Robust Yet Efficient Conformal Prediction Sets

Soroush H. Zargarbashi [1] **Mohammad Sadegh Akhondzadeh** [2] **Aleksandar Bojchevski** [2]

Abstract

Conformal prediction (CP) can convert any model's output into prediction sets guaranteed to include the true label with any user-specified probability. However, same as the model itself, CP is vulnerable to adversarial test examples (evasion) and perturbed calibration data (poisoning). We derive provably robust sets by bounding the worst-case change in conformity scores. Our tighter bounds lead to more efficient sets. We cover both continuous and discrete (sparse) data and our guarantees work both for evasion and poisoning attacks (on both features and labels).

1. Introduction

Uncertainty quantification (UQ) is crucial for deploying models, especially in safety-critical domains. The predicted probability is not a reliable source for UQ as it is often uncalibrated (Guo et al., 2017). Most methods do not provide any guarantees and require retraining or modifications in the model architecture (Abdar et al., 2021). Instead, conformal prediction (CP) returns prediction *sets* with a distribution-free guarantee to cover the true label. It only requires black-box access to the model and assumes exchangeable data (a weaker assumption than i.i.d.). This makes CP flexible – we can apply it to image classification, segmentation (Angelopoulos et al., 2023), question answering (Angelopoulos et al., 2022), and node classification (Huang et al., 2023).

Most models suffer a significant performance drop when fed noisy or manipulated data, even for indistinguishable (label-preserving) perturbations (Silva & Najafirad, 2020). Adversaries can exploit this vulnerability by perturbing the training data (poisoning) or the test data (evasion). CP's performance is also sensitive to the same attacks. One goal of the adversary is to break the guarantee – reducing the probability to cover the true label by perturbing the test inputs (evasion) or poisoning the calibration data. In all settings, the perturbations are limited according to a threat model, e.g. a ball of a given radius around the clean input (see § 2). Unlike heuristic defenses which are easily overcome by new attacks (Athalye et al., 2018; Mujkanovic et al., 2022), certificates provide worst-case guarantees that the prediction does not change. How can we extend robustness certificates to conformal prediction sets?

Given calibration data and a score function $s : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ capturing conformity (agreement) between inputs and all potential labels, CP finds a calibrated threshold q_α, and defines prediction sets $\mathcal{C}_\alpha(\boldsymbol{x}) = \{y : s(\boldsymbol{x}, y) \geq q_\alpha\}$ that include all labels with scores above it. CP guarantees that $\Pr\left[y_{\text{true}} \in \mathcal{C}_\alpha(\boldsymbol{x})\right] \geq 1 - \alpha$ for a clean $\boldsymbol{x}$, exchangeable with the calibration data, and any user-specfied α. To certify robustness, we can define *conservative* sets that ensure the coverage remains above $1 - \alpha$ even under perturbation.

To this end, Gendler et al. (2021) leverage the fact that the randomly smoothed scores $\mathbb{E}_{\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})}[s(\boldsymbol{x} + \boldsymbol{\delta}, y)]$ change slowly around the input to compute an upper bound on the worst-case score. Their randomly smoothed conformal prediction (RSCP) method has 4 limitations: (i) It considers only the mean of randomized scores resulting in a looser bound and thus larger sets; (ii) It only certifies evasion but not poisoning attacks; (iii) It only supports L_2-bounded perturbations of continuous data, ignoring discrete and sparse data such as graphs; (iv) It does not correct for finite-sample approximation errors. We address all of these limitations.

Our key insight is that we can use the cumulative distribution (CDF) of smooth scores to obtain tighter upper bounds. The resulting CDF-aware sets are smaller while maintaining the same robustness guarantee. For continuous data we reuse Kumar et al. (2020)'s bound developed to certify confidence, while for discrete/graph data we extend the bounds of Bojchevski et al. (2020).[1] We then propose an approach for finite sample correction. Different from Yan et al. (2024), we bound calibration points instead of test points. In addition to being significantly faster (especially for large datasets like ImageNet), our calibration-time algorithm also leads to smaller sets when correcting for finite samples.

[1]CISPA Helmholtz Center for Information Security [2]University of Cologne. Correspondence to: Soroush H. Zargarbashi <zargarbashi@cs.uni-koeln.de>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]Both of these methods do not provide sets or CP guarantees.

Currently, there are no CP methods designed to handle poisoning. To fill this gap, we further derive provably robust sets that maintain worst-case coverage when either the features or the labels of the calibration set can be perturbed. Moreover, the poisoning guarantee is independent of how the bound on conformity scores is derived. Hence, our poisoning-aware and evasion-aware methods can be combined to provide robustness to both attacks simultaneously.

In short, we introduce CDF-Aware smoothed prediction Sets (CAS) that provably cover the true label under adversarial attacks. For evasion, we show a consistent improvement on all metrics and datasets compared to RSCP. Moreover, for the first time, we additionally provide guarantees for poisoning, as well as discrete and sparse data.

2. Background

Conformal prediction. Given a holdout calibration set $\mathcal{D}_{\text{cal}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ exchangeably sampled from the data distribution (or a finite dataset) with labels unseen by the model (during training), and a user-specified coverage probability $1-\alpha$, for any test point $\boldsymbol{x}_{n+1}$, CP defines a prediction set $\mathcal{C}_\alpha(\boldsymbol{x}_{n+1}) \subseteq \mathcal{Y}$ that is guaranteed to cover the true label y_{n+1} with the predetermined probability.

Theorem 2.1 (Vovk et al. (2005))**.** *If $\mathcal{D}_{\text{cal}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, and $(\boldsymbol{x}_{n+1}, y_{n+1})$ are exchangeable, for any continuous score function $s : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ capturing the agreement between $\boldsymbol{x}$, and y, and user-specified $\alpha \in (0, 1)$, the prediction set defined as $\mathcal{C}_\alpha(\boldsymbol{x}_{n+1}) = \{y : \ s(\boldsymbol{x}_{n+1}, y) \geq q_\alpha\}$ has coverage probability*

$$\Pr\left[y_{n+1} \in \mathcal{C}_\alpha(\boldsymbol{x}_{n+1})\right] \geq 1 - \alpha \tag{1}$$

where $q_\alpha := \text{Quant}\left(\alpha; \{s(\boldsymbol{x}_i, y_i)\}_{i=1}^n\right)$ is the α-quantile of the true scores in the calibration set.

This theorem was extended to graphs (Zargarbashi et al., 2023; Huang et al., 2023) showing that the same guarantee holds for node classification. Although the coverage is guaranteed regardless of the choice of score function, a good choice is reflected in the size of the prediction sets (also called efficiency), the proportion of singleton sets covering the true label, and other metrics. A simple score function known as threshold prediction sets (TPS) directly considers the model's output $s(\boldsymbol{x}, y) = \pi(\boldsymbol{x}, y)$ where π are the class probability (softmax) estimates (Sadinle et al., 2018). TPS tends to over-cover easy examples and under-cover hard ones (Angelopoulos & Bates, 2021). This is remedied by the commonly used adaptive prediction sets (APS) score defined as $s(\boldsymbol{x}, y) := -\left(\rho(\boldsymbol{x}, y) + u \cdot \pi(\boldsymbol{x})_y\right)$. Here $\rho(\boldsymbol{x}, y) := \sum_{c=1}^K \pi(\boldsymbol{x})_c 1\left[\pi(\boldsymbol{x})_c > \pi(\boldsymbol{x})_y\right]$ is the sum of all classes predicted as more likely than y, and $u \in [0, 1]$ is a uniform random value that breaks the ties between different scores to allow exact $1 - \alpha$ coverage (Romano et al., 2020). While we report our results on both scoring functions, our approach is orthogonal and hence applicable to any other choice (see § A for an extended introduction to CP).

Adversarial attacks. We define the threat model – the set of all possible perturbations the adversary can apply – by a ball centered around a clean input $\boldsymbol{x}$. For continuous $\boldsymbol{x}$ we consider the l_2 ball of radius r around the input $\mathcal{B}_r(\boldsymbol{x}) = \{\tilde{\boldsymbol{x}} \in \mathcal{X} : ||\tilde{\boldsymbol{x}} - \boldsymbol{x}||_2 \leq r\}$. For binary data, we define the ball w.r.t. the number of flipped bits: $\mathcal{B}_{r_a, r_d}(\boldsymbol{x}) = \{\tilde{\boldsymbol{x}} \in \mathcal{X} : \sum_{i=1}^d \mathbf{1}[\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - 1] \leq r_d, \sum_{i=1}^d \mathbf{1}[\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + 1] \leq r_a\}$ where r_d and r_a are the numbers of deleted and added bits respectively. This distinction accounts for sparsity as shown by Bojchevski et al. (2020). We discuss categorical data in § C, extensions to other threat models are simple.

Evasion attacks. For a given input $\boldsymbol{x}$ and the model f, the adversary's usual goal is to find a perturbed input $\tilde{\boldsymbol{x}}$ such that $f(\tilde{\boldsymbol{x}}) \neq f(\boldsymbol{x})$ (Yuan et al., 2019; Madry et al., 2017). In CP, the goal changes to excluding the true label from the prediction set $\mathcal{C}_\alpha(\tilde{\boldsymbol{x}})$ which breaks the guarantee in Eq. 1. Here we assume that CP is calibrated with clean calibration points.

Poisoning attacks. The adversary can perturb the training data to e.g. decrease accuracy. However, since CP is model-agnostic, the guarantee holds regardless of the model's accuracy. Instead, here the goal of the adversary is to perturb the *calibration* set in order to decrease the empirical coverage – breaking the guarantee (see formal definition in § 3.2).

3. Robust Prediction Sets

3.1. Robustness to Evasion Attacks

Definition 3.1 (Robust coverage)**.** The prediction sets $\mathcal{C}_\alpha(\cdot)$ have adversarially robust $1 - \alpha$ coverage if for any $(\boldsymbol{x}_{n+1}, y_{n+1})$ exchangeable with $\mathcal{D}_{\text{cal}}$

$$\Pr\left[y_{n+1} \in \mathcal{C}_\alpha(\tilde{\boldsymbol{x}}_{n+1}) \mid \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})\right] \geq 1 - \alpha \tag{2}$$

where $\mathcal{B}(\boldsymbol{x})$ can be the l_2 ball $\mathcal{B}_r(\boldsymbol{x})$, the binary ball $\mathcal{B}_{r_a, r_d}$, or any other threat model. Gendler et al. (2021) define a score $s_{\text{rscp}}(\boldsymbol{x}, y) = \Phi^{-1}(\mathbb{E}_{\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})}[s(\boldsymbol{x} + \boldsymbol{\delta}, y)])$ based on Gaussian smoothing (Cohen et al., 2019) where $\Phi^{-1}(\cdot)$ is the inverse CDF of $\mathcal{N}(0, 1)$. Since the smooth score is bounded, $s_{\text{rscp}}(\tilde{\boldsymbol{x}}, y) \leq s_{\text{rscp}}(\boldsymbol{x}, y) + \frac{r}{\sigma}, \forall \tilde{\boldsymbol{x}} \in \mathcal{B}_r(\boldsymbol{x})$ they shift the quantile $\underline{q_\alpha} = q_\alpha - \frac{r}{\sigma}$ to ensure robustness. Instead of shifting the quantile we directly bound the conformal scores which is a slight generalization.

Proposition 3.1. *Define $\overline{s}(\boldsymbol{x}, y)$ as the upper bound for $\{s(\tilde{\boldsymbol{x}}, y) : \tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})\}$. With q_α as the α-quantile of the true (clean) calibration scores, let $\overline{\mathcal{C}}_\alpha(\boldsymbol{x}) = \{y : \overline{s}(\boldsymbol{x}, y) \geq q_\alpha\}$. For all $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$, if $(\boldsymbol{x}_{n+1}, y_{n+1})$ is exchangeable with $\mathcal{D}_{\text{cal}}$ then we have $\Pr\left[y_{n+1} \in \overline{\mathcal{C}}_\alpha(\tilde{\boldsymbol{x}}_{n+1})\right] \geq 1 - \alpha$.*

All omitted proofs are in § D.1. We summarize our notation

in § K. In short, the conservative set for any $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$ includes the labels of the vanilla prediction set for $\boldsymbol{x}$. Thus, the coverage guarantee also applies for the perturbed points.

RSCP is a special case with $\overline{s}(\boldsymbol{x}, y) = s_{\text{rscp}}(\boldsymbol{x}, y) + \frac{r}{\sigma}$. We can equivalently rewrite RSCP as an upper bound on $\mathbb{E}[s(\cdot,\cdot)]$ instead of $\Phi^{-1}(\mathbb{E}[s(\cdot,\cdot)])$ which matches the bound from Kumar et al. (2020) (see § F.1). In § 4 we significantly improve the bound using the CDF. Tighter bounds result in smaller (more efficient) sets.

3.2. Robustness to Feature Poisoning Attacks

We assume that the adversary can modify at most k instances, $0 \leq k \leq n = |\mathcal{D}_{\text{cal}}|$, whose features can be perturbed in a (continuous or discrete) ball $\mathcal{B}$ around the clean features. We define the threat model at dataset-level:

$$\mathbb{B}_{k,\mathcal{B}}(\mathcal{D}) = \{\tilde{\mathcal{D}} : \tilde{\mathcal{D}} = \{(\tilde{\boldsymbol{x}}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}, \tilde{\boldsymbol{x}}_i \in \mathcal{B}(\boldsymbol{x}_i), \sum_{j=1}^{n} \mathbf{1}[\tilde{\boldsymbol{x}}_j \neq \boldsymbol{x}_j] \leq k\}\}$$

Let q_α be the α-quantile of the clean calibration scores. To decrease coverage the adversary aims to find a perturbed calibration set $\tilde{\mathcal{D}}_{\text{cal}} \in \mathbb{B}_{k,\mathcal{B}}(\mathcal{D}_{\text{cal}})$ that moves the quantile $\tilde{q}_\alpha = \text{Quant}(\alpha; \tilde{\mathcal{D}}_{\text{cal}})$ as right as possible compared to q_α.[2] This shift increases the probability of rejecting true labels, resulting in a lower coverage. Namely, for $\tilde{\alpha} = \text{Quant}^{-1}(\tilde{q}_\alpha; \mathcal{D}_{\text{cal}})$, the quantile inverse of the poisoned threshold $\tilde{q}$ w.r.t. the clean calibration set, the poisoned calibration set results in near $1 - \tilde{\alpha}$ coverage where by definition $1 - \tilde{\alpha} \leq 1 - \alpha$. Given a potentially poisoned calibration set $\tilde{\mathcal{D}}_{\text{cal}}$ we certify the prediction sets via the following optimization problem:

$$\begin{aligned} \underline{q_\alpha} = \min_{\boldsymbol{z}_i \in \mathcal{X}} \ & \text{Quant}\left(\alpha; \{s(\boldsymbol{z}_i, y_i)\}_{i=1}^n\right) \\ \text{s.t.} \quad & \forall (\tilde{\boldsymbol{x}}_i, y_i) \in \tilde{\mathcal{D}}_{\text{cal}} : \boldsymbol{z}_i \in \mathcal{B}(\tilde{\boldsymbol{x}}_i) \\ & \sum_{i \leq n} \mathbf{1}[\boldsymbol{z}_i \neq \tilde{\boldsymbol{x}}_i] \leq k \end{aligned} \tag{3}$$

The problem in Eq. 3 finds the most conservative quantile $\underline{q_\alpha}$ and it holds that $\underline{q_\alpha} \leq q_\alpha$ since for any perturbed $\tilde{\mathcal{D}}_{\text{cal}}$ by definition it holds $\mathcal{D}_{\text{cal}} \in \mathbb{B}_{k,\mathcal{B}}(\tilde{\mathcal{D}}_{\text{cal}})$. We show that the minimizer of problem Eq. 3 certifies *at least* $1 - \alpha$ coverage.

