Test-time augmentation improves efficiency in conformal prediction

Divya Shanmugam, Helen Lu, Swami Sankaranarayanan, John Guttag
Massachusetts Institute of Technology, CSAIL
{divyas, helenlu, swamiviv, guttag}@mit.edu

Abstract

A conformal classifier produces a set of predicted classes and provides a probabilistic guarantee that the set includes the true class. Unfortunately, it is often the case that conformal classifiers produce uninformatively large sets. In this work, we show that test-time augmentation (TTA)–a technique that introduces inductive biases during inference–reduces the size of the sets produced by conformal classifiers. 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 10%-14% on average. We conduct an evaluation of the approach spanning three datasets, three models, two established conformal scoring methods, different guarantee strengths, and several distribution shifts to show when and why test-time augmentation is a useful addition to the conformal pipeline.

1. Introduction

Conformal prediction has emerged as a promising way to equip existing classifiers with statistically valid uncertainty estimates. It does so 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 [36].

Conformal prediction faces two limitations in practice. First, achieving a suitably strong guarantee often leads to prediction sets that are uninformatively large [10, 47]. Large prediction sets have been shown to *decrease* performance when provided as a decision-making aid [3, 10, 47].

Second, conformal classifiers inherit the instability of the underlying models. As a result, prediction sets can change significantly in response to small input perturbations, a well-known weakness of neural networks [39]. Applying a horizontal flip to each image in ImageNet, for example, changes the prediction set sizes for 75% of examples at a coverage guarantee of 99%. Such behavior also represents a barrier to broader use.

In this work, we show that test-time augmentation (TTA), a widely-used technique in computer vision, has the potential to address both limitations. TTA involves generating an ensemble of predictions by perturbing the input with label-preserving transformations. It has previously been shown that TTA can be used to make non-conformal classifiers more robust to small input perturbations [7], more accurate [37], and better calibrated [18]. However, previous work has not explored the utility of TTA in the context of conformal prediction.

We propose *test-time-augmented conformal prediction*, which transforms a classifier's predictions using a learned test-time augmentation policy prior to conformal classification. By using distinct sets of labeled data to learn the test-time augmentation policy and the conformal classifier, we preserve the assumption of exchangeability, and thereby the coverage guarantee associated with the conformal predictor.

In experiments testing the performance of conformal predictors subject to distribution shift, we see that test-time augmentation reduces prediction set sizes by 14% on average, with no loss of coverage. And even when there is no distribution shift, we see a reduction of 10% on average. Moreover, we find that classes with the largest average prediction set sizes benefit most from the introduction of test-time augmentation. We also show that test-time augmentation can bridge gaps between classifiers of different sizes (e.g., we show that test-time augmentation combined with ResNet-50 produces smaller set sizes than ResNet-101 without test-time augmentation).

Our analysis of *why* test-time augmentation reduces prediction set sizes reveals a previously unknown effect of test-time augmentation. Specifically, TTA 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). Although such behavior has no impact on the maximum predicted probability — commonly the focus of literature on test-time augmentation [2, 32, 37] — it is valuable in conformal prediction. This is because when the true class is promoted to a higher rank among a classifier's predicted probabilities, the conformal classifier includes fewer incorrect classes to meet the conformal guarantee.

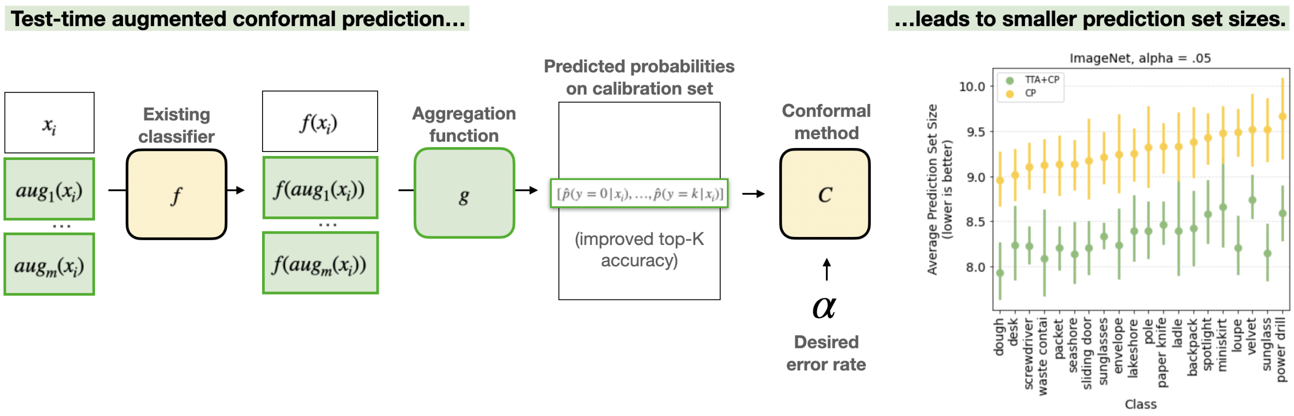

Figure 1. We illustrate the addition of test-time augmentation to conformal calibration in green (left) and provide a snapshot of the improvements it can confer (right). We show results on Imagenet, with a desired coverage of 95%, for the 20 classes with the largest predicted set sizes on average (computed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

Contributions We make three contributions. To begin, this is the first work to propose combining test-time augmentation, a popular technique in computer vision, with conformal prediction. Second, we present a method that reduces the prediction set sizes of existing conformal predictors by using *automatically learned* test-time augmentations. Finally, we demonstrate, in an extensive set of experiments, that our approach to combining conformal prediction and test-time augmentation leads to smaller prediction sets.

2. Related work

Conformal prediction was first introduced by Gammerman et al. [16], and further developed by Saunders and Holloway [35] and Vladmir Vovk [42]. 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 [43], bootstrap conformal prediction [44], aggregated conformal prediction [5, 25], and out-of-bag conformal prediction [26]. The approaches primarily differ in how data is sampled to create the training dataset for the classifier and the calibration dataset for the conformal predictor. 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 classifier. Many works have proposed new procedures to estimate and apply thresholds on conformal scores [1, 4, 13, 33, 40]. Romano et al. [34] proposed APS, a conformal score based on the cumulative probability required to include the correct class in a prediction set. Angelopoulos et al. [1] 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. [20] ensemble a set of base classifiers, where the classifiers are created by training models on subsets of the training data. Stutz et al. [38] provide a new way to train the base classifier and conformal predictor jointly through a novel 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.

