0.486**
0.10	APS	8.969 ± 0.158	6.671 ± 0.175	6.134 ± 0.163	8.969 ± 0.158	**6.671 ± 0.175**	**6.134 ± 0.163**
0.10	APS+TTA-Avg	**7.193 ± 0.101**	**5.454 ± 0.098**	**5.111 ± 0.096**	**8.787 ± 0.136**	6.838 ± 0.143	6.309 ± 0.178
0.10	APS+TTA-Learned	**7.215 ± 0.106**	**5.490 ± 0.090**	**5.131 ± 0.061**	**8.813 ± 0.180**	6.826 ± 0.121	6.311 ± 0.123

Table 8: Results across base classifiers using APS alone, APS + TTA-Avg, and APS + TTA-learned in conjunction with the expanded augmentation policy (left) and simple augmentation policy (right). TTA-Learned and the expanded augmentation policy produce the smallest prediction sets (on average).

		Expanded Aug Policy			Simple Aug Policy		
Alpha	Method	ResNet-50	ResNet-101	ResNet-152	ResNet-50	ResNet-101	ResNet-152
0.01	APS	0.980 ± 0.001	0.979 ± 0.002	0.978 ± 0.002	**0.980 ± 0.001**	**0.979 ± 0.002**	**0.978 ± 0.002**
0.01	APS+TTA-Avg	**0.985 ± 0.001**	**0.985 ± 0.001**	**0.984 ± 0.001**	**0.981 ± 0.001**	**0.980 ± 0.001**	**0.978 ± 0.002**
0.01	APS+TTA-Learned	**0.985 ± 0.001**	**0.985 ± 0.001**	**0.984 ± 0.001**	**0.980 ± 0.002**	**0.980 ± 0.002**	**0.979 ± 0.002**
0.05	APS	0.931 ± 0.002	0.930 ± 0.002	0.929 ± 0.002	0.931 ± 0.002	0.930 ± 0.002	0.929 ± 0.002
0.05	APS+TTA-Avg	**0.944 ± 0.002**	**0.942 ± 0.001**	**0.942 ± 0.002**	**0.937 ± 0.002**	**0.935 ± 0.002**	**0.934 ± 0.002**
0.05	APS+TTA-Learned	0.943 ± 0.002	**0.942 ± 0.001**	**0.942 ± 0.002**	**0.937 ± 0.002**	**0.935 ± 0.001**	**0.934 ± 0.002**
0.10	APS	0.896 ± 0.002	0.892 ± 0.002	0.893 ± 0.002	0.896 ± 0.002	0.892 ± 0.002	0.893 ± 0.002
0.10	APS+TTA-Avg	**0.903 ± 0.002**	**0.901 ± 0.001**	**0.902 ± 0.001**	**0.905 ± 0.002**	**0.903 ± 0.001**	**0.903 ± 0.002**
0.10	APS+TTA-Learned	**0.904 ± 0.002**	**0.902 ± 0.001**	**0.902 ± 0.001**	**0.906 ± 0.002**	**0.903 ± 0.002**	**0.903 ± 0.002**

Table 9: Coverage values for APS and TTA variants of APS across base classifiers, using ImageNet. TTA-Learned or TTA-Avg in combination with the expanded augmentation policy significantly improve coverage in every comparison.

distribution of conformal scores. However, at all percentages, test-time augmentation introduces a significant improvement in prediction set sizes over using all the labeled examples, and their original probabilities, to determine the threshold. This suggests that the benefits TTA confers outweigh the costs to the estimation of the conformal threshold, a practically useful insight to those who wish to apply conformal prediction in practice6

A.11. Impact of calibration set size

We plot the relationship between calibration set size and average prediction set size in Figure 9 across two augmentation policies, two datasets, and two values of α. We see that TTA is more effective the larger the calibration set, in the context of ImageNet. In the context of CUB-Birds, it appears that TTA approaches equivalence with the conformal score alone as the calibration set size increases.

A.12. TTA's effect on optimal Top-k for a given coverage α

As discussed in text, test-time augmentation improves the performance of conformal predictions by improving the top-k accuracy of the resulting probabilities, for some k. One way to understand this difference is to compare what value of k_{opt} is necessary for a given coverage α. Networks with higher top-k accuracy produce lower values of k_{opt} than networks with low top-k accuracy. We visualize the difference in the optimal k for TTA-Learned probabilities compared to the original probabilities in Figure 10.