Proposition 3.2. *Let $\underline{q_\alpha}$ to be the solution to the optimization problem in Eq. 3. With the conservative prediction sets*

$$\overline{\mathcal{C}}_\alpha(\boldsymbol{x}_{n+1}) = \left\{y_i : s(\boldsymbol{x}_{n+1}, y_i) \geq \underline{q_\alpha}\right\} \tag{4}$$

for any $(\boldsymbol{x}_{n+1}, y_{n+1})$ exchangeable with (clean) $\mathcal{D}_{\text{cal}}$ we have $\Pr\left[y_{n+1} \in \overline{\mathcal{C}}_\alpha(\boldsymbol{x}_{n+1})\right] \geq 1 - \alpha$.

[2] Our setup works with conformity score capturing the agreement between $\boldsymbol{x}$ and y. With a non-conformity score, the goal is to equivalently shift the quantile to the left (see § A).

With access to lower and upper bounds on the adversarial scores we can change the constraint $\boldsymbol{z}_i \in \mathcal{B}(\tilde{\boldsymbol{x}}_i)$ in Eq. 3 to $z_i \in [\underline{s}(\tilde{\boldsymbol{x}}_i, y_i), \overline{s}(\tilde{\boldsymbol{x}}_i, y_i)]$ where $z_i \in \mathbb{R}$ is a scalar variable, and solve the relaxed problem. We describe in § 4 how to obtain such bounds using randomized smoothing which we can use in both Prop. 3.1 and Prop. 3.2. Regardless of how we solve Eq. 3, as long as it finds a $\underline{q_\alpha} \leq q_\alpha$ conditional on the clean $\mathcal{D}_{\text{cal}}$ the guarantee holds.

3.3. Robustness to Label Poisoning Attacks

In the label poisoning setup, the adversary can flip the labels of at most k datapoints in the calibration set, again aiming to shift the quantile to the right. As before, we can find the most conservative quantile by solving the problem:

$$\begin{aligned} \underline{q_\alpha} = \min_{z_i \in \mathcal{Y}} \ & \text{Quant}\left(\alpha; \left\{s(\boldsymbol{x}_i, z_i) : (\boldsymbol{x}_i, \tilde{y}_i) \in \tilde{\mathcal{D}}_{\text{cal}}\right\}\right) \\ \text{s.t.} \quad & \sum_{i \leq n} \mathbf{1}[z_i \neq \tilde{y}_i] \leq k \end{aligned} \tag{5}$$

Similar to § 3.2, since $\underline{q_\alpha} \leq q_\alpha$, prediction sets defined as in Eq. 4 maintain $\geq 1 - \alpha$ coverage even under worst-case label perturbation. We can solve both problems (Eq. 3 and Eq. 5) by writing them as mixed-interger linear programs (MILPs). We present the technical details in § G.

Interestingly, our evasion-aware sets can easily be combined with our poisoning-aware threshold to obtain prediction sets that are robust to both types of attacks. Similarly, we can easily combine the feature and label poisoning constraints in a single problem. We discuss these extensions in § H.

4. Randomized Smoothing Bounds

To instantiate the conservative sets $\overline{\mathcal{C}}_\alpha(\cdot)$ defined in § 3 we need bounds on the worst-case change in conformity scores under perturbation. There is a rich literature on robustness certificates for standard classification (Li et al., 2023) that we can lean on, since they often need to compute similar bounds as a byproduct. We focus on methods based on the randomized smoothing framework (Cohen et al., 2019) given their high flexibility and black-box nature. This couples well with the flexibility of CP, ensuring that our final robust CP method can be broadly applied.

Smooth scores. A smoothing scheme $\xi : \mathcal{X} \mapsto \mathcal{X}$ is a function that maps the input $\boldsymbol{x}$ to a nearby random point. Given an arbitrary score $s(\cdot,\cdot)$, we compute the expected (smooth) conformal scores as $\hat{s}(\boldsymbol{x}, y) := \mathbb{E}[s(\xi(\boldsymbol{x}), y)]$. Following Cohen et al. (2019) for Gaussian smoothing, we add isotropic noise where the scale σ^2 determines the amount of smoothing $\hat{s}(\boldsymbol{x}, y) = \mathbb{E}_{\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})}[s(\boldsymbol{x} + \boldsymbol{\delta}, y)]$. For binary data, we use sparse smoothing (Bojchevski et al., 2020) and flip zeros and ones with probabilities p_0 and p_1 respectively: $\hat{s}(\boldsymbol{x}, y) = \mathbb{E}[s(\boldsymbol{x} \oplus \boldsymbol{\delta}, y)]$, where $\oplus$ is the XOR and each

entry $\boldsymbol{\delta}[i] \sim$ Bernoulli $(p = p_{\boldsymbol{x}[i]})$. See § C for more details. Our approach works with other smoothing schemes such as uniform noise for l_1 threat models (Levine & Feizi, 2021), but we focus on these two due to their popularity. Gaussian smoothing preserves exchangeability (Gendler et al., 2021). Similar argument applies to sparse smoothing and other methods that are symmetric w.r.t. $\boldsymbol{x}_{n+1}$ and $\mathcal{D}_{\text{cal}}$.

The goal is to bound the smooth score $\hat{s}(\tilde{\boldsymbol{x}}, y)$ of any adversarial $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$. Since the base score function $s(\cdot,\cdot)$ often depends on a complex model such as a neural network, even computing the expected score $\hat{s}(\cdot,\cdot)$ is challenging, let alone finding the worst-case $\tilde{\boldsymbol{x}}$. Therefore, we follow the general recipe of relaxing the problem by searching over the space of all possible score functions $h(\cdot,\cdot) \in \mathcal{H}$. We focus on upper bounds, but the entire discussion equivalently applies to lower bounds by switching from $\max$ to $\min$. By definition we have $s(\cdot,\cdot) \in \mathcal{H}$, therefore it holds that:

$$\max_{\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})} \mathbb{E}[s(\xi(\tilde{\boldsymbol{x}}), y)] \leq \max_{\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x}), h \in \mathcal{H}} \mathbb{E}[h(\xi(\tilde{\boldsymbol{x}}), y)] \tag{6}$$

The solution to Eq. 6 is trivial unless we add additional constraints to the functions $h(\cdot,\cdot) \in \mathcal{H}$ that capture information about the actual score function $s(\cdot,\cdot)$. The tightness of the resulting bound is directly controlled by the constraints. First, we describe a baseline bound that only captures information about the mean of $s(\cdot,\cdot)$. This is exactly the bound used by RSCP. Then, we describe a second bound that leverages information about the entire distribution of scores via the CDF. In both cases, we only need black-box access to the score function and the underlying classifier, and we assume that $s(\cdot,\cdot) \in [a, b]$ is bounded (w.l.o.g. $a = 0, b = 1$).

Canonical view. It turns out that for both Gaussian and sparse smoothing it is sufficient to derive a so-called pointwise bound for a given $(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ pair since it can be shown that the maximum in Eq. 6 is always attained at a canonical $\tilde{\boldsymbol{x}}$ which is on the sphere of the respective ball. Namely, for the continuous $\mathcal{B}_r(\boldsymbol{x})$ we have the canonical vectors $\boldsymbol{x} = \mathbf{0}, \tilde{\boldsymbol{x}} = [r, 0, 0, \dots]$ that completely specify the problem. For the binary $\mathcal{B}_{r_a, r_d}$ we have the canonical $\boldsymbol{x} = [1, \dots, 1, 0, \dots, 0]$ and $\tilde{\boldsymbol{x}} = \mathbf{1} - \boldsymbol{x}$ where $\|\boldsymbol{x}\|_0 = r_d$ and $\|\tilde{\boldsymbol{x}}\|_0 = r_a$. Intuitively, the reason is due to the symmetry of the smoothing distributions and the balls (see § C).

Baseline bound. A straightforward approach only incorporates the expected smoothed score (mean) for the given input $\boldsymbol{x}$. Let $p = \mathbb{E}[s(\xi(\boldsymbol{x}), y)]$ for simplicity. With $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$ the baseline upper-bound for $\hat{s}(\tilde{\boldsymbol{x}}, y) = \mathbb{E}[s(\xi(\tilde{\boldsymbol{x}}), y)]$ is determined by the following problem:

$$\begin{aligned} \overline{s}_{\text{mean}}(\boldsymbol{x}, y) = \max_{h \in \mathcal{H}} \quad & \mathbb{E}[h(\xi(\tilde{\boldsymbol{x}}), y)] \\ \text{s.t.} \quad & \mathbb{E}[h(\xi(\boldsymbol{x}), y)] = p \end{aligned} \tag{7}$$

This bound discards a lot of information about the distribution of scores around the given $\boldsymbol{x}$. To remedy this, we incorporate the information from the CDF of the scores.

CDF-based bound. Let $a = b_1 < b_2 \leq \dots \leq b_{m-1} < b_m = b$ be m real numbers that partition the output space. Let $p_i = \Pr[s(\xi(x), y) \leq b_i]$. We define the problem:

$$\begin{aligned} \overline{s}_{\text{cdf}}(\tilde{\boldsymbol{x}}, y) = \max_{h \in \mathcal{H}} \quad & \mathbb{E}[h(\xi(\tilde{\boldsymbol{x}}), y)] \\ \text{s.t.} \quad \forall b_i : \quad & \Pr[h(\xi(\boldsymbol{x}), y) \leq b_i] = p_i \end{aligned} \tag{8}$$

The key insight for solving Eq. 8 is to upper bound the mean of h via the CDF. Intuitively, we compute the probability of each bin $[b_j, b_{j+1}]$ and choose the upper end of the bin to get an upper bound. This can be rewritten in terms of the CDF. Let $F_h(b_j) = \Pr[h(\boldsymbol{x}, y) \leq b_j]$, for any function h

$$\begin{aligned} \mathbb{E}[h(\boldsymbol{x})] &\leq \sum_{j=2}^{m} b_j \cdot [(F_h(b_j) - F_h(b_{j-1})] \\ &= b_m - \sum_{j=2}^{m-1} F_h(b_j) \cdot (b_{j+1} - b_j) \end{aligned} \tag{9}$$

Next, we show how to solve both problems for the two different smoothing schemes. For Gaussian smoothing, both problems in Eq. 7 and Eq. 8 have closed-form solutions as shown by Kumar et al. (2020). For sparse smoothing, Bojchevski et al. (2020) provides an efficient algorithm to solve Eq. 7. We extend their approach to also solve Eq. 8 which is a novel contribution of potentially independent interest, e.g. to certify graph neural networks with regression tasks.

In practice, $\overline{s}_{\text{cdf}}$ is tighter than $\overline{s}_{\text{mean}}$, and the improvement depends on the distribution of random scores. While we can easily combine both mean and CDF constraints to get a provably tighter bound, we focus only on CDF constraints.

Bounds for Gaussian smoothing. For any perturbed $\tilde{\boldsymbol{x}}$ with $||\tilde{\boldsymbol{x}} - \boldsymbol{x}||_2 \leq r$ we have the baseline bound $\hat{s}(\tilde{\boldsymbol{x}}, y) \leq \overline{s}_{\text{mean}}(\boldsymbol{x}, y) = \Phi_\sigma\left(\Phi_\sigma^{-1}(p) + r\right)$ where Φ_σ is the CDF of $\mathcal{N}(0, \sigma^2)$ and $p = \mathbb{E}_{\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})}[s(\boldsymbol{x} + \boldsymbol{\delta}, y)]$ is the clean expected score. We can get the lower bound by flipping the sign of r. The CDF bound is $\hat{s}(\tilde{\boldsymbol{x}}, y) \leq \overline{s}_{\text{cdf}}(\boldsymbol{x}, y)$ with

$$\overline{s}_{\text{cdf}} = b_m - \sum_{j=2}^{m-1} \Phi_\sigma\left(\Phi_\sigma^{-1}(p_j) - r\right)(b_{j+1} - b_j) \tag{10}$$

where $p_j = \Pr_{\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})}[s(\boldsymbol{x} + \boldsymbol{\delta}, y) \leq b_j]$. The corresponding lower bound and derivations are in § D.2.

Bounds for sparse smoothing. To solve both optimization problems, we apply the same approach as Bojchevski et al. (2020), dividing the input space into regions of constant likelihood ratio $\mathcal{X} = \cup_i^I \mathcal{R}_i$ where $\mathcal{R}_i = \{\boldsymbol{z} : \Pr[\xi(\boldsymbol{x}) = \boldsymbol{z}] / \Pr[\xi(\tilde{\boldsymbol{x}}) = \boldsymbol{z}] = c_i\}$. For the mean variant, we greedily distribute the p mass to each region (from the highest to the lowest ratio) until the constraint is satisfied. For the CDF variant, we instead distribute

the p_j masses in each region and each bin $[b_j, b_{j+1}]$. Technical details, including the linear programming formulations, are in § C. The runtime complexity scales linearly with the number of regions which is $I = r_a + r_d + 1$. We provide an efficient algorithm that runs in less than a few milliseconds.

Clean vs. observed input. In the discussion we refer to a clean $\boldsymbol{x}$ and a perturbed $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$. In practice, we do not know whether the *observed* input $\boldsymbol{x}'$ is clean or perturbed. However, since the l_2-ball is symmetric, if $\boldsymbol{x}' \in \mathcal{B}_r(\boldsymbol{x})$ then also $\boldsymbol{x} \in \mathcal{B}_r(\boldsymbol{x}')$. Thus, computing an upper bound for any observed $\boldsymbol{x}'$ in the threat model yields a valid upper bound for the clean $\boldsymbol{x}$, $\hat{s}(\boldsymbol{x}) \leq \overline{s}(\boldsymbol{x}')$. That is, we do not assume that the clean input is given at test time. For sparse data $\boldsymbol{x}' \in \mathcal{B}_{r_a,r_d}(\boldsymbol{x}) \implies \boldsymbol{x} \in \mathcal{B}_{r_d,r_a}(\boldsymbol{x}')$, so we need to switch r_a and r_d when computing the certificate. Similar conclusions apply for an observed and potentially perturbed $\mathcal{D}'_{\text{cal}}$ since the clean $\mathcal{D}_{\text{cal}} \in \mathbb{B}_{k,\mathcal{B}}(\mathcal{D}'_{\text{cal}})$ for any $\mathcal{D}'_{\text{cal}} \in \mathbb{B}_{k,\mathcal{B}}(\mathcal{D}_{\text{cal}})$. This detail is not important for standard certificates since they only certify that the prediction does not change.

5. CAS: CDF-Aware Sets

We use the CDF-based bounds to obtain conservative prediction sets for evasion and conservative thresholds for poisoning attacks. We summarize our approach with the pseudocode in Algorithm 1 that works with any score function[3].

Algorithm 1 CDF-Aware Sets (CAS, Evasion)

$q_\alpha = \text{Quant}\left(\alpha; \{\hat{s}(\boldsymbol{x}, y)\}_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{cal}}}\right)$ ▷ Clean quantile
Compute $\overline{s}_{\text{cdf}}(\boldsymbol{x}, y)$, e.g. with Eq. 10 ▷ Upper bound
Return $\overline{\mathcal{C}}_\alpha = \{y : \overline{s}_{\text{cdf}}(\boldsymbol{x}, y) \geq q_\alpha\}$ ▷ Conservative set

Calibration-time variant. For evasion we need to compute $\overline{s}(\boldsymbol{x}, y)$ via solving Eq. 8 (or Eq. 7) for each test point and each class. This can be computationally costly if we have many classes (e.g. `ImageNet` has 1000) at deployment. We define an alternative approach that instead needs only a *lower bound* $\underline{s}(\boldsymbol{x}, y)$ for each $\boldsymbol{x} \in \mathcal{D}_{\text{cal}}$ and the true y. The key insight is that we can directly compare the smooth test score $\hat{s}(\tilde{\boldsymbol{x}}_{n+1}, y)$ against a conservative (lower) quantile.

Proposition 5.1. *For $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$ and $(\boldsymbol{x}_{n+1}, y_{n+1})$ exchangeable with $\mathcal{D}_{\text{cal}}$, define*

$$\underline{q_\alpha} = \text{Quant}\left(\alpha; \{\underline{s}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}\}\right) \tag{11}$$

For prediction sets $\overline{\mathcal{C}}_\alpha(\tilde{\boldsymbol{x}}_{n+1}) = \{y : \hat{s}(\tilde{\boldsymbol{x}}_{n+1}, y) \geq \underline{q_\alpha}\}$ we have $\Pr[y_{n+1} \in \overline{\mathcal{C}}_\alpha(\tilde{\boldsymbol{x}}_{n+1})] \geq 1 - \alpha$. Moreover, the vanilla CP covers the true label with probability $\geq 1 - \beta$ for

$$\beta = \text{Quant}^{-1}\left(q_\alpha; \{\underline{s}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}\}\right) \tag{12}$$

and $\text{Quant}^{-1}(t; A) = \min\{\tau' : \text{Quant}(\tau'; A) \geq t\}$.