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 [2, 9, 15, 18, 32, 37, 48]. TTA has been applied to a diverse range of predictive tasks across domains ranging from healthcare [8] to content moderation [27]. Consequently, many have proposed new ways to perform TTA—for example, learning when to apply TTA [29], which augmentations to use [6, 21, 28], and how to aggregate the resulting predictions [6, 9, 37]. 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** $f : \mathcal{X} \mapsto \Delta^K$, mapping input domain $\mathcal{X}$ to a probability distribution over K classes.
- **Desired upper bound on error rate** $\alpha \in [0,1]$, where $(1-\alpha)$ is the probability the set contains the true class.

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

- **Augmentation policy** $\mathcal{A} = \{a_0,\ldots,a_{m-1}\}$, consisting of m augmentation functions, where a_0 is the identity transform.
- **Aggregation function** $\hat{g}$, which aggregates a set of predictions to produce one prediction.

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

4. Approach

Our goal is to learn 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 briefly outline the conformal approach, and then detail the mechanics of our method (illustrated in Figure 1). For a detailed introduction to conformal prediction, refer to Shafer and Vovk [36].

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 $c(x_i, y_i)$, which produces a *conformal score* representing the uncertainty of the input example and label pair.
2. Produce a distribution of conformal scores 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, combined with $\{\infty\}$.

For a new example x, we compute $c(x_i, y)$ for all $y \in \{1,\ldots,K\}$, and include all y for which $c(x_i, y) < \hat{q}$. We adopt the conformal score proposed by Romano et al. [34], which equates to the cumulative probability required to include the correct class:

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

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

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

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. The variable $u \sim U(0,1)$. Conformal score $c(x_i, y_i)$ is thus composed of this cumulative probability and the predicted probability of class y_i.

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.

$$\hat{\pi}_x(y') = \hat{p}(y = y'|x_i) = g(x_i; f, \mathcal{A}, \theta) \tag{4}$$

Aggregation weights θ are applied to the logits output by classifier f, and transformed to be proper probabilities by applying a softmax function. We 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 learn the test-time augmentation policy.

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, g applies θ and $\mathcal{A}$ as follows:

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

where $\mathbf{A}$ uses f to map input x_i to a $M \times K$ matrix of predicted logits where M is the number of augmentations 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 logits on augmentation a_m of x_i. TTA-Learned refers to TTA combined with learned augmentation weights, while TTA-Avg refers to a simple average over the augmentations.

[1] We found no significant improvement by using alternate losses considered in the conformal prediction literature (e.g. focal or conformal training loss). See Table S7 in the Appendix.

We refer to the fraction of the validation set allotted to $D^{(TTA)}$ as β. Figure S3 shows that performance is not sensitive to the choice of β; as a result, all experiments use $\beta = .2$ (see supplement 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). Prior work has shown that the conformal procedure is valid under deterministic transformations [24]; by using distinct examples to learn the test-time augmentation policy, the proposed approach constitutes a deterministic transformation applied to both the calibration set and unseen examples. If we were to instead use the *same* examples to learn the test-time augmentation policy and the conformal threshold, exchangeability could be broken.

5. Experimental Set-Up

Datasets We show results on the test splits of three widely used image classification datasets: ImageNet [12] (50,000 natural images spanning 1,000 classes), iNaturalist [41] (100,000 images spanning 10,000 species), and CUB-Birds [45] (5,794 images spanning 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 [17]. 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 [31]. For iNaturalist, we use a model made public by Niers, Tom [30]. 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 its augmentations are likely to be label-preserving. The second, which we refer to as 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. 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 4). The first score is APS [34] (described in Eqn. 4), which represents the cumulative probability required to include the correct class, and the second is RAPS [1], which modifies APS by adding a term to penalize large set sizes. For all experiments, we do not allow sets of size 0. We implement RAPS and APS using code provided by Angelopoulos et al. [1], 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 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. [1]. We first partition examples based upon the size of the prediction set. We create bins for set sizes of $[0, 1]$, $[2, 3]$, $[4, 10]$, $[11, 100]$, 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 [46] for multiple hypothesis testing. Code to reproduce all experiments will be made publicly available.

		Expanded Augmentation Policy			Simple Augmentation 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. **Reductions in prediction set size 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 S8 reports achieved coverage. Corresponding results for APS can be found in Table S2.

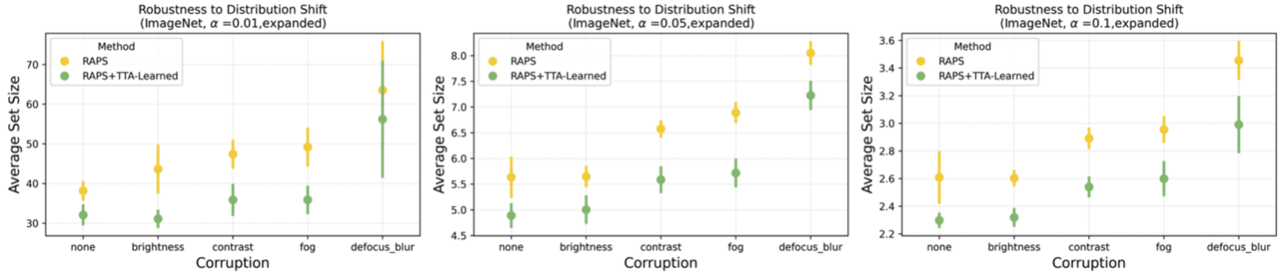

Figure 2. **Robustness to distribution shift.** We compare average prediction set size achieved by RAPS (yellow) to average prediction set size achieved when combining RAPS with TTA-Learned (green). Results reflect the distribution of average prediction set size across 10 runs using ImageNet and ResNet50. We evaluate performance on different corruptions (x-axis) and different coverage guarantees (left, middle, right). RAPS+TTA-Learned (green) produces a noticeable reduction in prediction set size, even when subject to distribution shift, with no loss in coverage. Refer to Figure S1 in the supplement for a comparison of coverage achieved by both methods.