		Expanded Aug Policy			Simple Aug Policy		
Alpha	Method	ImageNet	iNaturalist	CUB-Birds	ImageNet	iNaturalist	CUB-Birds
0.01	RAPS	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**
0.01	RAPS+TTA-Avg	**0.991 ± 0.001**	**0.990 ± 0.001**	**0.991 ± 0.002**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.991 ± 0.002**
0.01	RAPS+TTA-Learned	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.991 ± 0.002**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.002**
0.05	RAPS	**0.951 ± 0.002**	**0.954 ± 0.002**	**0.955 ± 0.007**	**0.951 ± 0.002**	**0.954 ± 0.002**	**0.955 ± 0.007**
0.05	RAPS+TTA-Avg	**0.951 ± 0.001**	**0.952 ± 0.002**	**0.954 ± 0.007**	**0.951 ± 0.001**	**0.953 ± 0.003**	**0.957 ± 0.004**
0.05	RAPS+TTA-Learned	**0.952 ± 0.001**	**0.954 ± 0.002**	**0.957 ± 0.007**	**0.951 ± 0.002**	**0.952 ± 0.002**	**0.956 ± 0.007**
0.10	RAPS	**0.906 ± 0.004**	**0.907 ± 0.003**	**0.919 ± 0.014**	**0.906 ± 0.004**	**0.907 ± 0.003**	**0.919 ± 0.014**
0.10	RAPS+TTA-Avg	**0.905 ± 0.005**	**0.908 ± 0.002**	**0.912 ± 0.014**	**0.905 ± 0.004**	**0.908 ± 0.002**	**0.915 ± 0.010**
0.10	RAPS+TTA-Learned	**0.905 ± 0.004**	**0.909 ± 0.003**	**0.919 ± 0.016**	**0.907 ± 0.004**	**0.908 ± 0.003**	**0.913 ± 0.011**

Table 10: Coverage values for RAPS, RAPS+TTA-Avg, and RAPS+TTA-Learned across datasets and coverage values. RAPS+TTA-Learned never decreases the coverage achieved by RAPS alone, and in some cases, improves it significantly (as in the case of ImageNet and iNaturalist).

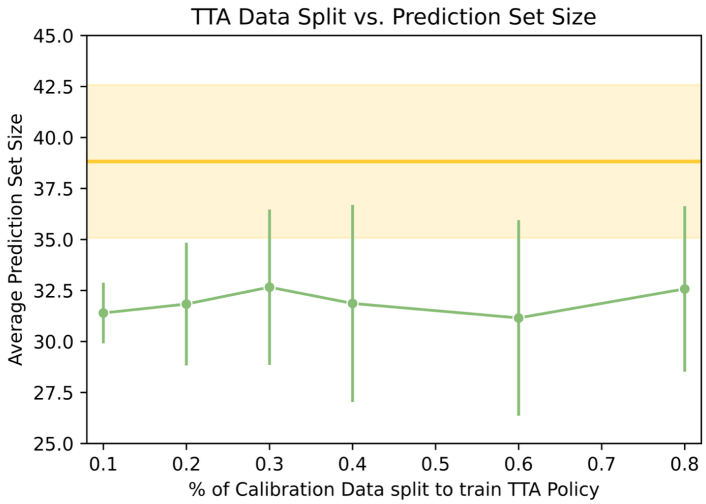

Figure 8: We plot the percentage of data used to train the TTA policy on the x-axis and the average prediction set size on the y-axis. Error bars describe variance over 10 random splits of the calibration and test set. We can make two observations: 1) as the data used to train the TTA policy increases and the data used to estimate the conformal threshold decreases, variance in performance grows and 2) across a wide range of data splits, learned TTA policies (green) introduce improvements to achieved prediction set sizes compared to the original probabilities (gold). These results also suggest that relatively little training data is required to learn a useful test-time augmentation policy; in this case, 2-3 images per class, or 10% of the available labeled data.