[3]Our code and experiments are in the github repository `soroushzargar/CAS`.

With Prop. 5.1 we need only $|\mathcal{D}_{\text{cal}}|$ certified bounds as a preprocessing step. At test time we directly plug in $\hat{s}(\tilde{\boldsymbol{x}}_{n+1}, y)$ and not its upper bound. Since $|\mathcal{D}_{\text{cal}}|$ is often significantly smaller than the test set the computational savings are substantial (see Table 3). With Eq. 12 we can compute a lower bound on the coverage of vanilla (non-robust) CP under perturbation, where by definition $1 - \beta \leq 1 - \alpha$. This is a generalization of Theorem 2 in Gendler et al. (2021).

Poisoning. For poisoning attacks we simply use the conservative threshold $\underline{q_\alpha}$ from Eq. 3 or Eq. 5 where we use the CDF-bounds in the constraints (see § 3.2). If the test examples are assumed clean we return $\overline{\mathcal{C}}_\alpha = \{y : \hat{s}(\boldsymbol{x}, y) \geq \underline{q_\alpha}\}$. Since robustness to evasion and poisoning are independent, we can achieve simultaneous robustness to both evasion and poisoning via $\overline{\mathcal{C}}_\alpha = \{y : \overline{s}_{\text{cdf}}(\boldsymbol{x}, y) \geq \underline{q_\alpha}\}$.

To solve the two poisoning optimization problems we rewrite them as mixed-integer linear programs and solve them with an off-the-shelf solver. We only need $2 \cdot |\mathcal{D}_{\text{cal}}|$ binary variables for Eq. 3 and $|\mathcal{D}_{\text{cal}}| \times |\mathcal{Y}|$ binary variables for Eq. 5. See § G for technical details. Since the calibration set is relatively small we can solve the MILPs in just a few minutes. Thus, our guarantees are practically feasible.

6. Finite Sample Correction

Solving Eq. 7, or Eq. 8 requires the true mean or CDF. Since exact computation is intractable, we use Monte-Carlo (MC) samples. To ensure a valid certificate, we bound the exact statistics via concentration inequalities. The resulting confidence intervals are valid together with adjustable $1 - \eta$ probability. To account for this we calibrate with $\alpha' = \alpha - \eta$ so that the final sets still have $1 - \alpha$ coverage (see § E). RSCP did not include such finite-sample correction, and the resulting sets are only asymptotically valid without it.

Yan et al. (2024) incorporates the correction directly in the conformity scores, leveraging exchangeability between MC-estimated calibration scores and clean test scores. We discuss this in § E and propose another approach built on Prop. 5.1. Our correction results in smaller sets for CAS with the same guarantee; and similar results for RSCP (see § 7).

Proposition 6.1. *Let $\underline{s}_{\text{cdf}+}(\boldsymbol{x}_i, y_i) \leq \underline{s}_{\text{cdf}}(\boldsymbol{x}_i, y_i)$ hold with $1 - \eta/(2|\mathcal{D}_{\text{cal}}|)$ probability for each $(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}$, and $\hat{s}_+(\tilde{\boldsymbol{x}}_{n+1}, y_{n+1}) \geq \hat{s}(\tilde{\boldsymbol{x}}_{n+1}, y_{n+1})$ hold with $1 - \eta/(2|\mathcal{Y}|)$ probability. Define the conservative $\underline{q}_{\alpha+} = \text{Quant}\left(\alpha - \eta; \left\{\underline{s}_{\text{cdf}+}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}\right\}\right)$ and $\overline{\mathcal{C}}_{\alpha+}(\boldsymbol{x}_{n+1}) = \{y : \hat{s}_+(\boldsymbol{x}_{n+1}, y) \geq \underline{q}_{\alpha+}\}$. Then*

$$\Pr[y_{n+1} \in \overline{\mathcal{C}}_{\alpha+}(\tilde{\boldsymbol{x}}_{n+1})] \geq 1 - \alpha \tag{13}$$

We compute $\underline{s}_{\text{cdf}+}(\boldsymbol{x}_i, y_i)$ by solving the minimization variant of Eq. 8 with CDF error correction through the Dvoretzky–Kiefer–Wolfowitz inequality (Dvoretzky et al.,

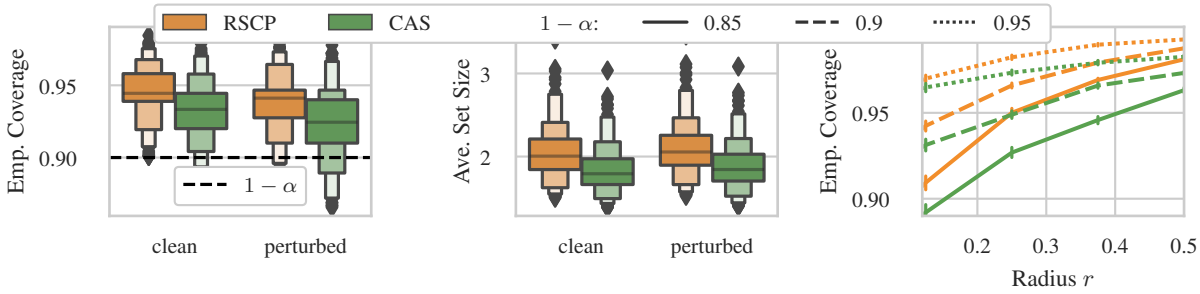

Figure 1. Empirical coverage [left] and average set size [middle] of RSCP and CAS for clean and perturbed data. All sets are certified robust up to radius $r = 0.125$. [Right] Empirical coverage for different certified radii (on clean data). All results are for `CIFAR-10` with Gaussian smoothing ($\sigma = 0.25$). CAS is less conservative since it is closer to the nominal $1 - \alpha$, and has smaller sets.

1956). We define $\hat{s}_+(\tilde{\boldsymbol{x}}_{n+1}, y) = \frac{1}{n_s}\sum^{n_s} s(\xi(\boldsymbol{x}_{n+1}), y) + \epsilon$ where ϵ is the error given by the Bernstein confidence interval. In short, we divide the η budget between $|\mathcal{D}_{\text{cal}}| + |\mathcal{Y}|$ estimates. This divides between all calibration scores (only for the true class), and $|\mathcal{Y}|$ classes for the test input.

The corrected CP in (Yan et al., 2024) compares $q_{\alpha,\text{mc}}$ and $\overline{s}_+(\boldsymbol{x}_{n+1}, y) + \epsilon_{\text{hoef}}$, where the quantile $q_{\alpha,\text{mc}}$ is computed on the clean scores estimated with MC-sampling without correction. Instead, ϵ_{hoef} is added to account for the difference between the unseen clean test MC-score (exchangeable with the MC-calibration scores) and the upper bound which only bounds the true (non-MC) mean. See § E for details. In our case, we compare the corrected quantile $\underline{q}_{\alpha+}$ and the corrected estimate of the input test score $\hat{s}_+(\boldsymbol{x}_{n+1}, y)$. Instead of a Hoeffding bound we can use the tighter Bernstein bound for $\hat{s}_+(\boldsymbol{x}_{n+1}, y)$ since we have access to it. In addition, to compute $\underline{q}_{\alpha+}$ we use DKW-corrected scores which introduce less error compared to the Hoeffding bound.

Feature Poisoning. We find the lower bound quantile $\underline{q}_\alpha$ (Eq. 3) using smooth scores (see § G, Eq. 24). To apply sample correction, again with an error budget of η we divide this budget equally between calibration points. For each calibration point, the CDF bound with correction finds a probabilistic lower bound on the clean smooth score. Since the test scores are computed with MC-estimation, we directly bound the MC-estimated clean calibration scores for exchangeability. Since we do not have access to the clean calibration scores, following Yan et al. (2024) we use Hoeffding's inequality. The corrected quantile is lower than the clean quantile for MC-estimated calibration scores.

Proposition 6.2. *Let $\underline{s}_{\text{cdf}+}(\tilde{\boldsymbol{x}}_i, y_i) \leq \underline{s}_{\text{cdf}}(\tilde{\boldsymbol{x}}_i, y_i)$ hold with $1 - \eta/(2|\mathcal{D}_{\text{cal}}|)$ probability for all $(\tilde{\boldsymbol{x}}_i, y_i) \in \mathcal{D}_{\text{cal}}$. Let $\underline{q}_{\alpha+}$ be the solution to Eq. 24 (Eq. 3 with CDF bounds) for $\alpha = \alpha' - \eta$ with $\underline{s}_i = \underline{s}_{\text{cdf}+}(\tilde{\boldsymbol{x}}_i, y_i) - \epsilon_{\text{hoef}}$. Then for each new test point $\boldsymbol{x}_{n+1}$ exchangeable with $\mathcal{D}_{\text{cal}}$ the prediction set defined as $\overline{\mathcal{C}}(\boldsymbol{x}_{n+1}) = \{y : s_{\text{mc}}(\boldsymbol{x}_{n+1}, y) \geq \underline{q}_{\alpha+}\}$ has $1 - \alpha'$ coverage.*

Here $\epsilon_{\text{hoef}} = \sqrt{\log(2/\eta)/2|\mathcal{D}_{\text{cal}}|}$ comes from the Hoeffding inequality. In label poisoning we do not use randomly smoothed scores, therefore sample correction is not needed.

7. Experiments

For evasion, we compare CAS with RSCP (Gendler et al., 2021). Even though the original RSCP is not able to handle sparse or discrete data, we extend it and use it as an additional baseline (see § C). There are no baselines for poisoning. Since both RSCP and CAS have the same guaranteed coverage we focus on two main metrics: the average size of prediction sets (or efficiency) and the empirical coverage. Ideally, we want the coverage to be concentrated around the nominal $1 - \alpha$. Higher coverage costs larger prediction sets. In § J we report additional experiments including the singleton hits ratio metric. We also consider the maximum perturbation radius such that robust CP has the same set size as standard CP (averaged across test points). This size-preserving r is the largest certified radius which we can get "for free". On all metrics CAS outperforms RSCP.

Setup. We evaluate our method on two image datasets: `CIFAR-10` (Krizhevsky, 2009) and `ImageNet` (Deng et al., 2009), and one node-classification (graph) dataset `Cora-ML` (McCallum et al., 2004). We used `ResNet-110` and `ResNet-50` pretrained on `CIFAR-10` and `ImageNet` with noisy data augmentation from Cohen et al. (2019). We trained a `GCN` model (Kipf & Welling, 2017) for node classification. All models are trained on data augmented with noise. The GNN is trained with 20 nodes per class with stratified sampling as the training set and similarly sampled validation set. The size of the calibration set is between 100 and 150 (sparsely labeled setting). We use APS as the main score function.

For each dataset, we pick a number of test points at random (900 for `CIFAR-10`, 400 for `ImageNet`, and 2480 nodes for `Cora`). We estimate the expected smooth scores with

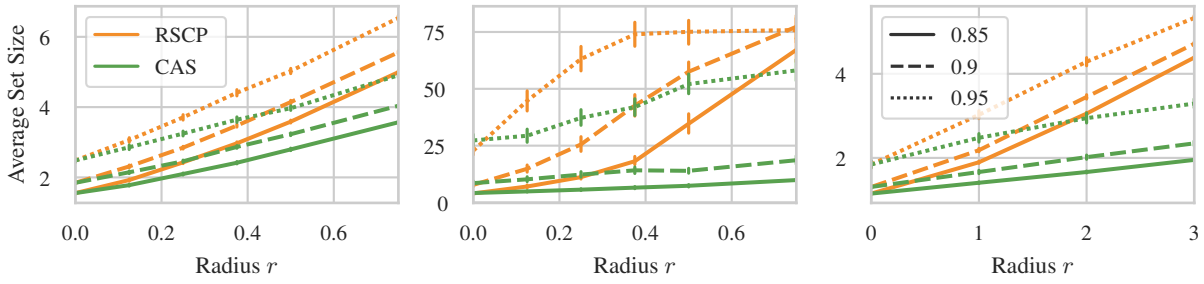

Figure 2. Average set size of CAS and RSCP under evasion for (from left to right) `CIFAR-10`, `ImageNet` (with TPS), and `Cora-ML`.

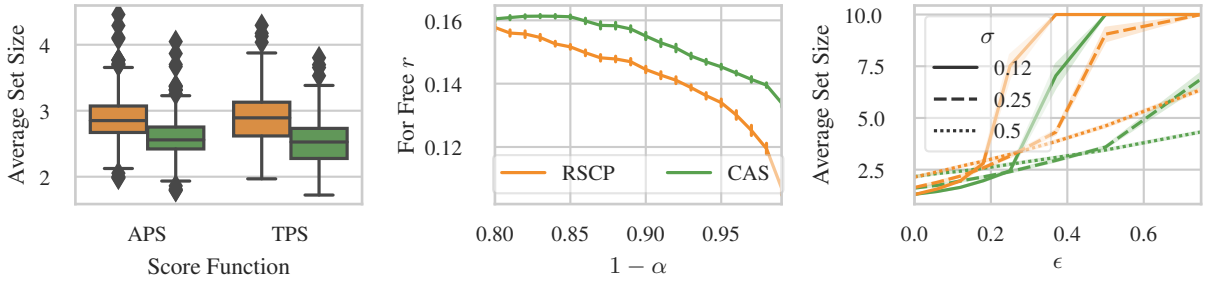

Figure 3. [Left] Set size for $r = 0.12$ with different scores. [Middle] Maximum set size-preserving radius (average over test points). Both results are on `CIFAR-10` dataset and $\sigma = 0.25$. [Right] The effect of smoothing parameter σ on the set size across a range of radii for `CIFAR-10` dataset with error correction for 10^4 samples.

10^4 Monte-Carlo samples. All results are an average of 100 runs with exchangeable calibration sampling (details in § J).

Evasion Certificate. The conservative robust sets are necessarily larger than non-robust sets. Consequently, on Fig. 1 (left) we observe a higher empirical coverage on clean data compared to the nominal $1-\alpha$. The coverage on perturbed inputs which we find with a PGD attack (Madry et al., 2017) is above $1-\alpha$ verifying our theory. In Fig. 1 (right) we see that the empirical coverage increases with the certified radius r and is $1-\alpha$ for $r = 0$. CAS is less needlessly conservative (grows slower with r) than RSCP while still providing the same guarantee. This leads to improved efficiency (smaller sets) as shown in Fig. 1 (middle). The set size is slightly higher for perturbed inputs.

In Fig. 2 we see that CAS's results in smaller prediction sets, across all radii, and all nominal $1-\alpha$ values, and as in Fig. 3 (left) all scores. The improvement is substantial and also grows with r – for larger radii it is doubled or even tripled, especially on `ImageNet` and `Cora-ML`. Similarly, Fig. 3 (middle) shows that with CAS we can consistently certify a larger maximum radius "for free".

Calibration-time evasion. Following Prop. 5.1 if we use vanilla (non-robust) CP, in the adversarial setup we can certify a lower bound $1-\beta$ on the worst-case robust coverage. In Fig. 4 (left) we see that the certificate based on CAS leads to a better (higher) lower bound. At the same time, Prop. 5.1 implies that we can avoid computing upper bounds for the test points and instead account for the effect of the adversary by choosing a conservative conformal threshold ($\underline{q_\alpha}$) via the lower bound on the calibration scores. Fig. 4 (middle) show the set size distribution for test-time vs. calibration-time evasion. The results for RSCP are comparable. CAS shows smaller sets for the calibration-time certificate. This approach is also computationally faster especially for datasets with a high number of classes, which is discussed in § B.