6. Results

We compare against RAPS, which outperformed other baselines in every experiment . (We provide results comparing our method to APS and the Top-K baselines in the supplement. Variants of each experiment across multiple α and datasets are also in the supplement.) We then examine the dependence of these results on dataset, base model, and class. We conclude by providing intuition about why test-time augmentation improves the efficiency of conformal predictors.

6.1. 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 S1). 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.

TTA-Learned performs comparably or better than TTA-Avg in all comparisons. Certain augmentations in the expanded augmentation policy (blur, decrease sharpness, and shear) are consistently assigned a weight of 0, while certain augmentations are consistently included in learned policies (autocontrast, translate). Augmentations assigned zero weight provide no additional information about the true label (for example, they may not preserve the label within the image, or they may be redundant with other augmentations included in the policy).

While TTA improves both RAPS and APS, the improvements are larger for APS. This is because TTA, like RAPS, tempers the predicted probabilities. TTA lowers the maximum predicted probability on average, thereby reducing

		Expanded Augmentation Policy			Simple Augmentation 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. **Reductions in prediction set size across base classifiers on 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 S9.

model overconfidence. Consequently, the predicted probability assigned to the remaining classes is higher. This is why the expanded augmentation policy demonstrates stronger performance than 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 (results can be found in Tables S8 and S9).

We next evaluate adaptivity using size stratified coverage violation (SSCV). At low alpha ($\alpha = .01$, and $\alpha = .05$), TTA-Learned improves efficiency without diminishing adaptivity (Table S10).

6.2. Robustness to distribution shift

Next, we evaluate the performance of test-time augmented conformal prediction on out-of-distribution examples. While conformal prediction does not guarantee coverage in these settings, distribution shifts are ubiquitous in practice [22]. Empirical performance is thus of practical interest. The training procedures for both test-time augmented conformal prediction and conformal prediction remain the same and use in-distribution examples from ImageNet and a ResNet50 classifier. We evaluate each conformal predictor on four types of image corruptions drawn from ImageNet-C [19]. Figure 2 plots the results; across all corruptions and coverage guarantees, test-time augmented conformal prediction produces smaller prediction sets than conformal prediction alone. Importantly, test-time augmented conformal prediction achieves this with *no* loss of coverage (Figure S1).

6.3. Datasets, augmentation policies, and models

Dependence on dataset TTA consistently improves prediction set sizes on ImageNet and iNaturalist, but not on 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 S4).

Dependence on augmentation policy We find that the expanded augmentation policy produces larger reductions in set size than the simple augmentation policy, primarily at low α. This is in spite of the fact that the expanded augmentation policy contains many augmentations outside of the base classifier's training-time augmentation policy. When we vary the number of augmentations included in an augmentation policy, we see that larger augmentation policies also yield greater reductions in average prediction set size (Figure S2). 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 under-performing in 4).

Dependence on base model We tested 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 cover-

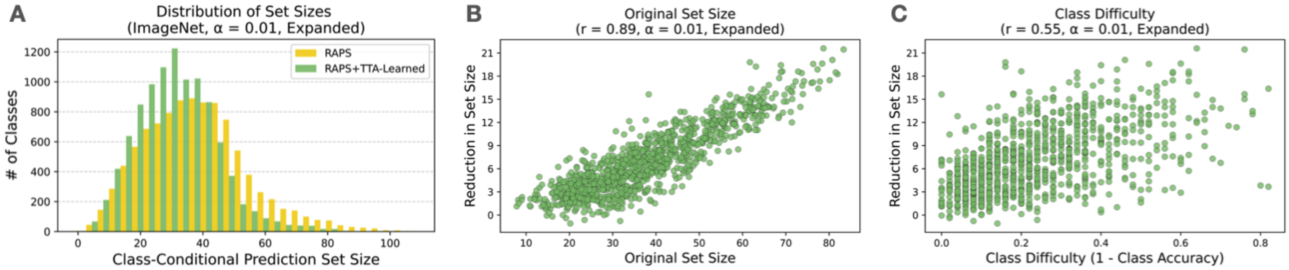

Figure 3. (A) **Class-conditional prediction set sizes.** We plot the distribution of class-conditional prediction set sizes, for ImageNet and ResNet-50 with $\alpha = .01$. RAPS+TTA-Learned (green) produces a noticeable reduction in class-conditional prediction set sizes. (B, C) **Relationship between TTA improvements and original class set sizes and class difficulty**. TTA introduces the largest improvements for classes with the largest original prediction set sizes (B) and classes on which the underlying classifier is often incorrect (C). Each point represents the average prediction set size for each class, across 10 splits.

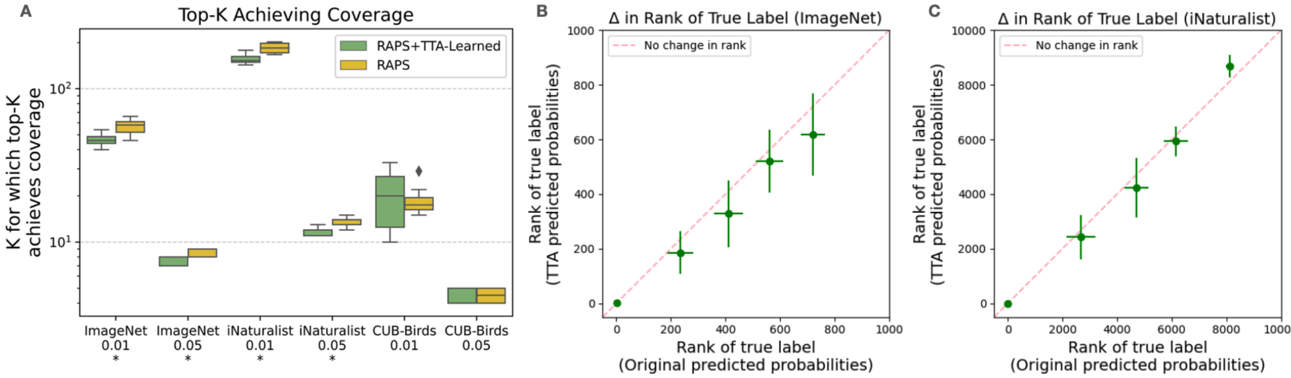

Figure 4. (A) **Effect of TTA-Learned on optimal Top-K**: TTA-Learned significantly lowers the value of k required for Top-k prediction sets to achieve coverage on ImageNet and iNaturalist, but not on CUB-Birds. (B,C) **Effect of TTA-Learned on rank of true class**: TTA-Learned improves the rank of the 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 compared to the TTA-transformed probabilities, binning all examples in the validation set into five equal-width bins. Dots that fall below the red line indicate that TTA improves the rank of the true class.

age. We note 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.