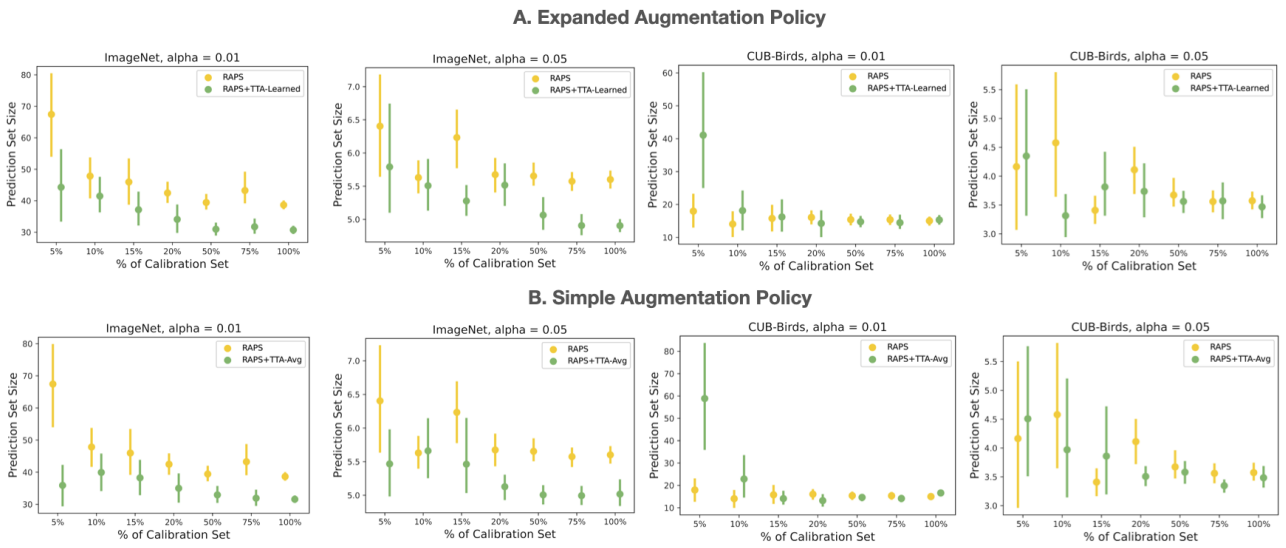

Figure 9: We plot the relationship between calibration set size and average prediction set size across two values of alpha, two augmentation policies, and two datasets (ImageNet and CUB-Birds). For ImageNet, larger calibration set sizes correlate with larger and more consistent improvements from the addition of TTA, where the improvement flattens out for calibration set sizes larger than 50%, or 12,500 images (12-13 per class). TTA does appear to be able to improve average prediction set size even with a calibration set size of 1,250 (5% of original ImageNet calibration set size). For CUB-Birds, a dataset on which TTA does not perform as well, we see that TTA performs comparably to RAPS alone the larger the calibration set.

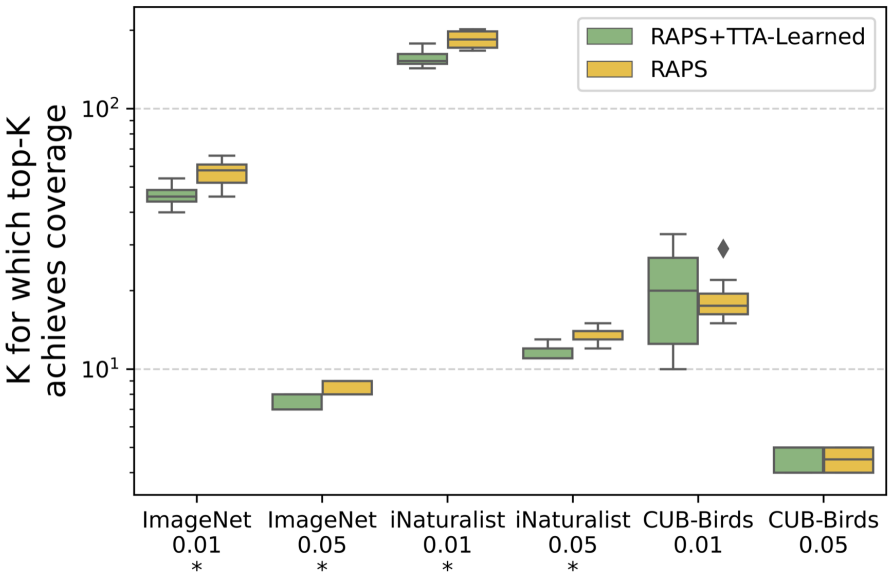

Figure 10: We plot the distribution of optimal k for each dataset given two coverage values (.01 and .05). Probabilities transformed by TTA-Learned produce significantly lower values for k (measured using a pairwise t-test) than the original probabilities on both ImageNet and iNaturalist, two datasets for which test-time augmentation produces consistent improvements.