Ablation study. In Fig. 3 (right) we study the effect of the smoothing strength as controlled by σ in $\mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$. For all σ values and all radii r we get the same $1-\alpha$ coverage guarantee, however, there is a clear trade-off for choosing σ. A smaller amount of smoothing results in a smaller set size in the beginning, but the set sizes grows rapidly by increasing the certified radius. In all cases, CAS is better than RSCP. Here for each σ we use the model that is pretrained on the same noise augmentation.

Finite sample correction. The previous results were without error correction since RSCP did not account for finite-sample errors when estimating the smooth scores with

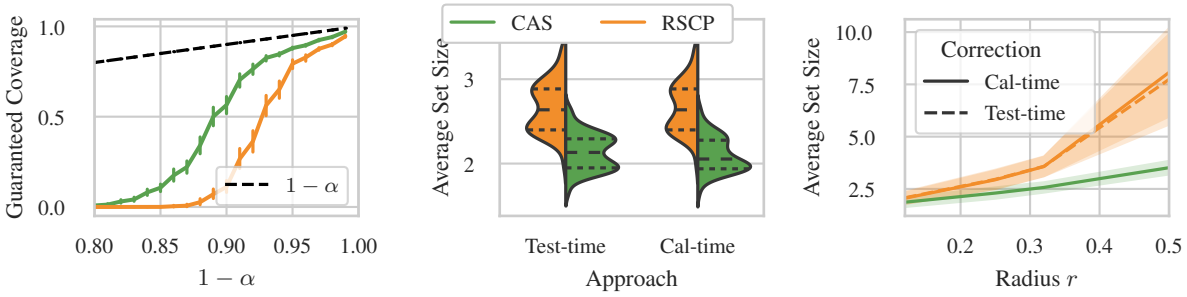

Figure 4. [Left] Lower bound $1-\beta$ on the robust coverage of vanilla CP (Prop. 5.1). CAS certifies a larger lower bound. [Middle] Distribution of prediction set sizes using the slower test-time vs. the faster calibration-time evasion certificate. [Right] Set sizes for RSCP and CAS with account for finite sample error. All results are for `CIFAR-10` with $\sigma = 0.25$.

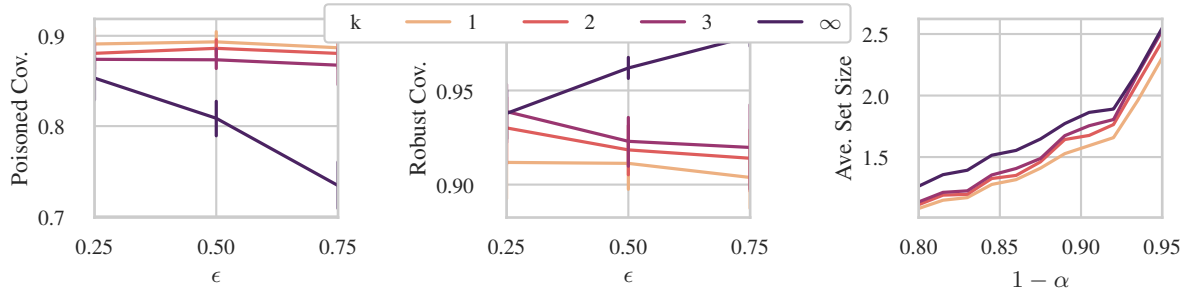

Figure 5. [Left] The coverage of vanilla CP under calibration set with poisoned features. [Middle] The result of robust CP on the same calibration set. [Right] The average set size of the CP robust to feature poisoning for a range of nominal coverages given the clean calibration data, and $r = 0.12$. All results are for `CIFAR-10` dataset with $\sigma = 0.25$.

Monte-Carlo samples. The sets are still asymptotically valid without correction, as confirmed by Fig. 1 (left); however, correction is necessary for a valid certificate as argued by Yan et al. (2024). In Fig. 4 (right) we see that the size for RSCP quickly explodes, reaching almost all classes ($|\mathcal{Y}| = 10$) for large radii, while CAS maintains low average size. Moreover, CAS has smaller standard deviation across test inputs. CAS uses calibration-time correction (see § E).

Label poisoning. Next, we study label poisoning where now the attacker can perturb the ground-truth labels of the calibration points. In Table 1 we see that increasing the budget k leads to predictably larger set size and larger empirical coverage. The difference to the clean calibration set ($k = 0$) is minor, showing that provable label robustness comes almost for free for small k. While Einbinder et al. (2022) show that standard CP is already naturally robust to random (non-worst case) label noise, Table 1 shows that adversarial label noise can break the guarantee even for small budget k.

Feature poisoning. Since there are no baselines that provide robust coverage guarantees under poisoning we can only study the behaviour of CAS. First, we consider feature poisoning where the attacker is allowed to change k calibration points which we refer to as the budget, each of which can be perturbed in a given ball $\mathcal{B}_r(\boldsymbol{x})$ (see Eq. 3). In Fig. 5 (left) we show that the coverage can slightly decrease via poisoning the features with a limited budget. This drop becomes significant when the adversary can perturb all the calibration points. To poison the data, we run the PGD attack on all calibration points and decide which point to perturb by solving Eq. 3 (specifically Eq. 24) with maximization goal. Fig. 5 (middle) shows the robustness of CAS even under an infinite budget which verifies Prop. 3.2. We also show the set size of robust CP in Fig. 5 (right). We see that as expected a smaller budget k leads to less conservative sets which translates to smaller set sizes. Interestingly, for small

Table 1. Label poisoning for `CIFAR-10`.

k	Cov. (Clean)	Vanilla Cov. (Pert)	Robust Cov. (Pert)	Set Size (Clean)
0	0.897	0.897	0.897	1.41
1	0.916	0.872	0.900	1.58
2	0.923	0.859	0.901	1.62

		Emp. Coverage		Ave. Set Size	
r	k	With	Without	With	Without
0.12	3	94.6	94.5	1.84	1.83
	$\lvert\mathcal{D}_{\text{cal}}\rvert$	97.7	96.3	3.17	2.47
0.25	3	94.0	94.0	1.756	1.752
	$\lvert\mathcal{D}_{\text{cal}}\rvert$	99.6	98.7	7.32	4.48

Table 2. CAS for feature poisoning with and without finite-sample correction.

r (e.g. $r = 0.12$) even with an infinite budget the set size does not increase drastically. Making CP robust to poisoning comes at only a small cost. Note that for $k = \infty$ setting each calibration score to its lower bound is one solution to Eq. 3, which equals calibration-time evasion.

Similar to evasion, in Table 2 we show the results of CAS robust to feature poisoning with and without sample correction. For sample correction, we use Prop. 6.2. Note that label-poisoning does not require sample correction.

8. Related Work

Ghosh et al. (2023) introduce the notion of probabilistically robust CP. Intuitively, their guarantee is w.r.t. the average adversarial input, while for RSCP and our method the guarantee is w.r.t. the worst-case input. They produce more efficient sets via a quantile of quantiles method – one quantile considers the adversarial examples around a datapoint and the other finds the CP threshold over the first set of quantiles. This enables a tuneable trade-off between nominal performance and robustness. Our method is orthogonal since we consider exact coverage, and Ghosh et al. (2023)'s probabilistic robustness can be applied on top of ours.

Cauchois et al. (2020) propose an approach which returns prediction sets that are robust to distribution shift between the calibration and the test distribution. As input, their method needs an upper bound ρ on the f-divergence between the two distributions, which they estimate from data. In principle, for a given radius r one can derive a suitable ρ, however, the resulting sets can be needlessly too conservative. We can conclude this from the fact that the optimization problem with the resulting f-divergence constraint is a relaxation as shown by Dvijotham et al. (2020) in a different context (classification certificates). Gendler et al. (2021) extensively discuss the differences between RSCP and Cauchois et al. (2020)'s approach across various settings (e.g. model trained with and without noise) and report better or equal efficiency. With CAS outperforming RSCP, we draw similar conclusions by transitivity. Further discussion is in § F.2.

Two concurrent works use the same bound as RSCP, but improve the sets by modifying other aspects of the algorithm. Yan et al. (2024) adopt robust conformal training (Stutz et al., 2022) and propose to transform the smooth score (ranking + sigmoid scaling) using an additional holdout set. Kang et al. (2024) integrate a reasoning component via probabilistic circuits. Both are completely orthogonal to our method and can be directly improved with our CDF bounds.

Angelopoulos et al. (2022) extend conformal prediction to control the expected value of any monotone loss function, including adversarial risk (see Proposition 7). However, they do not propose an algorithm to compute the worst-case adversarial loss. Einbinder et al. (2022) show that standard CP is already robust to *random* label noise, e.g. resulting from wrong annotation or any other natural source of noise. Unlike our work, they do not study robustness to adversarial (worst-case) label perturbations.

9. Conclusion

We provide certified robustness for conformal prediction both for evasion and poisoning attacks. We propose a CDF-aware bound on the conformity scores under adversarial perturbation. Our bound is empirically tighter and leads to consistent improvements compared to previous certificates. We further propose novel certificates against feature and/or label poisoning of the calibration set. We generalize both results to discrete and binary (sparse) data. Finally, we show how we can correct for finite-sample error. Our calibration-time approach for robustness to evasion that reduces the inflation of set sizes when correcting for finite samples. Overall, our method CAS yields provably robust yet efficient (small) prediction sets.

Limitations. We identify three main limitations. First, the coverage guarantee is marginal, which means that it holds on average across the entire input domain. Conditional coverage $\Pr[y \in \mathcal{C}(\boldsymbol{x}) \mid \boldsymbol{x}]$ is impossible to achieve without strong assumptions (Barber et al., 2019). Achieving near-conditional coverage is still and open problem. This means that CP can have over-coverage or under-coverage for different groups which can be unfair. This holds true for both vanilla CP and robust CP. Lu et al. (2022) consider a group-conformal variant to equalize coverage across groups, however, unfairness can still be reflected in the set-size. Studying the intersection of robustness and fairness is an exciting future direction. Second, while randomized smoothing is a powerful and flexible method, estimating empirical statistics requires a large number of Monte-Carlo samples. This can be computationally expensive. Finally, we assumed that the goal of the attacker is to reduce the empirical coverage and designed our certificate to prevent this. However, the attacker may have other goals, e.g. to increase the set size, or to attack only a subset of labels.

Acknowledgements

We thank Giuliana Thomanek for feedbacks on our draft.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021.

Angelopoulos, A. N. and Bates, S. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., and Schuster, T. Conformal risk control. *ArXiv*, abs/2208.02814, 2022. URL https://api.semanticscholar.org/CorpusID:251320513.

Angelopoulos, A. N., Bates, S., et al. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591, 2023.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Barber, R. F., Candès, E. J., Ramdas, A., and Tibshirani, R. J. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 2019.

Bojchevski, A., Gasteiger, J., and Günnemann, S. Efficient robustness certificates for discrete data: Sparsity-aware randomized smoothing for graphs, images and more. In *International Conference on Machine Learning*, pp. 1003–1013. PMLR, 2020.

Cauchois, M., Gupta, S., Ali, A., and Duchi, J. C. Robust validation: Confident predictions even when distributions shift. *arXiv preprint arXiv:2008.04267*, 2020.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pp. 1310–1320. PMLR, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. URL https://api.semanticscholar.org/CorpusID:57246310.

Dvijotham, K., Hayes, J., Balle, B., Kolter, Z., Qin, C., György, A., Xiao, K. Y., Gowal, S., and Kohli, P. A framework for robustness certification of smoothed classifiers using f-divergences. In *International Conference on Learning Representations*, 2020. URL https://api.semanticscholar.org/CorpusID:213452491.

Dvoretzky, A., Kiefer, J., and Wolfowitz, J. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, pp. 642–669, 1956.

Einbinder, B.-S., Bates, S., Angelopoulos, A. N., Gendler, A., and Romano, Y. Conformal prediction is robust to label noise. *ArXiv*, abs/2209.14295, 2022. URL https://api.semanticscholar.org/CorpusID:262091979.

Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Gendler, A., Weng, T.-W., Daniel, L., and Romano, Y. Adversarially robust conformal prediction. In *International Conference on Learning Representations*, 2021.

Ghosh, S., Shi, Y., Belkhouja, T., Yan, Y., Doppa, J., and Jones, B. Probabilistically robust conformal prediction. In Evans, R. J. and Shpitser, I. (eds.), *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 681–690. PMLR, 31 Jul–04 Aug 2023. URL https://proceedings.mlr.press/v216/ghosh23a.html.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Huang, K., Jin, Y., Candes, E., and Leskovec, J. Uncertainty quantification over graph with conformalized graph neural networks. *arXiv preprint arXiv:2305.14535*, 2023.

Kang, M., Gürel, N. M., Li, L., and Li, B. COLEP: Certifiably robust learning-reasoning conformal prediction via probabilistic circuits. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=XN6ZPINdSg.

Kipf, T. and Welling, M. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907, 2017.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Kumar, A., Levine, A., Feizi, S., and Goldstein, T. Certifying confidence via randomized smoothing. *Advances in Neural Information Processing Systems*, 33:5165–5177, 2020.

Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE symposium on security and privacy (SP)*, pp. 656–672. IEEE, 2019.

Lee, G.-H., Yuan, Y., Chang, S., and Jaakkola, T. Tight certificates of adversarial robustness for randomly smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019.

Levine, A. and Feizi, S. Improved, deterministic smoothing for l_1 certified robustness. 2021.

Li, L., Xie, T., and Li, B. Sok: Certified robustness for deep neural networks. 2023.

Lu, C., Lemay, A., Chang, K., Höbel, K., and Kalpathy-Cramer, J. Fair conformal predictors for applications in medical imaging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 12008–12016, 2022.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

McCallum, A., Nigam, K., Rennie, J. D. M., and Seymore, K. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3:127–163, 2004.

Mujkanovic, F., Geisler, S., Günnemann, S., and Bojchevski, A. Are defenses for graph neural networks robust? *Advances in Neural Information Processing Systems*, 35: 8954–8968, 2022.

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 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Romano, Y., Sesia, M., and Candès, E. J. Classification with valid and adaptive coverage. *arXiv: Methodology*, 2020.

Sadinle, M., Lei, J., and Wasserman, L. A. Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 114:223 – 234, 2018.

Salman, H., Li, J., Razenshteyn, I., Zhang, P., Zhang, H., Bubeck, S., and Yang, G. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019.

Silva, S. H. and Najafirad, P. Opportunities and challenges in deep learning adversarial robustness: A survey. *arXiv preprint arXiv:2007.00753*, 2020.

Stutz, D., Dvijotham, K. D., Cemgil, A. T., and Doucet, A. Learning optimal conformal classifiers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=t8O-4LKFVx.

Teng, J., Wen, C., Zhang, D., Bengio, Y., Gao, Y., and Yuan, Y. Predictive inference with feature conformal prediction. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0uRm1YmFTu.

Vovk, V., Gammerman, A., and Shafer, G. Algorithmic learning in a random world. 2005.

Yan, G., Romano, Y., and Weng, T.-W. Provably robust conformal prediction with improved efficiency. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=BWAhEjXjeG.

Yuan, X., He, P., Zhu, Q., and Li, X. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9): 2805–2824, 2019.

Zargarbashi, S. H., Antonelli, S., and Bojchevski, A. Conformal prediction sets for graph neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://openreview.net/forum?id=zGf8J0bNfX.

A. More On Conformal Prediction

Conformity vs. non-conformity scores. As mentioned in § 2, for CP we need to define a score function that quantifies the agreement between the input and each label. Equivalently, one can define CP with a *non*-conformity score function that captures disagreement instead. In this case, the conformal threshold is the $1-\alpha$ quantile of the calibration true scores. Similarly, in the test time, labels with score *less* than the threshold are included in the prediction set. Both approaches are equivalent up to a change in the sign of the scores. The latter setup is used in (Gendler et al., 2021) and is equivalent to our implementation that uses conformity scores. Our choice of agreement score is due to simplicity.

Score function. In § 2 we mentioned that conformal prediction returns guaranteed sets regardless of the score function employed. Specifically, any score function maintaining the exchangeability (between calibration and test) is viable. In brief, the exchangeability of random variable $Z_1, \dots, Z_n$ means that the joint distribution of the variables is insensitive to the order/index. In other words for any permutation function $\psi : [n] \mapsto [n]$ we have $\Pr[Z_1, \dots, Z_n] = \Pr\left[Z_{\psi(1)}, \dots, Z_{\psi(n)}\right]$. Assuming the calibration set to be exchangeably sampled from the data distribution, any permutation equivariant transformation on the data still preserves the exchangeability. Conclusively, the smooth scores from Gendler et al. (2021) and Bojchevski et al. (2020) are both permutation equivariant (the smoothing applies similarly to all calibration and test points regardless of their order). Therefore, smoothing scores maintains exchangeability.