6.4. Class-Specific Analysis

We have established that on average TTA is a useful addition to the conformal pipeline. We now investigate the source of this improvement. We make two empirical observations. First, classes with larger predicted set sizes benefit most from the introduction of TTA. Figure 3 shows that a class's average prediction set size is significantly correlated with the change in set size produced by TTA-Learned (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). Prediction sets for classes that are difficult to predict 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.

6.5. Intuition

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 for those datasets (ImageNet and iNaturalist) where TTA produces significant reductions in set size, TTA-transformed predictions–both with a simple average and learned weights– produce significantly lower values for k compared to the original predictions (Figure 4A). 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 calibration dataset.

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. This finding does not fully explain the value of test-time augmentation to conformal prediction since only 3% of the prediction sets which reduce in size are associated with a corrected Top-1 classification. What TTA did do was increase 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 4 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 that 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).

7. 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 are also limited to image classification. We do not consider other modalities, for which appropriate transformations will substantially differ. Future work should consider how these results may generalize to non-vision tasks. Finally, test-time augmentation is one approach to generating ensembles in conformal prediction. Many other more computationally expensive approaches exist. The tradeoff between computation and ensemble performance remains a useful avenue for future work.

8. Conclusion

We show that test-time augmented conformal prediction produces smaller sets than conformal prediction alone. The proposed 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. Our experiments also indicate that test-time augmented conformal prediction exhibits greater efficiency under four common corruption-based distribution shifts. Test-time augmentation is able to do this by improving the underlying classifier's robustness to domain-specific invariances, in the form of data augmentation. Efforts to improve the efficiency of conformal predictors could, as a first step, aim to improve the robustness of the underlying classifier. The performance of TTA-Learned also suggests that there are settings in which it is wise to use a portion of the labeled data to improve the underlying model, 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.

References

[1] Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I. Jordan. Uncertainty Sets for Image Classifiers using Conformal Prediction, 2022. arXiv:2009.14193 [cs, math, stat]. 2, 4, 11

[2] Murat Seçkin Ayhan and Philipp Berens. Test-time Data Augmentation for Estimation of Heteroscedastic Aleatoric Uncertainty in Deep Neural Networks. 2018. 1, 2

[3] Varun Babbar, Umang Bhatt, and Adrian Weller. On the Utility of Prediction Sets in Human-AI Teams, 2022. arXiv:2205.01411 [cs]. 1

[4] Anthony Bellotti. Optimized conformal classification using gradient descent approximation, 2021. arXiv:2105.11255 [cs]. 2

[5] Lars Carlsson, Martin Eklund, and Ulf Norinder. Aggregated Conformal Prediction. In *Artificial Intelligence Applications and Innovations*, pages 231–240, Berlin, Heidelberg, 2014. Springer. 2

[6] Sewhan Chun, Jae Young Lee, and Junmo Kim. Cyclic test time augmentation with entropy weight method. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 433–442. PMLR, 2022. ISSN: 2640-3498. 2

[7] Gilad Cohen and Raja Giryes. Simple post-training robustness using test time augmentations and random forest. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3996–4006, 2024. 1

[8] Seffi Cohen, Noa Dagan, Nurit Cohen-Inger, Dan Ofer, and Lior Rokach. ICU Survival Prediction Incorporating Test-Time Augmentation to Improve the Accuracy of Ensemble-Based Models. *IEEE Access*, 9:91584–91592, 2021. Conference Name: IEEE Access. 2

[9] Pedro Conde, Tiago Barros, Rui L. Lopes, Cristiano Premebida, and Urbano J. Nunes. Approaching Test Time Augmentation in the Context of Uncertainty Calibration for Deep Neural Networks, 2023. arXiv:2304.05104 [cs]. 2

[10] Jesse C Cresswell, Yi Sui, Bhargava Kumar, and Noël Vouitsis. Conformal prediction sets improve human decision making. *arXiv preprint arXiv:2401.13744*, 2024. 1

[11] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Policies from Data, 2019. arXiv:1805.09501 [cs, stat]. 11

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. ISSN: 1063-6919. 4

[13] Tiffany Ding, Anastasios N. Angelopoulos, Stephen Bates, Michael I. Jordan, and Ryan J. Tibshirani. Class-Conditional Conformal Prediction With Many Classes, 2023. arXiv:2306.09335 [cs, stat]. 2

[14] Bat-Sheva Einbinder, Yaniv Romano, Matteo Sesia, and Yanfei Zhou. Training Uncertainty-Aware Classifiers with Conformalized Deep Learning. 15

[15] Shohei Enomoto, Monikka Roslianna Busto, and Takeharu Eda. Dynamic Test-Time Augmentation via Differentiable Functions, 2023. arXiv:2212.04681 [cs]. 2

[16] A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 148–155, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, 2016. IEEE. 4

[18] Achim Hekler, Titus J. Brinker, and Florian Buettner. 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, 2023. Number: 12. 1, 2

[19] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 6

[20] Vilde Jensen, Filippo Maria Bianchi, and Stian Normann Anfinsen. Ensemble Conformalized Quantile Regression for Probabilistic Time Series Forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–12, 2022. Conference Name: IEEE Transactions on Neural Networks and Learning Systems. 2

[21] Ildoo Kim, Younghoon Kim, and Sungwoong Kim. Learning Loss for Test-Time Augmentation. In *Advances in Neural Information Processing Systems*, pages 4163–4174. Curran Associates, Inc., 2020. 2