		Expanded Aug Policy			Simple Aug Policy		
Alpha	Method	ResNet-50	ResNet-101	ResNet-152	ResNet-50	ResNet-101	ResNet-152
0.01	RAPS	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**
0.01	RAPS+TTA-Avg	**0.991 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**
0.01	RAPS+TTA-Learned	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**	**0.990 ± 0.001**
0.05	RAPS	**0.951 ± 0.002**	**0.952 ± 0.002**	**0.952 ± 0.002**	**0.951 ± 0.002**	**0.952 ± 0.002**	**0.952 ± 0.002**
0.05	RAPS+TTA-Avg	**0.951 ± 0.001**	**0.951 ± 0.001**	**0.952 ± 0.002**	**0.951 ± 0.001**	**0.952 ± 0.002**	**0.952 ± 0.002**
0.05	RAPS+TTA-Learned	**0.952 ± 0.001**	**0.952 ± 0.002**	**0.952 ± 0.002**	**0.951 ± 0.002**	**0.952 ± 0.002**	**0.952 ± 0.002**
0.10	RAPS	**0.906 ± 0.004**	**0.906 ± 0.004**	0.906 ± 0.002	**0.906 ± 0.004**	0.906 ± 0.004	0.906 ± 0.002
0.10	RAPS+TTA-Avg	**0.905 ± 0.005**	0.905 ± 0.002	0.908 ± 0.002	**0.905 ± 0.004**	**0.908 ± 0.004**	**0.910 ± 0.002**
0.10	RAPS+TTA-Learned	**0.905 ± 0.004**	**0.907 ± 0.003**	**0.911 ± 0.002**	**0.907 ± 0.004**	**0.908 ± 0.004**	**0.910 ± 0.002**

Table 11: Coverage values for TTA variants of conformal prediction compared to RAPS alone, across different base classifiers on ImageNet. TTA-Learned preserves coverage across all comparisons and significantly improves upon the achieved coverage using ResNet-101 with RAPS (granted, the magnitude of this improvement is small).

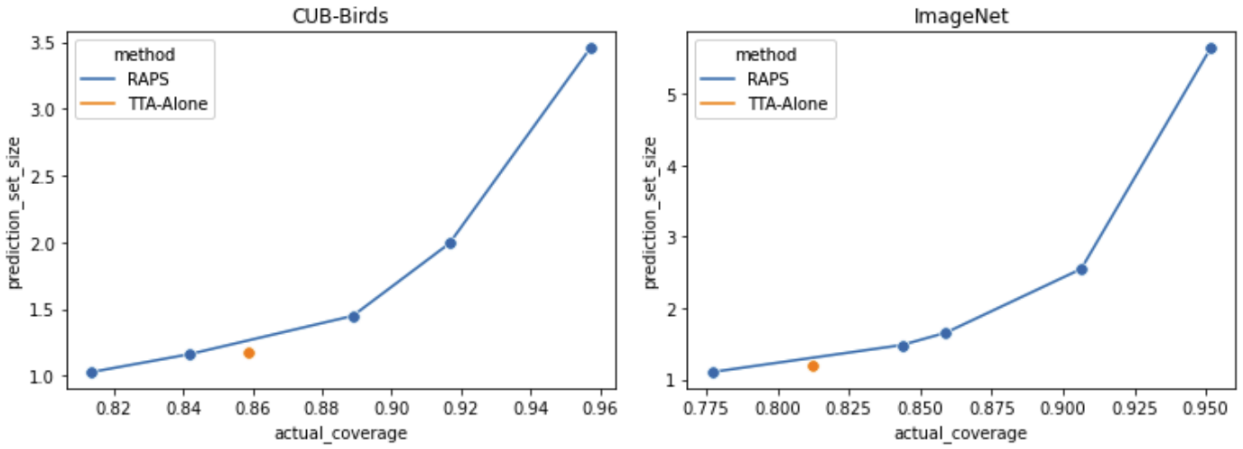

Figure 11: Comparison of uncertainty sets produced using the simple augmentation policy (orange) compared to the tradeoff RAPS achieves between prediction set size and coverage (blue).

A.13. TTA Uncertainty Sets

What if we instead generated uncertainty sets by creating a set out of the predictions made on each augmentations in a TTA policy? Interestingly, this approach can provide marginal improvements compared to the RAPS tradeoff between prediction set size and coverage—see Figure 11 for a comparison with the simple test-time augmentation policy. The sets are far less practically useful compared to those produced by a conformal predictor, but these differences may suggest ways to further improve the efficiency of conformal predictors.