While any score function preserving the exchangeability maintains the conformal guarantee, better scores result in better performance with respect to the metric of interest. For instance, even a function that returns uniform conformity scores at random provides a valid guarantee, although the prediction sets will be large.

Various score functions are proposed in the literature of conformal classification ranging from simple softmax function on top of model's result (Sadinle et al., 2018), to more complex functions leveraging information from embedding spaces of the model (Teng et al., 2023), or from the confidence of adjacent datapoints within a network structure (Zargarbashi et al., 2023). The expected score within the smoothing scheme around an input is no exception as it only involves the datapoint itself and applies symmetrically to all datapoints. Similar conditions hold for any approximation of that expectation e.g. the mean of Monte-Carlo samples. See §B in Yan et al. (2024) for a longer discussion.

Effect of the calibration set size. With a calibration set exchangeably sampled from the data distribution (infinite samples), conformal prediction provides a marginal coverage of at least $1-\alpha$ (Eq. 1). This probability is also upper bounded by $1-\alpha+1/(n+1)$. Precisely, the coverage is distributed as $\text{Beta}(n+1-l, l)$ with $l = \lfloor (n+1)\alpha \rfloor$.

For a finite set of points and an exchangeably sampled calibration subset, e.g. transductive node-classification, Huang et al. (2023) show that the coverage probability, $\text{Cov}(\mathcal{D}) = (1/|\mathcal{D}|)\sum_{(\boldsymbol{x}_i, y_i)\in\mathcal{D}} \mathbf{1}[y_i \in \mathcal{C}(\boldsymbol{x}_i)]$ is distributed as

$$\Pr\left[\text{Cov}(\mathcal{D}) \le t\right] = 1 - \Phi_{\text{HG}}(\lfloor \alpha(n+1) \rfloor - 1; M+N, N, \lceil Mt \rceil + \lfloor \alpha(n+1) \rfloor) \tag{14}$$

Where $M = |\mathcal{D}|$, $N = |\mathcal{D}_{\text{cal}}|$ is the size of the calibration set, and $\Phi_{\text{HG}}(P, p, K)$ is the CDF function of hypergeometric distribution of population P, sample size p, and K successful samples within the population.

This means that the coverage probability on standard CP is concentrated around $1-\alpha$. It also means that the variance around $1-\alpha$ decrease as the size of $\mathcal{D}_{\text{cal}}$ increases. When moving the threshold from q_α to any other value q' within the domain of the score function (as in poisoning), the new threshold will correspond to another quantile $\beta = \text{Quant}^{-1}(q'; \mathcal{D}_{\text{cal}})$ and the coverage will be similarly concentrated around $1-\beta$.

Access to a large calibration set (e.g. 1000 points) is unrealistic. Even with a large set of labeled points, there is an open question of whether to use a portion of it for training the model toward better accuracy which can help even in the efficiency of CP. While we ran our experiments with the sparse labeled setting, increasing the size of the calibration set will result in similar values on average but the results will be more concentrated following the distribution of conformal probability.

Conservative coverage. Both RSCP and CAS result in an empirical coverage higher than $1-\alpha$ for clean data. This is since the vanilla prediction set is a subset of their conservative prediction set. The empirical coverage for RSCP is even higher compared to CAS since it uses looser bounds on the score and the prediction sets are *unnecessarily* more conservative. Higher empirical coverage is gained by larger prediction sets; therefore the goal of Robust CP is to find conservative sets that cover the worst-case perturbed input with higher than $1-\alpha$ probability but not by increasing the set size significantly.

One-sided robust guarantee. Although CP comes with a two-sided coverage guarantee (upper and lower bound on the coverage probability), our robust coverage guarantee is one-sided – we only guarantee that the coverage is larger than $1-\alpha$. The standard two-sided guarantee relies on exchangeability. However, since the adversary might perturb each point

differently, i.e. we have a non-symmetric mapping from clean $\boldsymbol{x}$ to perturbed $\tilde{\boldsymbol{x}}$; therefore, the perturbed points are no longer exchangeable. Another strategy to obtain the second side, would be to compute $\max_{\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})} \overline{s}(\tilde{\boldsymbol{x}}, y)$ which needs access to the clean test point. Given the difficulty, we leave computing two-sided guarantees for future work.

A.1. Impelementation Details

We based our implementation on PyTorch (Paszke et al., 2019) and Pytorch Geometric (Fey & Lenssen, 2019). We run all our experiments both on CPU (Intel(R) Xeon(R) Platinum 8368 CPU @ 2.40GHz) and, and on GPU (NVIDIA A100-SXM4-40GB).

B. Faster Evasion-Robustness via Calibration-time Bound

The evasion-robust CP algorithm (see § 5) requires an estimation of the expected smooth score for (i) the true class for all calibration points, (ii) and all classes for each test point. Moreover, for the standard evasion-aware robustness, we need to additionally compute adversarial upper bounds (solutions to Eq. 8) within the threat model for all classes of all test points. This upper bound has a closed-form for continuous data, and an efficient algorithm for binary/discrete data (see § C). Nonetheless, it can be beneficial to reduce the overall runtime. Let t_{bound} be the time complexity for the upper bound computation for a single $(\boldsymbol{x}, y)$ and t_{MC} be the time complexity of approximating the expected smooth score with M Monte-Carlo samples. With n calibration points and c classes, we need $\mathcal{O}(n \times t_{\text{MC}})$ time for calibration, including the quantile computation. Then, for each test point we need $\mathcal{O}(c \times t_{\text{MC}} \times t_{\text{bound}})$ time.

We define a computationally more efficient and robust alternative built upon Prop. 5.1 in which we offload the computational overhead from the test set to the calibration set. Prop. 5.1 gives a worst-case coverage lower bound for vanilla CP – even if we evaluate vanilla CP with smooth (but not upper bounded) scores. Alternatively, we can find a conservative quantile that results in a certified $1 - \alpha$ coverage probability for the worst case input. We call this approach the faster evasion method.

This method of producing prediction sets significantly reduces the computation in two ways: (i) instead of test points (which are larger in number), we compute the upper bounds on calibration points, (ii) instead of computing the upper bound for all classes, we only compute it for the true class. Thus, we need $\mathcal{O}(n \times t_{\text{MC}} \times t_{\text{bound}})$ for calibration, and $\mathcal{O}(c \times t_{\text{MC}})$ for each test point. In practical scenarios where the test set (during deployment) is larger than the calibration set, the computational savings of the faster approach become significant, especially for tasks with a large number of classes (e.g. `ImageNet` with 1000 classes). As shown in Table 3 we gain a significant speed up (more than 3X) on `CIFAR-10` with 204 calibration points and just 100 test points. Here we have gains despite using a relatively tiny test set (even smaller than the calibration set) since we have $c = 10$ classes. Similar, and even better speed-ups can be achieved for datasets with a larger test set and larger number of classes.

Table 3. Run-time comparison between test-time (slower) and calibration-time (faster) upper bound computation. The result is for `CIFAR-10` with 10^4 number of Monte Carlo samples. Here, m is the number of test samples.

	Time (seconds)		No. Datapoints
Runtime	Standard Evasion Robust Sets	Faster Evasion Robust Sets	
Calibration	0.15 $\mathcal{O}(n \times t_{\text{MC}})$	0.79 $\mathcal{O}(n \times t_{\text{MC}} \times t_{\text{b}})$	204
Testing	2.93 $\mathcal{O}(m \times c \times t_{\text{MC}} \times t_{\text{b}})$	0.15 $\mathcal{O}(m \times c \times t_{\text{MC}})$	100
Total	3.08	0.94	

C. Technical Details On Randomized Smoothing

RSCP uses the closed-form solution to Eq. 7 as an upperbound on the score function within the L_2 perturbation radius (details are in § F). The same equation can be used to address other perturbation schemes (e.g. perturbations for sparse data). We use the results from Bojchevski et al. (2020) to find extend RSCP to sparse and discrete data and use it as a baseline.

To apply randomized smoothing we need to define a smoothing scheme $\xi(\cdot)$ – a probabilistic function that adds random noise to the input. Given any score function s, we define $\hat{s}(\boldsymbol{x}, y) = \mathbb{E}[s(\xi(\boldsymbol{x}), y)]$. Now $\Pr[\xi(\boldsymbol{x}) = \boldsymbol{z}]$ is the probability of

visiting some $\boldsymbol{z}$ in the domain by smoothing from $\boldsymbol{x}$. For a continuous data we use Gaussian smoothing where $\xi(\boldsymbol{x}) = \boldsymbol{x} + \delta$ with $\delta \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ coming from an isotropic Gaussian distribution with zero mean and variance σ. We can compute the adversarial upper bounds using the closed-form expressions from Kumar et al. (2020) (see e.g. Eq. 10 in § 4).

For binary data, following Bojchevski et al. (2020), we use the following smoothing function:

$$\Pr\left[\xi(\boldsymbol{x})[i] \neq \boldsymbol{x}[i]\right] = p_{\boldsymbol{x}[i]} \tag{15}$$

This means that ξ toggles each 1-bit of $\boldsymbol{x}$ with probability p_1 and each 0-bit with p_0. This distinction allows us to preserve sparsity by specifying a lower p_0. Setting $p_1 = p_0 = p$ we have the special case of flipping each bit with the same probability p. Similarly Bojchevski et al. (2020) generalizes the binary case to the discrete case. Assuming that $\boldsymbol{x} \in \mathcal{X}_K = \{0, 1, \ldots, K\}^d$ the sparsity aware randomization scheme is defined as

$$\Pr\left[\xi(\boldsymbol{x})_i = k\right] = \begin{cases} \left(\frac{p_0}{K-1}\right)^{(\boldsymbol{x}[i] \neq k)} (1-p_0)^{(\boldsymbol{x}[i]=k)} & \boldsymbol{x}[i] = 0 \\ \left(\frac{p_1}{K-1}\right)^{(\boldsymbol{x}[i] \neq k)} (1-p_1)^{(\boldsymbol{x}[i]=k)} & \boldsymbol{x}[i] \neq 0 \end{cases} \tag{16}$$

that flips any zero bit with probability p_0 and any non-zero bit with p_1 to any other $(K-1)$ possible value.

For the baseline bound we can rewrite Eq. 7 as a linear program by partitioning the input space $\mathcal{X}$ into regions of constant likelihood ratio (Lee et al., 2019). Let $\mathcal{X} = \bigcup_i \mathcal{R}_i$ and $\mathcal{R}_i \bigcap \mathcal{R}_j = \emptyset$ be a partitioning into disjoint regions of constant likelihood ratio such that for ever $\boldsymbol{z} \in \mathcal{R}_i$ it holds $\frac{\Pr[\xi(\boldsymbol{x})=\boldsymbol{z}]}{\Pr[\xi(\tilde{\boldsymbol{x}})=\boldsymbol{z}]} = c_i$ for some constant c_i. Let $t_i = \Pr\left[\xi(\boldsymbol{x}) \in \mathcal{R}_i\right]$ and $\tilde{t}_i = \Pr\left[\xi(\tilde{\boldsymbol{x}}) \in \mathcal{R}_i\right]$ for for each region $\mathcal{R}_i$. Then Eq. 7 is equivalent to:

$$\max_{\boldsymbol{h}} \boldsymbol{h}^T \tilde{\boldsymbol{t}} \quad \text{s.t.} \quad \boldsymbol{h}^T \boldsymbol{t} = p, \quad 0 \leq \boldsymbol{h} \leq 1 \tag{17}$$

where $\boldsymbol{h} \in [0,1]^I$ is the vector that we are optimizing over corresponding to the score function $h \in \mathcal{H}$, $\boldsymbol{t}$ and $\tilde{\boldsymbol{t}}$ are the vectors with t_i and $\tilde{t}_i$ as elements, and $I = r_a + r_d + 1$ is the number of regions. Note, that by replacing the constraint with $a \leq \boldsymbol{h} \leq b$ we can handle score functions that are bounded in $[a, b]$. The exact solution to this LP can be easily obtained with a simple algorithm. We visit each region in increasing order w.r.t. c_i where

$$c_i = \left[\frac{p_0}{1-p_1}\right]^{i-r_d} \left[\frac{p_1}{1-p_0}\right]^{i-r_a} \tag{18}$$

and assign $h_i = 1$ for all regions $\mathcal{R}_i$ until the budget constraint is met, and $h_i = 0$ for the remaining regions, with the exception of the region in between where h_i is a value between 0 and 1 such that the equality constraint is exactly met. Since the likelihood ratios c_i are monotonic in i, the regions are automatically sorted so the solution to the LP can be obtain in linear $O(I)$ time. See Bojchevski et al. (2020) for more details and the pseudo-code.

For the CDF-based bound we can similarly rewrite Eq. 8 as the following linear program:

$$\max_{\boldsymbol{H}} b_m - \boldsymbol{H}\tilde{\boldsymbol{t}}\boldsymbol{d} \quad \text{s.t.} \quad \boldsymbol{H}\boldsymbol{t} = \boldsymbol{p}, \quad 0 \leq \boldsymbol{H} \leq 1 \tag{19}$$

where $\boldsymbol{H} \in [0,1]^{(m-1)\times I}$ is the matrix that we are optimizing over with H_{ji} being the score that we assign to the j-th bin and the i-th region, $\boldsymbol{d}$ is the vector of bin widths such that $d_j = b_j - b_{j-1}$, and $\boldsymbol{p}$ is a vector where $p_i = \Pr\left[s(\xi(\boldsymbol{x}), y) \leq b_i\right]$. Intuitively, for each bin and each region the worst-case score function $h \in \mathcal{H}$ assigns the same score to all $\boldsymbol{z}$ in that region since the likelihood ratio is constant. As before we have a simple algorithm to obtain the exact solution to this LP. Observe that Eq. 19 can be decomposed into $m-1$ separate LPs similar to Eq. 17 which can be solved in parallel using the same algorithm as above. The reason is that there is no interaction between the different bins (different rows of $\boldsymbol{H}$) in neither the constraint nor the objective function. Therefore, the solution can be obtain in $O(m \times I)$ with serial computation and $O(I)$ with parallel computation.

Tightness. All four bounds are tight, i.e. cannot be improved unless we make additional assumption or provide additional constraints. The reason is that there exists a base score function s such that when relaxing to $h \in \mathcal{H}$ we get an equality in Eq. 6. See Kumar et al. (2020) for a discussion of why the two Gaussian bounds are tight when certifying the confidence of a classifier and observe that their analysis immediately applies to our score functions. Similarly, the two discrete bounds are tight since there exists an s for which we obtain equality. The s can be constructed using the optimal $\boldsymbol{h}^*$ from the problem in Eq. 17 and similarly for Eq. 19.