[22] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021. 6

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. 11

[24] Arun Kumar Kuchibhotla. Exchangeability, conformal prediction, and rank tests. *arXiv preprint arXiv:2005.06095*, 2020. 4

[25] Henrik Linusson, Ulf Norinder, Henrik Boström, Ulf Johansson, and Tuve Löfström. On the Calibration of Aggregated Conformal Predictors. In *Proceedings of the Sixth Workshop on Conformal and Probabilistic Prediction and Applications*, pages 154–173. PMLR, 2017. ISSN: 2640-3498. 2

[26] H. Linusson, U. Johansson, and H. Boström. Efficient conformal predictor ensembles. *Neurocomputing*, 397:266–278, 2020. 2

[27] Helen Lu, Divya Shanmugam, Harini Suresh, and John Guttag. Improved Text Classification via Test-Time Augmentation, 2022. arXiv:2206.13607 [cs]. 2

[28] Alexander Lyzhov, Yuliya Molchanova, Arsenii Ashukha, Dmitry Molchanov, and Dmitry Vetrov. Greedy Policy Search: A Simple Baseline for Learnable Test-Time Augmentation. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 1308–1317. PMLR, 2020. ISSN: 2640-3498. 2

[29] Luca Mocerino, Roberto G. Rizzo, Valentino Peluso, Andrea Calimera, and Enrico Macii. Adaptive Test-Time Augmentation for Low-Power CPU, 2021. arXiv:2105.06183 [cs, eess]. 2

[30] Niers, Tom. iNaturalist_competition, 2021. original-date: 2021-12-10T10:56:46Z. 4

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 4

[32] Juan C. Perez, Motasem Alfarra, Guillaume Jeanneret, Laura Rueda, Ali Thabet, Bernard Ghanem, and Pablo Arbelaez. Enhancing Adversarial Robustness via Test-time Transformation Ensembling. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 81–91, Montreal, BC, Canada, 2021. IEEE. 1, 2

[33] Drew Prinster, Anqi Liu, and Suchi Saria. JAWS: Auditing Predictive Uncertainty Under Covariate Shift, 2022. arXiv:2207.10716 [cs, stat]. 2

[34] Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. Classification with Valid and Adaptive Coverage, 2020. arXiv:2006.02544 [stat]. 2, 3, 4

[35] C Saunders and Royal Holloway. Transduction with Confidence and Credibility. 1999. 2

[36] Glenn Shafer and Vladimir Vovk. A Tutorial on Conformal Prediction. 2008. 1, 3

[37] Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag. Better Aggregation in Test-Time Augmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1194–1203, Montreal, QC, Canada, 2021. IEEE. 1, 2

[38] David Stutz, Krishnamurthy, Dvijotham, Ali Taylan Cemgil, and Arnaud Doucet. Learning Optimal Conformal Classifiers, 2022. arXiv:2110.09192 [cs, stat]. 2

[39] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[40] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal Prediction Under Covariate Shift. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 2

[41] Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. Benchmarking Representation Learning for Natural World Image Collections, 2021. arXiv:2103.16483 [cs]. 4

[42] Vladmir Vovk. *Algorithmic Learning in a Random World*. Springer-Verlag, New York, 2005. 2

[43] Vladimir Vovk. Cross-conformal predictors, 2012. arXiv:1208.0806 [cs, stat]. 2

[44] Vladimir Vovk. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74(1-2):9–28, 2015. 2

[45] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 4

[46] Eric W Weisstein. Bonferroni correction. *https://mathworld. wolfram. com/*, 2004. 4

[47] Dongping Zhang, Angelos Chatzimparmpas, Negar Kamali, and Jessica Hullman. Evaluating the utility of conformal prediction sets for ai-advised image labeling. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2024. 1

[48] Marvin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test Time Robustness via Adaptation and Augmentation, 2022. arXiv:2110.09506 [cs]. 2

Supplementary Material

S1. Experimental Details

S1.1. 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 [23]. 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 [11]. 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.

S1.2. Learning aggregation function

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 our 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. We train all models using a machine equipped with 4 Titan Xp GPUs, 2 Octa Intel Xeon E5-2620 CPUs, and 1TB of RAM.

S2. Supplementary Results

S2.1. Test-Time Augmentation and APS

TTA-Learned combined with the expanded augmentation policy produces the smallest set sizes when combined with APS, across the datasets considered (Table S1) and each base classifier (Table S3). 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. [1] 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.

S2.2. Comparison to Top-1 and Top-5

We expand Table 1 to include the Top-1 and Top-5 baselines in Table S6. Unsurprisingly, neither outperform RAPS, and consequently none outperform the combination of RAPS, TTA-Learned, and the expanded augmentation policy.

S2.3. Comparison to minimizing 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. Table S7 reports our results. 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.

S2.4. Impact on coverage

We provide exact values of coverage for the main experiments 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). Coverage values for the RAPS experiment across coverage values and datasets can be found in Table S8 and coverage values for the RAPS experiment across base classifiers can be found in Table S9. Similarly, we provide coverage values for the APS experiment across datasets (Table S2) and across models (Table S3).

S2.5. Impact of different coverage guarantees and datasets

We replicate the class-specific analysis for ImageNet at a value of $\alpha = .05$ (Figure S5), iNaturalist (Figure S6), and CUB-Birds (Figure S7). 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 *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.

S2.6. Impact of augmentation policy size

We also analyze the impact of augmentation policy size on average prediction set size for CUB-Birds (Figure S2), 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 $\alpha = .01$.

S2.7. 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 S3. 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 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

S2.8. Impact of calibration set size

We plot the relationship between calibration set size and average prediction set size in Figure S4 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.

S2.9. Impact of different backbone architecture

Our results in the main text are limited to a single architecture (residual networks). Here, we provide evidence of generalizability to different architectures by replicating our ImageNet results using MobileNetV2, across a range of coverage guarantees and both augmentation policies (Table S11) and find consistent results, which support the versatility of the proposed method.

		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 S1. 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 S2. Coverage values associated with experiments in Table S1. 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 $\alpha = .05$.