D. Supplementary to Theoretical Support

D.1. Proofs

Proof of Prop. 3.1. Given the exchangeability of $(\boldsymbol{x}_{n+1}, y_{n+1})$ with the calibration set, Eq. 1 holds for the clean point. Since $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$ we have $\forall y_i : \overline{s}(\tilde{\boldsymbol{x}}_{n+1}, y_i) \geq s(\boldsymbol{x}_{n+1}, y_i)$. By the definition of CP for any label y_i we have

$$y_i \in \mathcal{C}(\boldsymbol{x}_{n+1}) \Rightarrow s(\boldsymbol{x}_{n+1}, y_i) \geq q_\alpha$$
$$\boldsymbol{x}_{n+1} \in \mathcal{B}(\tilde{\boldsymbol{x}}_{n+1}) \Rightarrow \overline{s}(\tilde{\boldsymbol{x}}_{n+1}, y_i) \geq s(\boldsymbol{x}_{n+1}, y_i) \geq q_\alpha \Rightarrow \mathcal{C}(\boldsymbol{x}_{n+1}) \subseteq \overline{\mathcal{C}}(\tilde{\boldsymbol{x}})$$

Which clearly implies that $\Pr\left[y_{n+1} \in \overline{\mathcal{C}}(\tilde{\boldsymbol{x}})\right] \geq \Pr\left[y_{n+1} \in \mathcal{C}(\boldsymbol{x})\right] \geq 1 - \alpha$. □

Proof of Prop. 3.2. By definition $\mathcal{D}_{\text{cal}} \in \mathbb{B}_{k,\mathcal{B}}(\mathcal{D}_{\text{cal}})$; therefore q_α is a feasible solution to Eq. 3 and we have $\underline{q_\alpha} \leq q_\alpha$. It follows that $\mathcal{C}_\alpha(\boldsymbol{x}) \subseteq \overline{\mathcal{C}}_\alpha(\boldsymbol{x})$ where $\mathcal{C}_\alpha(\boldsymbol{x}) = \{y_i : s(\boldsymbol{x}, y_i) \geq q_\alpha\}$ and $\overline{\mathcal{C}}_\alpha(\boldsymbol{x}) = \left\{y_i : s(\boldsymbol{x}, y_i) \geq \underline{q_\alpha}\right\}$. Since $\Pr\left[y_{n+1} \in \mathcal{C}_\alpha(\boldsymbol{x})\right] \geq 1 - \alpha$ due to exchangeability it follows that $\Pr\left[y_{n+1} \in \overline{\mathcal{C}}_\alpha(\boldsymbol{x})\right] \geq 1 - \alpha$. In summary the following chain of inequlities hold:

$$\forall y_i : y_i \in C(\boldsymbol{x}_{n+1}) \Rightarrow s(\boldsymbol{x}_{n+1}, y_i) \geq q_\alpha$$
$$\underline{q_\alpha} \leq q_\alpha \Rightarrow s(\boldsymbol{x}_{n+1}, y_i) \geq q_\alpha \geq \underline{q_\alpha} \Rightarrow y_i \in \overline{\mathcal{C}}(\boldsymbol{x}_{n+1})$$

□

Proof of Prop. 5.1. Setting $\underline{q_\alpha} = \text{Quant}\left(\alpha; \{\underline{s}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}\}\right)$ we have:

$$\begin{aligned} \Pr\left[\hat{s}(\tilde{\boldsymbol{x}}_{n+1}, y_{n+1}) \geq \underline{q_\alpha}\right] &\geq \Pr\left[\underline{s}(\boldsymbol{x}_{n+1}, y_{n+1}) \geq \underline{q_\alpha}\right] && \text{Lower bound within the threat model} \\ &\geq 1 - \alpha && \text{Exchangeability between lower bounds} \end{aligned}$$

Alternatively, the vanilla CP is calibrated with quantile $q_\alpha = \text{Quant}\left(\alpha; \{\hat{s}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}\}\right)$. The probability of a given potentially perturbed $\tilde{\boldsymbol{x}}_{n+1}$ being covered is:

$$\begin{aligned} \Pr\left[y_{n+1} \in \mathcal{C}_\alpha(\tilde{\boldsymbol{x}}_{n+1})\right] &= \Pr\left[\hat{s}(\tilde{\boldsymbol{x}}_{n+1}, y_{n+1}) \geq q_\alpha\right] && \text{Definition of CP} \\ &\geq \Pr\left[\underline{s}(\boldsymbol{x}_{n+1}, y_{n+1}) \geq q_\alpha\right] && \text{Lower bound within the threat model} \end{aligned}$$

Let $\beta = \text{Quant}^{-1}\left(q_\alpha; \{\underline{s}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}\}\right)$. If $\underline{s}$ is computed symmetrically – indices are invariant to $\underline{s}$, then $\underline{s}(\boldsymbol{x}_{n+1})$ and $\{\underline{s}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}\}$ are exchangeable. Hence, via quantile lemma we have:

$$\Pr\left[\underline{s}(\boldsymbol{x}_{n+1}, y_{n+1}) \geq q_\alpha\right] \geq 1 - \beta$$

□

Proof of Prop. 6.1. Since for each calibration point $\underline{s}_{\text{cdf}+}(\boldsymbol{x}_i, y_i) \leq \underline{s}_{\text{cdf}}(\boldsymbol{x}_i, y_i)$ has at most $\frac{\eta}{2|\mathcal{D}_{\text{cal}}|}$ failure probability following holds with $1 - \eta/2$ probability via the union bound:

$$\underline{q}_{\alpha+} := \text{Quant}\left(\alpha - \eta; \left\{\underline{s}_{\text{cdf}+}(\boldsymbol{x}_i, y_i)\right\}_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}}\right) \leq \underline{q_\alpha} := \text{Quant}\left(\alpha - \eta/2; \left\{\underline{s}_{\text{cdf}}(\boldsymbol{x}_i, y_i)\right\}_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}}\right)$$

This is because every element $\underline{s}_{\text{cdf}+}(\boldsymbol{x}_i, y_i)$ in the first set is lower than the corresponding element in the other set. Now given the new test datapoint $\tilde{\boldsymbol{x}}_{n+1}$, the new calibration scores $\{\underline{s}_{\text{cdf}}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}\}$ and $\underline{s}_{\text{cdf}}(\boldsymbol{x}_{n+1}, y_{n+1})$ are exchangeable, as a result for the clean corresponding point $\boldsymbol{x}_{n+1}$ we have $\Pr\left[\underline{s}_{\text{cdf}}(\boldsymbol{x}_{n+1}, y_{n+1}) \geq \underline{q_\alpha}\right] \geq 1 - \alpha + \eta$. Therefore we have the following chain of inequalities:

$$\underline{q}_{\alpha+} \underset{1-\eta/2}{\leq} \underline{q_\alpha} \underset{1-\alpha+\eta}{\leq} \underline{s}_{\text{cdf}}(\boldsymbol{x}_{n+1}, y_{n+1}) \leq \hat{s}(\tilde{\boldsymbol{x}}_{n+1}, y_{n+1}) \underset{1-\eta/2}{\leq} \hat{s}_+(\boldsymbol{x}_{n+1}, y_{n+1})$$

summing up the probability of each inequality we have $\Pr\left[y_{n+1} \in \overline{\mathcal{C}}_{\alpha+}\right] = \Pr\left[\hat{s}_+(\boldsymbol{x}_{n+1}, y_{n+1}) \geq \underline{q}_{\alpha+}\right] \geq 1 - \alpha$ □

Proof of Prop. 6.2. Since Eq. 24 (and Eq. 3) is a minimization problem, dropping $a_i \leq \overline{s}_i$ does not change the optimal solution. For each calibration point, we have:

$$s_{\text{mc}}(\boldsymbol{x}_i, y_i) \underset{1-\eta/2|\mathcal{D}_{\text{cal}}|}{\geq} \hat{s}(\boldsymbol{x}_i, y_i) - \epsilon_{\text{hoef}} \geq \underline{s}_{\text{cdf}}(\tilde{\boldsymbol{x}}_i, y_i) - \epsilon_{\text{hoef}} \underset{1-\eta/2|\mathcal{D}_{\text{cal}}|}{\geq} \underline{s}_{\text{cdf+}}(\tilde{\boldsymbol{x}}_i, y_i) - \epsilon_{\text{hoef}}$$

Thus $\underline{s}_{\text{cdf+}}(\tilde{\boldsymbol{x}}_i, y_i) - \epsilon_{\text{hoef}} \leq s_{\text{mc}}(\boldsymbol{x}_i, y_i)$ holds with $1-\eta$ probability for all i via union bound. This follows that $\underline{q}_{\alpha+} \leq q_{\alpha,\text{mc}}$ where $q_{\alpha,\text{mc}}$ is the α-quantile of MC scores for clean calibration set. Therefore by exchangeability, we have $\Pr\left[s_{\text{mc}}(\boldsymbol{x}_{n+1}, y_{n+1}) \geq q_{\alpha,\text{mc}}\right] \geq 1-\alpha = 1-\alpha'+\eta$. Finally

$$\Pr\left[s_{\text{mc}}(\boldsymbol{x}_{n+1}, y_{n+1}) \geq \underline{q}_{\alpha+}\right] \underset{1-\eta}{\geq} \Pr\left[s_{\text{mc}}(\boldsymbol{x}_{n+1}, y_{n+1}) \geq q_{\alpha,\text{mc}}\right] \geq 1-\alpha'+\eta$$

Therefore, $\Pr\left[s_{\text{mc}}(\boldsymbol{x}_{n+1}, y_{n+1}) \geq \underline{q}_{\alpha+}\right] \geq 1-\alpha'$. □

D.2. Details on l_2 CDF bounds

Rephrase from Kumar et al. (2020). The upper bound in Eq. 10 is a rephrasing of Theorem 2 from Kumar et al. (2020). In the original version the bins are defined as $a < c_1 \leq c_2 \leq \cdots \leq c_n < b$. For the for a score function s and (clean) input $\boldsymbol{x}$ the statistics p_{c_j} is defined as

$$p_{c_j} = \Pr_{\boldsymbol{\delta}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})}[s(\boldsymbol{x}+\boldsymbol{\delta}, y) \geq c_i]$$

Here we correct for finite sample estimation via Dvoretzky–Kiefer–Wolfowitz inequality. With the detailed discussion on Monte-Carlo sample correction in § E, we assume that the statistics are computed with the error correction. For the adversarial point $\tilde{\boldsymbol{x}} \in \mathcal{B}_r(\boldsymbol{x})$ we have the following upper bound:

$$\hat{s}(\tilde{\boldsymbol{x}}, y) \leq c_1 + (b-c_n)\Phi_\sigma\left(\Phi_\sigma^{-1}(p_{c_n}) + r\right) + \sum_{j=1}^{n-1}(c_{j+1}-c_j)\Phi_\sigma\left(\Phi_\sigma^{-1}(p_{c_j}) + r\right) \tag{20}$$

In Eq. 10 we rewrote the same inequality with a simpler notation. Here we show that the two inequalities are the same. Our bins are indexed as $a = b_1 < b_2 \leq b_3 \leq \cdots \leq b_{m-1} < b_m = b$; therefore for the same number of bins ($n = m-2$), there is an index mapping as $\forall 1 \leq i < m : c_{i-1} = b_i$. Rewriting Eq. 20 with the new bins, we have:

$$\begin{aligned}\hat{s}(\tilde{\boldsymbol{x}}, y) &\leq b_2 + (b_m - b_{m-1})\Phi_\sigma\left(\Phi_\sigma^{-1}(p_{b_{m-1}}) + r\right) + \sum_{j=1}^{m-3}(b_{j+2}-b_{j+1})\Phi_\sigma\left(\Phi_\sigma^{-1}(p_{b_{j+1}}) + r\right) \\ &= b_2 + \sum_{j=1}^{m-2}(b_{j+2}-b_{j+1})\Phi_\sigma\left(\Phi_\sigma^{-1}(p_{b_{j+1}}) + r\right) = b_2 + \sum_{j=2}^{m-1}(b_{j+1}-b_j)\Phi_\sigma\left(\Phi_\sigma^{-1}(p_{b_j}) + r\right)\end{aligned}$$

We write the upper bound in terms of CDF function where $p_j = \Pr_{\boldsymbol{\delta}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})}[s(\boldsymbol{x}+\boldsymbol{\delta}, y) \leq b_j]$. We use two properties from Gaussian distribution (i) for the CDF function it holds that $\Phi_\sigma(-z) = 1-\Phi_\sigma(z)$ (ii) for the quantile (inverse CDF) function it holds that $\Phi_\sigma^{-1}(1-z) = -\Phi_\sigma^{-1}(z)$. Hence, we have

$$\begin{aligned}p_{b_j} = 1 - p_j \Rightarrow \Phi_\sigma\left(\Phi_\sigma^{-1}(p_{b_j}) + r\right) &= \Phi_\sigma\left(\Phi_\sigma^{-1}(1-p_j) + r\right) \\ &= \Phi_\sigma\left(-\Phi_\sigma^{-1}(p_j) + r\right) \\ &= \Phi_\sigma\left(-[\Phi_\sigma^{-1}(p_j) - r]\right) \\ &= 1 - \Phi_\sigma\left(\Phi_\sigma^{-1}(p_j) - r\right)\end{aligned}$$

It follows

$$\begin{aligned} b_2 + \sum_{j=2}^{m-1}(b_{j+1}-b_j)\Phi_\sigma\left(\Phi_\sigma^{-1}(p_{b_j})+r\right) &= b_2 + \sum_{j=2}^{m-1}(b_{j+1}-b_j)\left[1-\Phi_\sigma\left(\Phi_\sigma^{-1}(p_j)-r\right)\right] \\ &= b_2 + \sum_{j=2}^{m-1}(b_{j+1}-b_j) - \sum_{j=2}^{m-1}(b_{j+1}-b_j)\Phi_\sigma\left(\Phi_\sigma^{-1}(p_j)-r\right) \\ &= b_m - \sum_{j=2}^{m-1}(b_{j+1}-b_j)\Phi_\sigma\left(\Phi_\sigma^{-1}(p_j)-r\right) \end{aligned}$$

Intuitively, with a fixed set of bins, the mass of each bin can be bounded within $\mathcal{B}_r(\boldsymbol{x})$ independently (the bound for each bin is similar to the mean bound). Therefore for a discrete empirical CDF of scores around $\boldsymbol{x}$, first we find a worst-case upper bound CDF, then we bound the mean via the Anderson inequality (Eq. 9) given the worst-case CDF.

Lower bounds within $\mathcal{B}_r(\cdot)$. Similar to the mean upper bound from the Anderson inequality (Eq. 9), the mean can be lower bounded as:

$$\mathbb{E}[h(\boldsymbol{x})] \geq \sum_{j=2}^{m} b_{j-1}\cdot[F_h(b_j)-F_h(b_{j-1})] = b_{m-1} - \sum_{j=2}^{m-1} F_h(b_j)\cdot(b_j-b_{j-1}) \tag{21}$$

The lower and upper bounds are intuitive as they assume every point within each bin $[b_{j-1}, b_j)$ is equal to b_{j-1} for lower and b_j for the upper bound. The rest is just computing the average based on the relative frequency $F_h(b_j) - F_h(b_{j-1})$. With that the lower bound version of Eq. 8 is

$$\hat{s}(\tilde{\boldsymbol{x}}, y) \geq \underline{s}_{\text{cdf}}(\boldsymbol{x}, y) = b_{m-1} - \sum_{j=2}^{m-1} \Phi_\sigma\left(\Phi_\sigma^{-1}(p_j)+r\right)\cdot(b_j-b_{j-1}) \tag{22}$$

Here we derive the equality in Eq. 9 – namely the following;

$$\sum_{j=2}^{m} b_j\cdot[(F_h(b_j)-F_h(b_{j-1})] = b_m - \sum_{j=2}^{m-1} F_h(b_j)\cdot(b_{j+1}-b_j)$$

The lower bound follows a similar way to derive. We have

$$\begin{aligned} &\sum_{j=2}^{m} b_j\cdot[(F_h(b_j)-F_h(b_{j-1})] \\ &\quad= b_2\cdot[(F_h(b_2)-F_h(b_1)] + b_3\cdot[(F_h(b_3)-F_h(b_2)] + \cdots + b_m\cdot[(F_h(b_m)-F_h(b_{m-1})] \\ &\quad= -b_2\cdot F_h(b_1) + [b_2\cdot F_h(b_2) - b_3\cdot F_h(b_2)] + \cdots + [b_{m-1}\cdot F_h(b_{m-1}) - b_m\cdot F_h(b_{m-1})] + b_m\cdot F_h(b_m) \end{aligned}$$

With $F_h(b_1)=0$ and $F_h(b_m)=1$ we have

$$\begin{aligned} &\sum_{j=2}^{m} b_j\cdot[(F_h(b_j)-F_h(b_{j-1})] \\ &\quad= 0 + [-F_h(b_2)\cdot(b_3-b_2)] + \cdots + [-F_h(b_{m-1})(b_m-b_{m-1})] + b_m \\ &\quad= b_m - \sum_{j=2}^{m-1} F_h(b_j)\cdot(b_{j+1}-b_j) \end{aligned}$$

E. Estimating Expectations with Monte-Carlo Sampling

Concentration inequalities. For any random variable z, let $z_1, \ldots, z_m$ be Monte-Carlo samples of z. With $\mathbb{E}_m[z] = \frac{1}{m}\sum_{i=1}^{m} z_i$, we bound the true expectation around the MC-estimate via Hoeffding's inequality. The following holds with

any adjustable $1-\eta$ probability;

$$|\mathbb{E}[z] - \mathbb{E}_m[z]| \leq \sqrt{\frac{\log(\frac{2}{\eta})}{2m}}$$

This bound only accesses to the empirical mean and not the samples. Therefore, in cases where we want to account for the distance of empirical mean, and the true expectation for an unknown variable, we can use this bound. An example of this case is the test-time correction where the upper bound on the mean of the unseen point is computed while the empirical mean is not computable (since there are no samples).

Let σ_m^2 be the variance of the MC samples, then empirical Bernstein inequality produces variance-dependent confidence intervals as following:

$$|\mathbb{E}[z] - \mathbb{E}_m[z]| \leq \sqrt{2\sigma_m^2 \frac{\ln(\frac{4}{\eta})}{m}} + \frac{7\ln(\frac{4}{\eta})}{3(m-1)}$$

Similar to the mean, the empirical CDF is also bounded between an upper and a lower CDF, via the Dvoretzky–Kiefer–Wolfowitz (DKW) inequality (Dvoretzky et al., 1956). Let $F(b_i) = \Pr[z \leq b_i]$ and $F_m(b_i) = \sum_{j=1}^{m} \mathbf{1}[z_j \leq b_i]$,

$$|F(b_i) - F_m(b_i)| \leq \sqrt{\frac{\log(\frac{2}{\eta})}{2m}}$$

The above inequality holds simultaneously for all b_i.

For Eq. 7 we use the Bernstein inequality as is has shown a better empirical result compare to Hoeffding's inequality. For Eq. 8 we use the DKW inequality to find confidence intervals the empirical CDF.

Error correction in Eq. 7 and Eq. 8. To find the upper (or lower) bound in Eq. 7, we need to estimate the mean of the smooth score around the input $\boldsymbol{x}$. We use the mean corrected with the Bernstein confidence interval. For the upper bound problem, we use the upper end of the interval since it is more conservative. The same logic follows for the lower bound.

For Eq. 8 we use the Dvoretzky–Kiefer–Wolfowitz inequality to find an upper (or lower) CDF. Since in the Eq. 9 the CDF is added with a negative sign, the lower endpoint of the confidence interval should be used to find a conservative upper bound.

Empirically, Bernstein's confidence intervals are tighter than Hoeffding's intervals. Therefore we only use the Hoeffding error anytime we need a correction without having access to the variance.

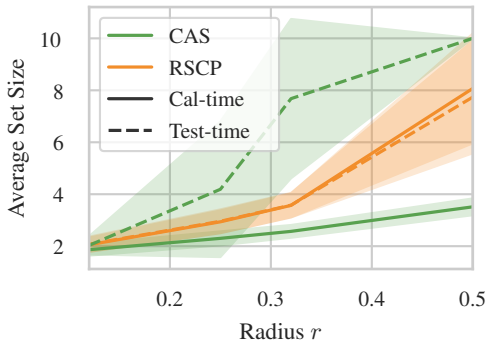

Figure 6. Comparison of CAS and RSCP for faster (calibration-time) and test-time error correction.

Test-time correction (Yan et al., 2024). The MC-sampled smooth score does not break the exchangeability since this estimation is permutation invariant. This means that given the clean input $\boldsymbol{x}_{n+1}$ the estimated scores are exchangeable and the guarantee is valid without any error correction. However, given $\tilde{\boldsymbol{x}}$, CAS and RSCP find bounds on the true mean. Given $\tilde{\boldsymbol{x}}$, we compute $\overline{s}_+$ via solving either Eq. 7 (RSCP) or Eq. 8 (CAS) with the error corrected estimate. For both methods, the following holds:

$$q_{\alpha,\text{mc}} \underset{1-\alpha}{\leq} \hat{s}_{mc}(\boldsymbol{x}_{n+1}, y_{n+1}) \underset{1-\eta_1}{\leq} \hat{s}(\boldsymbol{x}_{n+1}, y_{n+1}) + \epsilon_{\text{hoef}} \leq \overline{s}(\tilde{\boldsymbol{x}}_{n+1}, y_{n+1}) + \epsilon_{\text{hoef}} \underset{1-\eta_2}{\leq} \overline{s}_+(\tilde{\boldsymbol{x}}_{n+1}, y_{n+1}) + \epsilon_{\text{hoef}}$$

By setting $\alpha' = \alpha + \eta_1 + \eta_2$ we have a valid CP guarantee with certified $1-\alpha$ probability.

Calibration-time vs test-time correction. As shown in Fig. 6, CAS benefits significantly from calibration-time robustness. The reason is that the CDF bound (Eq. 8) performs significantly better than the mean bound (Eq. 7) when the score

distribution is more spread out. For distributions concentrated around each endpoint of the domain, the CDF has a high slope at the endpoint and is almost flat elsewhere. While using DKW inequality, a large penalty is added to the distribution resulting in larger CDF intervals. Meanwhile, in these distributions, the mean bound can benefit from Bernstein inequality which due to the low variance performs even better. In calibration-time robustness, we find the lower bound for true scores (which are often more spread) while in test-time unlikely classes that have scores concentrated to 0 are bounded by large value (due to DKW for concentrated scores) which directly affects the set size. In addition, Yan et al. (2024) adds a Hoeffding error to the unseen clean score, where in our method we bound the estimation of the input which can use the Bernstein error. Since we are free to choose between test-time and calibration-time correction, and RSCP has equal performance for both, we argue that we should use calibration-time correction as a default. For Fig. 4 we choose the best performance of each method in either calibration- or test-time robustness with error correction.

F. Details on RSCP

F.1. Equivalence Between RSCP and our Gaussian Baseline Bound

For a given score function $s : \mathcal{X} \times \mathcal{Y} \mapsto [0, 1]$ on continuous inputs, Gendler et al. (2021) define the new scoring function as follows:

$$s_{\text{rscp}}(x, y) = \Phi^{-1}(\mathbb{E}[s(\xi(x), y)]) \tag{23}$$

RSCP computes the α-quantile q_α of the new calibration scores (Eq. 23) and compares each score with the modified threshold[4] $\underline{q_\alpha} = q_\alpha - r/\sigma$, where r is the radius of the l_2 ball from the threat model, and σ is the scale of the smoothing distribution. We can equivalently add an additional r/σ term to test scores instead and compare the augmented score with unchanged q_α. Using $\Phi_\sigma^{-1}(p) = \sigma\Phi^{-1}(p)$ as a property of the inverse CDF function of the Gaussian distribution, we have

$$\Phi^{-1}\left(\mathbb{E}[s(\xi_\sigma(\boldsymbol{x}), y)]\right) \leq \Phi^{-1}\left(\mathbb{E}[s(\xi_\sigma(\tilde{\boldsymbol{x}}), y)]\right) + \frac{r}{\sigma} \Rightarrow \Phi_\sigma^{-1}\left(\mathbb{E}[s(\xi_\sigma(\boldsymbol{x}), y)]\right) \leq \Phi_\sigma^{-1}\left(\mathbb{E}[s(\xi_\sigma(\tilde{\boldsymbol{x}}), y)]\right) + r$$

Since the CDF is a monotonically increasing function we apply Φ_σ on both sides of the inequality:

$$\begin{aligned}\Phi_\sigma\left(\Phi_\sigma^{-1}\left(\mathbb{E}[s(\xi_\sigma(\boldsymbol{x}), y)]\right)\right) &\leq \Phi_\sigma\left(\Phi_\sigma^{-1}\left(\mathbb{E}[s(\xi_\sigma(\tilde{\boldsymbol{x}}), y)]\right) + r\right) \\ \Rightarrow \mathbb{E}[s(\xi_\sigma(\tilde{\boldsymbol{x}}), y) &\leq \Phi_\sigma\left(\Phi_\sigma^{-1}\left(\mathbb{E}[s(\xi_\sigma(\tilde{\boldsymbol{x}}), y)]\right) + r\right)\end{aligned}$$

Substituting $p = \mathbb{E}[s(\xi(\tilde{\boldsymbol{x}}), y)]$ we see that this is equivalent to the Gaussian $\overline{s}_{\text{mean}}$ upper-bound defined in § 4.

F.2. Comparison with Cauchois et al. (2020)

Cauchois et al. (2020) derive robust prediction sets when the f-divergence between the test distribution and the calibration distribution of the non-conformity scores is bounded by a fixed value ρ. We can connect their approach to our definition of adversarial robustness using the results from Dvijotham et al. (2020). Specifically, we can rewrite the optimization problem $\max_{\tilde{\boldsymbol{x}} \in \mathcal{B}_r(\boldsymbol{x})} \mathbb{E}[s(\xi(\tilde{\boldsymbol{x}}, y)]$ over the ball $\mathcal{B}_r(\boldsymbol{x})$ to the optimization problem $\max_{\nu \in \mathcal{P}} \mathbb{E}[s(\nu(\boldsymbol{x}, y)]$ over the space of probability measures $\mathcal{P} = \{\xi(\tilde{\boldsymbol{x}}) \mid \boldsymbol{x} \in \mathcal{B}_r(\boldsymbol{x})\}$. Since this set is intractable we can relax the problem using the fact that $\mathcal{P} \subseteq \{D_f(\nu||\xi) \leq \rho_r^f\}$ for an appropriately chosen ρ_r^f where $D_f(\nu||\xi)$ is the f-divergence between the smoothing distribution ν centered at a perturbed example and the smoothing distribution ξ centered at the clean example. See Dvijotham et al. (2020) for a derivation of the optimal ρ_r^f for different different divergence functions f and different smoothing distributions. Thus, for smooth scores there is a direct connection between RSCP, CAS and Cauchois et al. (2020)'s method.

Importantly however, for most choices of f (e.g. the KL divergence) the relaxation results in a looser (though potentially easier to compute) bound. The analysis in Dvijotham et al. (2020) was developed for classification problems but it also directly applies to our setting. They show that we need to use the Hockey- Stick divergences with the right parameters to obtain tight certificates. Specifically, for Gaussian smoothing and an l_2 norm the result is equivalent to the tight certificate from Cohen et al. (2019). Disregarding that Hockey-Stick divergences are harder to estimate in general, it means that in the best case, the approach by Cauchois et al. (2020) can recover the baseline $\overline{s}_{\text{mean}}(\tilde{\boldsymbol{x}}, y)$ which we have shown is looser than our $\overline{s}_{\text{cdf}}(\tilde{\boldsymbol{x}}, y)$.

[4]Originally RSCP shifts the quantile forward $\underline{q_\alpha} = q_\alpha + r/\sigma$ since it is defined with the non-conformity setup where scores lower than the quantile are accepted. Here since we use conformity (agreement) scores and the acceptance criteria is to be larger than q_α we shift the quantile backward. The setups are equivalent via changing the sign of the scores (see § A).

G. Technical Details on Poisoning Certificate

Feature poisoning. The solution of the optimization problem in Eq. 3 is robust to feature poisoning; however, the problem is hard to solve since: (i) we need to optimize over each $\boldsymbol{z}_i$ in $\mathcal{B}(\tilde{\boldsymbol{x}}_i)$, (ii) it involves a quantile computation, (iii) and it has a cardinality constraint as the sum of indicator functions. Therefore, we relax the problem to a MILP which can solve with standard solvers. First, we replace each $\boldsymbol{z}_i \in \mathcal{B}(\tilde{\boldsymbol{x}}_i)$ constraint with a $\underline{s}_i \leq \tilde{s}_i \leq \overline{s}_i$ constraint directly over scores s_i where the lower and upper bounds are computed as in discussed in § 4. This is a sound relaxation and the optimal $\underline{q_\alpha}$ of the relaxed problem is smaller or equal than the $\underline{q_\alpha}$ of the original problem. Then, we introduce $|\mathcal{D}_{\text{cal}}|$ binary variables to compute the α quantile, and additional $|\mathcal{D}_{\text{cal}}|$ binary variables to enforce the perturbation budget. The resulting MILP is:

$$
\begin{aligned}
\underline{q_\alpha} = \quad & \min_{s_i, q} \quad q \\
s.t. \quad & \forall \tilde{s}_i : \underline{s}_i \leq s_i \leq \overline{s}_i \\
& t_i := \mathbf{1}[s_i \leq q], \quad \sum_{i=1}^{n} z_i \leq \lfloor \alpha n \rfloor, \quad \text{and} \quad \sum_{i=1}^{n} (1 - t_i) \leq \lceil (1-\alpha) n \rceil \\
& b_i := \mathbf{1}[s_i \neq \tilde{s}_i], \quad \sum_{i=1}^{n} b_i \leq k
\end{aligned}
\tag{24}
$$

In Eq. 24, the z_i variables indicate whether the calibration point is below or above the α quantile q, and the b_i variables indicate whether the point is perturbed or not. We use the standard big-M technique to translate this into a canonical form which we solve with MOSEK.

Label poisoning. We can directly rewrite Eq. 5 as a MILP without any relaxations. Let $\boldsymbol{S}$ be an $n = |\mathcal{D}_{\text{cal}}|$ by c matrix of scores for each class and each calibration point, where c is the number of classes. We have

$$
\begin{aligned}
\underline{q_\alpha} = \quad & \min_{q, \boldsymbol{C} \in \{0,1\}^{n \times c}} \quad q \\
\text{s.t.} \quad & \boldsymbol{C} \mathbf{1}^{c \times 1} = \mathbf{1}^{n \times 1} \\
& \boldsymbol{r} = (\boldsymbol{C} \odot \boldsymbol{S}) \times \mathbf{1}^{c \times 1} \\
& \sum_{i} \boldsymbol{C}[i, y_i] \geq n - k \\
& z_i := \mathbf{1}[r_i \leq q], \quad \sum_{i=1}^{n} z_i \leq \lfloor \alpha n \rfloor, \quad \text{and} \quad \sum_{i=1}^{n} (1 - z_i) \leq \lceil (1-\alpha) n \rceil
\end{aligned}
\tag{25}
$$

where the binary one-hot matrix $\boldsymbol{C}$ is responsible for selecting one score per calibration point (i.e. one of the c possible labels), $\boldsymbol{r}$ is the resulting set of chosen scores, and the z_i variables implement the quantile as before.

Complexity. Note that while in general, solving MILPs is computationally expensive, since our calibration sets are relatively small, we can still obtain the exact solution in reasonable wall-clock time. We leave it as future work to derive more efficient algorithms for the feature and label poisoning problems.

H. Robustness to Poisoning and Evasion Attacks Combined

In § 3.2 we make CP robust to poisonings (in feature or label domain) by finding a conservative $\hat{q}$ that in the most adverse case of attack (within the defined budget and threat model) the coverage probability remains above $1 - \alpha$. Once this threshold is defined, we can consider the calibrated quantile to safely satisfy the guarantee on the clean test – we can assume that CP was calibrated on clean calibration data. Formally the solution to Eq. 3, and Eq. 5, is a threshold with which the prediction sets constructed for clean $\boldsymbol{x}$ has larger than $1 - \alpha$ coverage probability.

While making CP robust to evasion, we only consider the confidence interval of scores for the clean test point given the potentially perturbed point $\tilde{\boldsymbol{x}}$. This process only involves computing upper bounds on the given test point and hence is independent of the prior robustness to poisoning. In other words, the resulting conservative prediction set includes the prediction set of the clean datapoint $\mathcal{C}(\boldsymbol{x}) \subseteq \overline{\mathcal{C}}(\tilde{\boldsymbol{x}})$.