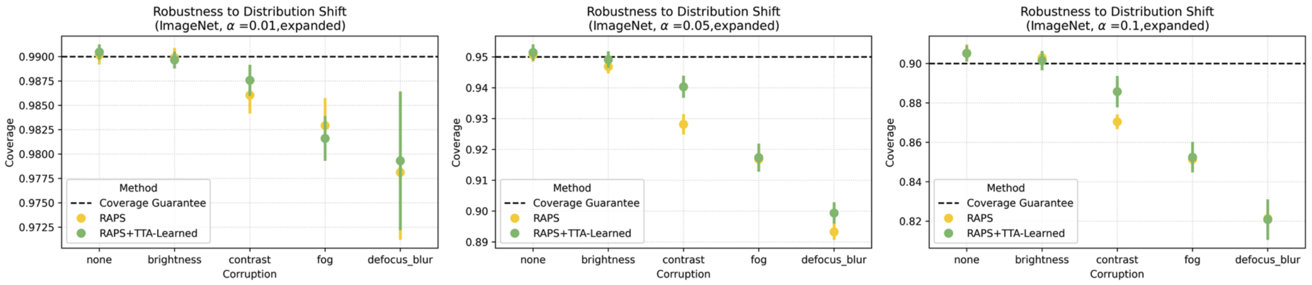

Figure S1. **Impact on coverage.** We plot achieved coverage for both RAPS and RAPS+TTA-Learned across several coverage guarantees and distribution shifts. As expected, distribution shift leads conformal predictors to not meet the coverage guarantee. In each case, the addition of TTA does not worsen coverage; in some cases (for example, given the contrast corruption and a coverage guarantee of 0.05) it even improves coverage.

		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 S3. 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 S4. 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.

	Expanded Aug Policy			Simple Aug Policy		
Method	ResNet50	ResNet101	ResNet152	ResNet50	ResNet101	ResNet152
Original	0.761 ± 0.002	0.773 ± 0.001	0.783 ± 0.002	0.761 ± 0.002	0.773 ± 0.001	0.783 ± 0.002
TTA-Avg	0.764 ± 0.002	0.778 ± 0.001	0.788 ± 0.002	0.77 ± 0.002	0.783 ± 0.001	0.792 ± 0.002
TTA-Learned	0.771 ± 0.002	0.785 ± 0.001	0.793 ± 0.002	0.771 ± 0.002	0.784 ± 0.001	0.793 ± 0.002

Table S5. **TTA effect on classifier performance.** We report differences in classifier performance using a learned test-time augmentation policy compared to a simple average (TTA-Avg) and no test-time augmentation (Original). TTA-Learned offers small improvements over a simpler average and the original model across architectures. FILL IN THE REST, explain how TTA's improvement to Top-1 accuracy alone is small, and does not fully explain the value of test-time augmentation to conformal prediction.

		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 S6. **Comparison to Top-1 and Top-5 baselines.** 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	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 S7. **Alternate training objectives.** 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 [14]. 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.

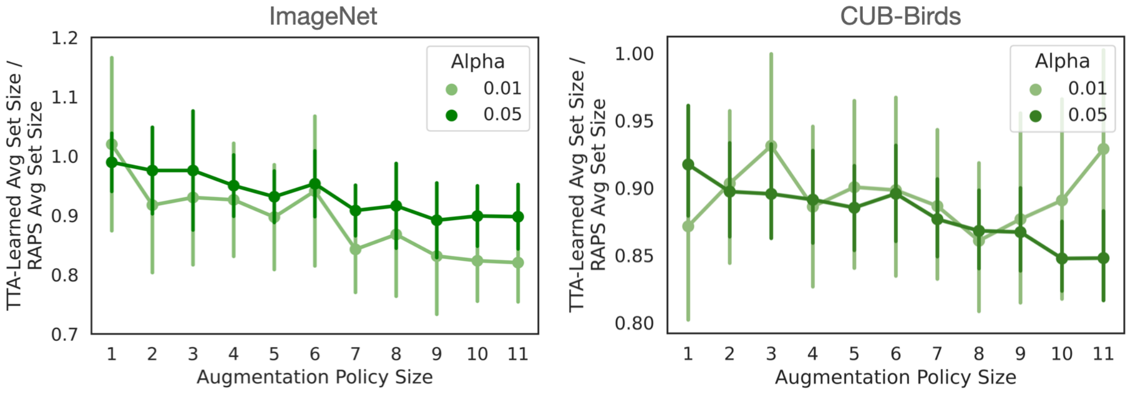

Figure S2. **Impact of augmentation policy size.** We see that larger policy sizes translate to a greater improvement (in terms of the ratio of average prediction set sizes using RAPS+TTA-Learned to average prediction set sizes using RAPS alone) for $\alpha = .05$. For $\alpha = .01$, there is no clear trend.

		Expanded Aug Policy			Simple Aug Policy		
Alpha	Method	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 S8. **Comparison of achieved coverage.** 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).

		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 S9. **Comparison of coverage across base classifiers.** 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).

Alpha	Method	ImageNet	iNaturalist	CUB-Birds
0.01	RAPS	0.0112 ± 0.0043	0.0207 ± 0.0043	0.0076 ± 0.0031
0.01	RAPS+TTA-Learned	0.0113 ± 0.0067	0.0247 ± 0.0027	0.0046 ± 0.0026
0.05	RAPS	0.2134 ± 0.0348	0.0609 ± 0.0217	0.0112 ± 0.0105
0.05	RAPS+TTA-Learned	0.3338 ± 0.0994	0.0899 ± 0.0520	0.0350 ± 0.0412
0.10	RAPS	0.1318 ± 0.0696	0.0852 ± 0.0151	0.2218 ± 0.1260
0.10	RAPS+TTA-Learned	0.3198 ± 0.0977	0.1008 ± 0.0058	0.1931 ± 0.1208

Table S10. **Effect of test-time augmented conformal prediction on adaptivity.** We show results in the context of ResNet-50 and RAPS, across several coverage guarantees. We compute size-stratified coverage violation (SSCV) for each run as described in Sec. 5, and report the mean and standard deviation of SSCV across runs here. Test-time augmentation does not significantly diminish adaptivity at each coverage guarantee considered (assessed via a two-sample t-test, $p > 0.05$).

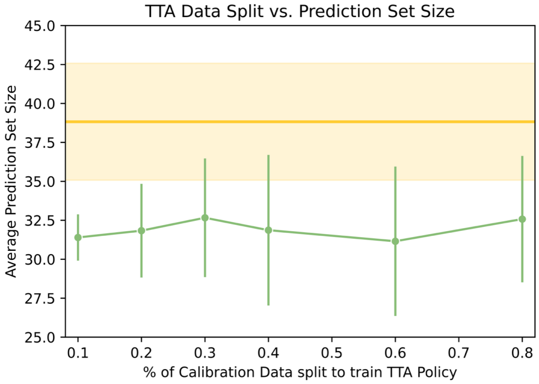

Figure S3. **Robustness to size of dataset used to train test-time augmentation policy.** We plot the percentage of data used to train the TTA policy on the x-axis and the average prediction set size on the y-axis. Error bars describe variance over 10 random splits of the calibration and test set. We 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.

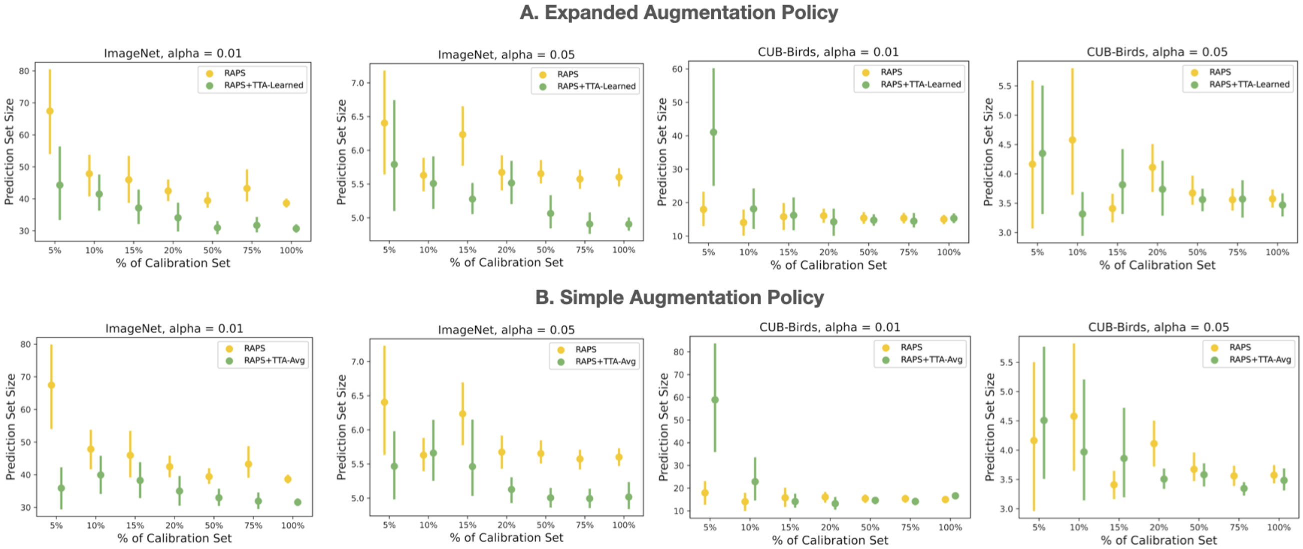

Figure S4. **Impact of calibration set size.** We plot the relationship between calibration set size and average prediction set size across two values of alpha, two augmentation policies, and two datasets (ImageNet and CUB-Birds). For ImageNet, larger calibration set sizes correlate with larger and more 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.

α	Method	ImageNet (Expanded)	ImageNet (Simple)
0.01	RAPS	52.332 ± 8.970	52.332 ± 8.970
0.01	RAPS+TTA-Avg	45.604 ± 1.515	42.431 ± 1.516
0.01	RAPS+TTA-Learned	40.872 ± 1.377	40.843 ± 1.707
0.05	RAPS	8.872 ± 0.417	8.872 ± 0.417
0.05	RAPS+TTA-Avg	8.304 ± 0.322	7.945 ± 0.861
0.05	RAPS+TTA-Learned	7.723 ± 0.916	7.609 ± 1.027
0.10	RAPS	3.677 ± 0.104	3.677 ± 0.104
0.10	RAPS+TTA-Avg	3.480 ± 0.056	3.298 ± 0.069
0.10	RAPS+TTA-Learned	3.321 ± 0.289	3.348 ± 0.275

Table S11. **Replicated results on MobileNetV2.** We observe trends similar to those reported to in the main text in the context of MobileNetV2. In short, RAPS combined with a learned test-time augmentation policy (RAPS+TTA-Learned) produces the smallest set sizes across the considered coverage guarantees ($\alpha \in \{0.01, 0.05, 0.10\}$) and augmentation policies.

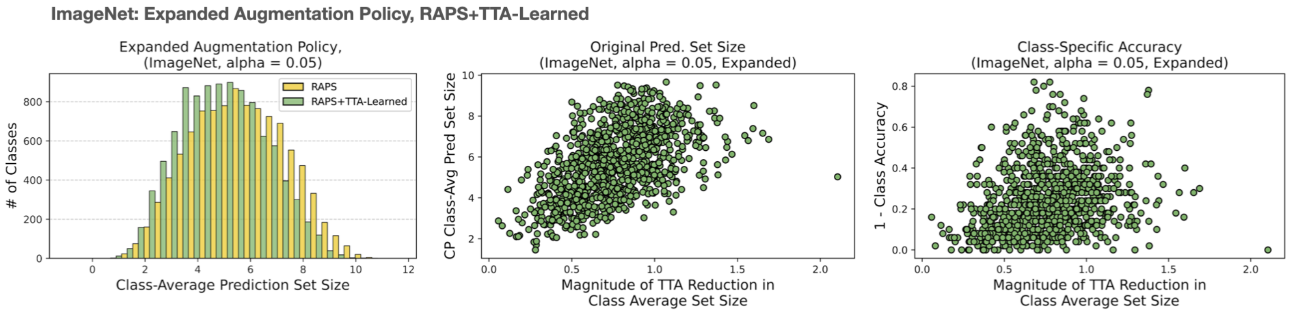

Figure S5. **Class-specific performance for ImageNet**, for a coverage of 95% $\alpha = .05$. Using the expanded augmentation policy RAPS+TTA-Learned produces a noticeable shift in class-average prediction set sizes to the left. There is a significant correlation between original prediction set size and improvements from TTA (middle) and between class difficulty and improvements from TTA (right).