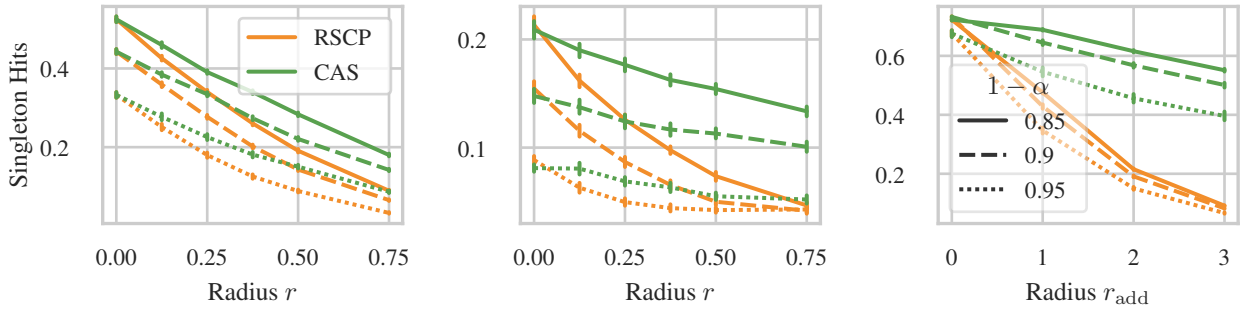

Figure 7. Singleton hit ratio of CAS and RSCP under evasion for (from left to right) `CIFAR-10` with APS, `ImageNet` with TPS, and `Cora` with APS.

This shows that we can make CP robust to poisoning and evasion attacks at the same time. However, this combined robustness comes at the price of comparably larger prediction sets. The robust $\underline{q}$ is less than q_α which allows more labels to be included in the prediction sets. At the same time, for each test point, the upper-bound scores introduce a higher probability for a label to be included in the prediction sets again. So there will be two conservative processes each increasing the chance of accepting a label which increases the expected set size.

I. Time and Space Complexity

Our robust CP approach breaks down into four computations (i) computing the score function, (ii) estimating expectations for randomized smoothing (in practice the MC sampling and computing the confidence intervals), (iii) computing upper-bounds, and (iv) standard CP processes including calibration and constructing prediction sets. Here we omit the time complexity analysis of the model, and with the black-box access, we assume the model's prediction of logits to take $\mathcal{O}(1)$ step. The computation of the conformity score depends on the choice of this function. TPS takes $\mathcal{O}(K)$ (K is the number of classes) to compute the categorical distribution via softmax function. APS score function takes an additional $\mathcal{O}(K)$ steps to sort the class probabilities and compute the summation of confidences (see § 2 for the definition of the score function). This additional sort can become time-consuming for datasets with large number of classes (like `ImageNet`) For simplicity, we call the score function to take t_s steps. Standard CP procedures are calibration and constructing prediction sets. Given n calibration score finding the $1-\alpha$ quantile takes $\mathcal{O}(n)$ steps (median computation) and the prediction sets take $\mathcal{O}(C)$ to be constructed for each test input. All the time complexities are reported w.r.t. serial computation, while with enough number of parallel processing cores, all above computations can be done in relatively lower number of steps.

In the randomized smoothing we need to estimate the expected score function within the smoothing scheme. For that, we use Monte-Carlo sampling which takes $\mathcal{O}(N \times M)$ steps to compute the mean of M Monte-Carlo samples and N is the number of datapoints in total.

With the Monte-Carlo samples each upper- and lower-bound need solving an optimization problem. The optimal value is found via a closed-form solution for Gaussian smoothing. Given S bins for the binary (and discrete) CDF computing this bound takes $\mathcal{O}(S \times R)$ time where R is the number of regions of similar likelihood and we have $R = r_a + r_d + 1$. We refer to the time computation time of the bound as t_b.

As a result, in the evasion setup, we take $\mathcal{O}(NM)$ additional steps for calibration on smooth scores and $\mathcal{O}(MK + K \cdot t_b)$ for constructing the prediction sets. We also proposed a faster way to provide robust prediction sets in § B. For that we compute an upperbound per each calibration datapoint but only for the true class. For any given test point we only compute smooth scores which in total reduces the computation to $\mathcal{O}(MK)$ for test time (per test datapoint), and increases the calibration time complexity to $\mathcal{O}(N \cdot t_b)$. This procedure decreases the number of steps in total. Table 3 compares the runtime of both approaches for a limited number of calibration, and test points.

For poisoning in the feature space, we should first compute the upper and lower bounds for each calibration data which takes $\mathcal{O}(NM + M \cdot t_b)$ steps. Here we just compute the bounds for the true label. We then solve a mixed integer linear programming which is computationally hard. We apply tricks like big-M method to make the problem solvable and enable

the use of standard convex optimization solvers. Similarly for the label poisoning, the problem is hard involving ILP solvers, but here we do not need to compute bounds on scores as the perturbations are in the label domain.

J. Supplementary To Experiments

J.1. Details on the Experiments in the Manuscript

In our core experiment, we utilized a `ResNet-110` model pre-trained on the `CIFAR-10` dataset and a `ResNet-50` model pre-trained on the `ImageNet` dataset. Both models were trained using noisy training by Gaussian data augmentation across various noise variances, as proposed by Lecuyer et al. (2019) and later used by Cohen et al. (2019) for randomized smoothing. Detailed insights into the model training and augmentation processes are elaborated in Cohen et al. (2019); Salman et al. (2019).

For evaluation, we employed an l_2 norm smoothing paradigm and applied various noise levels, identifying the model that delivered optimal performance based on findings from Cohen et al. (2019). On `CIFAR-10` dataset we used a skip parameter and ran the experiments on between 1000 to 2000 samples. Similarly, 500 data points are used from the sampling of every 100-image from the `ImageNet` dataset. Noise variance settings used were $\sigma = 0.25$ for `CIFAR-10` and $\sigma = 0.5$ for `ImageNet`. During the Monte Carlo sampling, each datapoint was processed through 10^4 iterations to calculate the expected probability or mean.

For our experiments on the `Cora-ML` dataset. we utilized a two-layer `GCN` equipped with 64 hidden units. Followed by Bojchevski et al. (2020), our training procedure incorporated randomized perturbations of the node features. Specifically, we used a perturbation addition probability (p_+) of 0.01 and a deletion probability (p_-) of 0.6. For the training process, we employed 20 node labels per class for training and similar number of nodes for validation. We conducted the training over 1,000 epochs. The remaining portion of the dataset was set aside for evaluation purposes.

In our conformal prediction strategy, the split conformal method was adopted. To account for the effect of randomness in calibration set sampling, we reported our result in terms of mean and confidence bounds over 100 calibration samplings.

Moreover, results for adversarial cases are discussed. For these attacks, we employed the projected gradient descent (PGD) attack (Madry et al., 2017), using an alpha value of 0.1 across 40 iterations. The attack outcomes, constrained by L2 norm distance from the original image, are presented for $r = 0.125$.

Singleton hits ratio. This metric quantifies the proportion of correct singleton predictions which can be used without any further post-processing. Similar to the prediction set size, Fig. 7 shows that CAS outperform RSCP on all datasets.

Proportion of Empty, Singleton, and Multi-sets. While we report the average set size (similar to many other studies in CP), a CP method might misleadingly show to be more efficient by returning more empty sets. That is why an alternative metric is to only report the average size of non-empty sets. In Fig. 8 we report the proportion of empty, singleton and multi-prediction sets for various radii. In vanilla CP, as we increase the $1 - \alpha$ guarantee to higher values, CP adds more elements to prediction sets to satisfy the increased coverage guarantee. Since there are almost no empty prediction sets for various α, and r, both effective and average set size are the same.

Different Score Functions. As mentioned in § 2 (and in § A extensively), coverage guarantee in vanilla CP, and robustness methods defined on top (including RSCP, and CAS) are defined agnostic to the score function leaving the freedom of choosing the score based on the domain of application. Here we empirically support this argument. Fig. 9 compares RSCP with CAS applied on TPS and APS score functions. In all scores, and all metrics CAS shows an improved result.

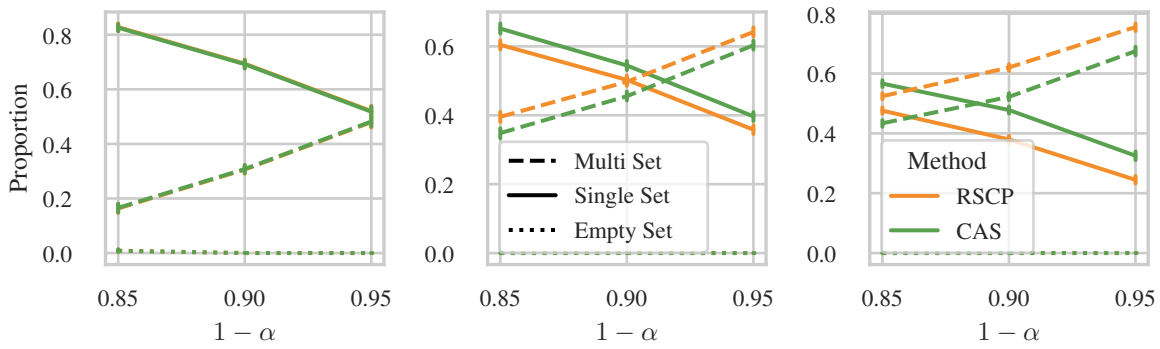

Figure 8. The proportion of singleton, empty and multi-sets for RSCP and CAS across radii (left) $r = 0$, (middle) $r = 0.12$, and (right) $r = 0.25$.

K. Notations and Definition Guide

For a complete guide to all notations used in the paper see Table 4.

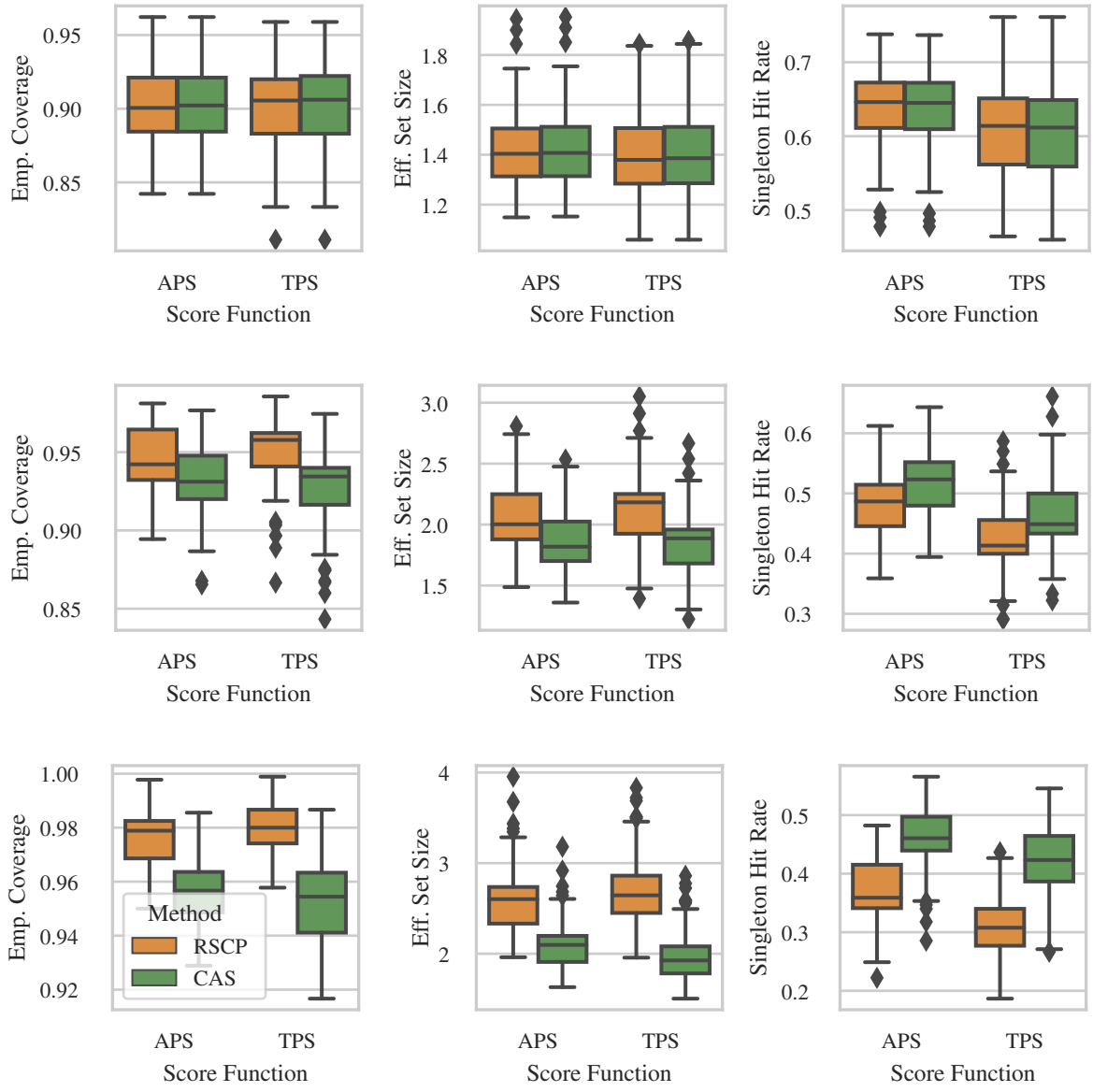

Figure 9. Comparison of RSCP and CAS for smooth APS and TPS score across various radii for (left column) empirical coverage (middle column) set size, and (right column) singleton hits. From upper to lower row results are respectively for $r = 0$, $r = 0.12$, and $r = 0.25$. All results are for `CIFAR-10`, and smoothing with $\sigma = 0.25$

Notation	Desciption
$\boldsymbol{x}$	The clean input
$(\boldsymbol{x}_i, y_i)$	The clean input alongside its true label.
$\mathcal{D}_{\text{cal}}$	Clean calibration set. A set of labeled datapoints which its labels are unseen by the model during the training. Precisely, the conformity score (e.g. model softmax) is exchangeable between elements of this set and the test set.
$\tilde{\boldsymbol{x}}$	The input perturbed by the adversary
$(\boldsymbol{x}_i, \tilde{y}_i)$	The clean input alongside a label that is potentially flipped by the adversary.
$\tilde{\mathcal{D}}_{\text{cal}}$	The poisoned calibration set. Here the adversary has returned a set, given clean $\mathcal{D}_{\text{cal}}$, where under threat model either features are perturbed, or labels are fliped (or both).
$\mathcal{B}(\cdot)$	Point-level threat model: The set of all allowed perturbations w.r.t. the clean point; e.g. all points that are closer than r in l_2 distance
$\mathbb{B}_{k,\mathcal{B}}(\mathcal{D})$	Set-level threat model: The set of all allowed perturbations changing an input set; e.g. CP's calibration set. As an example the set of all perturbed sets where the adversary has changed at most k points within a point-level threat model.
$s(\cdot, \cdot)$	Conformity score function originally defined for vanilla CP
$s_{\text{rscp}}(\cdot, \cdot)$	Scores defined by Gendler et al. (2021).
$\overline{s}(\cdot, \cdot), \underline{s}(\cdot, \cdot)$	Upper- and lower-bounds for given score function $s(\cdot, \cdot)$ within the specified threat model.
q_α	Conformal quantile computed by CP on the clean calibration set with nominal coverage probability $1 - \alpha$
$\tilde{q}_\alpha$	Adversarial conformal quantile; this is a quantile of the calibration set that is poisoned by the adversary. It is expected that this quantile results in lower coverage compared to q_α.
$\underline{q_\alpha}$	Conservative lower-bound for q_α; This is computed by the defender to return robustness prediction sets.
$\mathcal{C}_\alpha(\cdot)$	Prediction set of vanilla CP with $1 - \alpha$ nominal coverage.
$\overline{\mathcal{C}}_\alpha(\cdot)$	Prediction set of robust CP with $1 - \alpha$ nominal coverage. Dependent on the attack scenario (evasion or poisoning), this set is robust to the perturbations within the threat model.
$\hat{s}(\cdot, \cdot)$	The smooth score for the input $\boldsymbol{x}$. This score is the expectation of the score under a predefined randomized smoothing framework.
$\overline{s}_{\text{mean}}(\cdot, \cdot)$	The upperbound score calculated by solving Eq. 7. This problem only has the mean similarity constraint.
$\overline{s}_{\text{cdf}}(\cdot, \cdot)$	The upperbound score calculated by solving Eq. 8. This problem only has the CDF similarity constraint.

Table 4. Table of notations used in the paper.