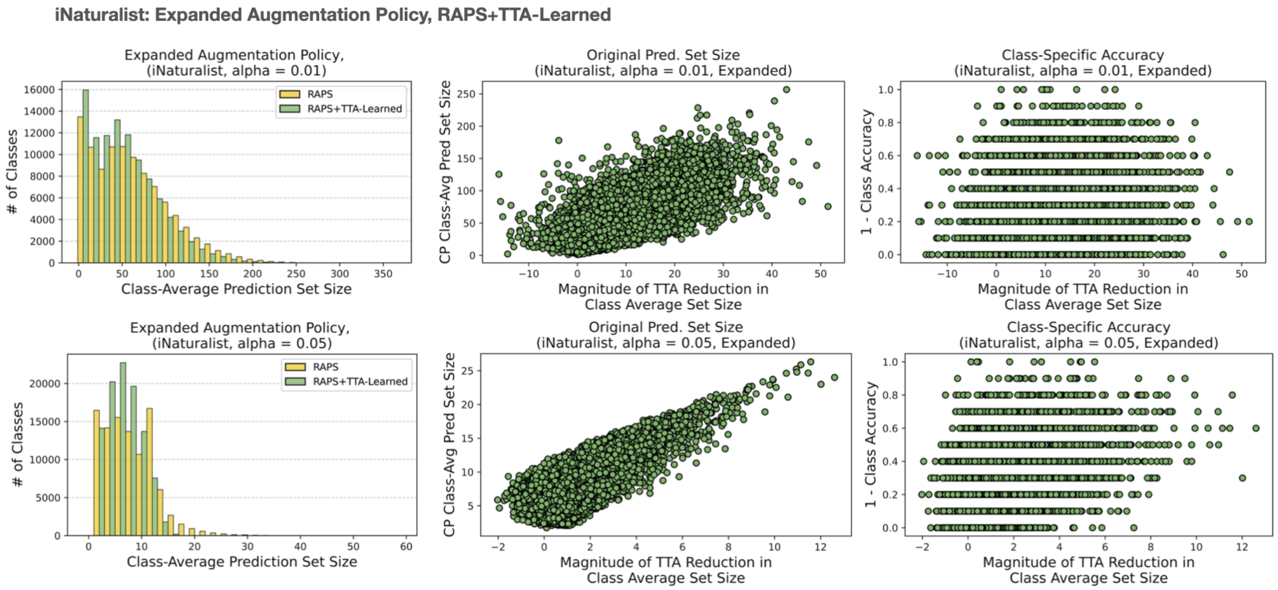

Figure S6. **Class-specific performance for iNaturalist**, for $\alpha = .01$ (top) and $\alpha = .05$ (bottom). We see a consistent relationship between TTA improvements and original class-average prediction set size (middle) and class difficulty (right). Estimates of class-specific accuracy on iNaturalist are quite noisy because there are 10 images per class (which produces distinct accuracy bands).

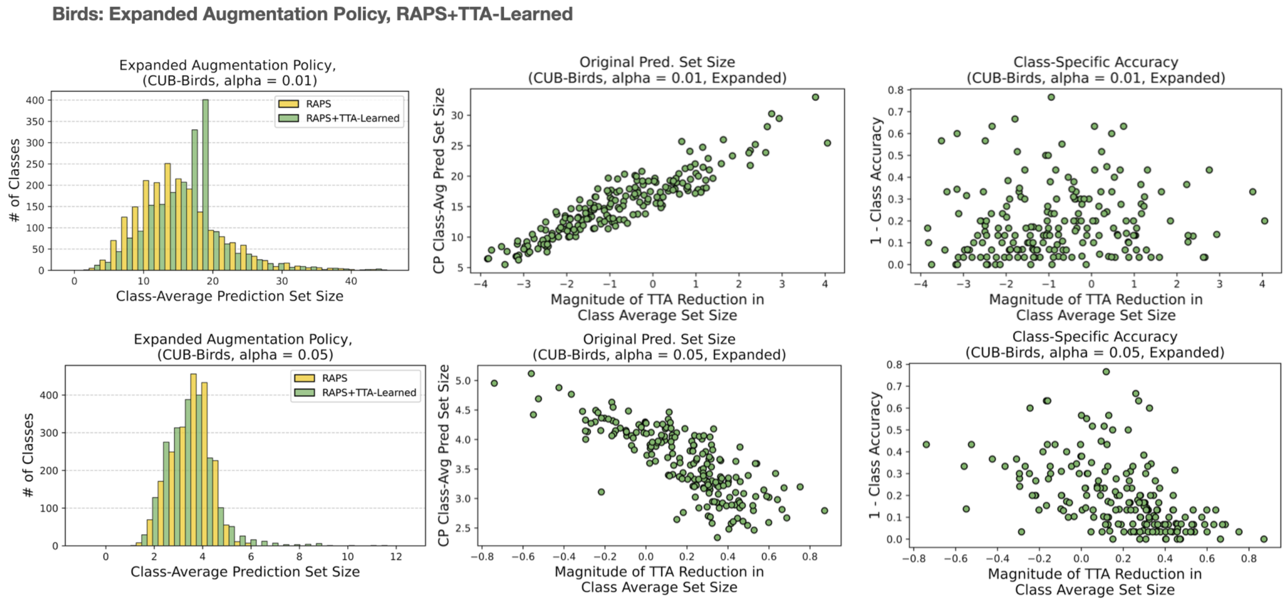

Figure S7. **Class-specific performance for CUB-Birds**, for $\alpha = .01$ (top) and $\alpha = .05\%$ (bottom). These graphs show an example for which TTA-Learned does *not* produce improvements in average prediction set size (computed across all examples). Interestingly, behavior on a class-specific 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.