Verifiably Robust Conformal Prediction

Linus Jeary[*]
Department of Informatics
King's College London, UK
linus.jeary@kcl.ac.uk

Tom Kuipers[*]
Department of Informatics
King's College London, UK
tom.kuipers@kcl.ac.uk

Mehran Hosseini
Department of Informatics
King's College London, UK
mehran.hosseini@kcl.ac.uk

Nicola Paoletti
Department of Informatics
King's College London, UK
nicola.paoletti@kcl.ac.uk

Abstract

Conformal Prediction (CP) is a popular uncertainty quantification method that provides distribution-free, statistically valid prediction sets, assuming that training and test data are exchangeable. In such a case, CP's prediction sets are guaranteed to cover the (unknown) true test output with a user-specified probability. Nevertheless, this guarantee is violated when the data is subjected to adversarial attacks, which often result in a significant loss of coverage. Recently, several approaches have been put forward to recover CP guarantees in this setting. These approaches leverage variations of randomised smoothing to produce conservative sets which account for the effect of the adversarial perturbations. They are, however, limited in that they only support ℓ_2-bounded perturbations and classification tasks. This paper introduces *VRCP (Verifiably Robust Conformal Prediction)*, a new framework that leverages recent neural network verification methods to recover coverage guarantees under adversarial attacks. Our VRCP method is the first to support perturbations bounded by arbitrary norms including ℓ_1, ℓ_2, and ℓ_∞, as well as regression tasks. We evaluate and compare our approach on image classification tasks (CIFAR10, CIFAR100, and TinyImageNet) and regression tasks for deep reinforcement learning environments. In every case, VRCP achieves above nominal coverage and yields significantly more efficient and informative prediction regions than the SotA.

1 Introduction

Conformal Prediction (CP) (Vovk et al., 2005; Angelopoulos and Bates, 2021) is a popular uncertainty quantification method. In essence, it is a model-agnostic, distribution-free framework that allows one to construct prediction sets that are guaranteed to include the true (unknown) output with probability greater than $1 - \alpha$, where $\alpha \in (0, 1)$ is a user-specified miscoverage/error rate. In other words, for a test point $(\boldsymbol{x}_{n+1}, y_{n+1})$, CP seeks to construct a prediction set $C(\boldsymbol{x}_{n+1})$ such that the following coverage (a.k.a. validity) guarantee holds:

$$\mathbb{P}_{\boldsymbol{x}_{n+1}, y_{n+1}}[y_{n+1} \in C(\boldsymbol{x}_{n+1})] \geq 1 - \alpha. \tag{1}$$

Importantly, the above guarantee holds when the calibration data, used to construct $C(\boldsymbol{x}_{n+1})$, and the test point are exchangeable (a special case is when calibration and test data are i.i.d.).
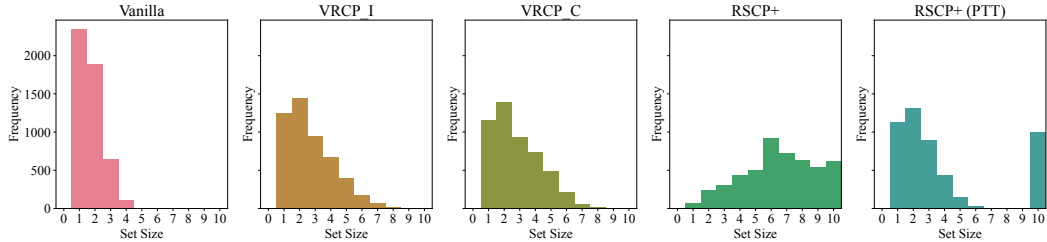
---

[*]Authors contributed equally.

Figure 1: Distribution of prediction set sizes for vanilla conformal prediction (vanilla CP) which violates Eq. (2), as well as for our proposed robust algorithms (VRCP–I and VRCP–C) along with the SotA (RSCP+ and RSCP+ (PTT), see Section 3) on the CIFAR10 dataset. As we observe, VRCP–I and VRCP–C closely resemble the spread of vanilla CP prediction set sizes, whilst the SotA falls short of achieving this. Here we use an adversarial perturbation of radius $\epsilon = 0.02$, error rate $\alpha = 0.1$, number of splits $n_{\text{splits}} = 50$ and smoothing parameter (used in RSCP+ and RSCP+ (PTT)) $\sigma = 2\epsilon$.

When exchangeability is violated, e.g., in the presence of test-time distribution shifts, CP's coverage guarantee (1) ceases to hold, and we cannot rely on the prediction sets it produces. In this work, we address shifts induced by adversarial perturbations on the test inputs. In particular, we focus on perturbations in the form of additive ℓ_p-bounded noise.

To recover guarantees under adversarial inputs, the general mechanism is to inflate the prediction set to permit larger degrees of uncertainty. However, special care must be taken to avoid producing overly large or even trivial sets – i.e. those containing all possible outputs – as such sets do not provide any useful inference.

Contributions We propose a CP framework that provides statistically valid prediction sets despite the presence of ℓ_p-bounded adversarial perturbations at inference time. Formally, for any adversarially perturbed test point $\tilde{x}_{n+1} = x_{n+1} + \delta$, our method produces adversarially robust sets C_ϵ that enjoy the following guarantee:

$$\mathbb{P}[y_{n+1} \in C_\epsilon(\tilde{x}_{n+1})] \geq 1 - \alpha \quad \forall \delta \text{ s.t. } \|\delta\|_p \leq \epsilon. \tag{2}$$

While CP uses an underlying predictor f, often a neural network (NN), to construct prediction regions, the novelty of our approach is to leverage NN verification algorithms to compute upper and lower output bounds of $f(x')$ for any $x' \in B_\epsilon(x) = \{x' : \|x' - x\|_p \leq \epsilon\}$. We use these bounds to inflate the CP regions, resulting in provably robust and *efficient* prediction sets. To the best of our knowledge, this is the first work that combines NN verification algorithms and CP to construct adversarially robust prediction sets. We call our method *VRCP (Verifiably Robust Conformal Prediction)*.

Recent work (discussed in Section 3) achieves adversarially robust coverage using probabilistic methods, specifically, randomised smoothing (Cohen et al., 2019). Our approach overcomes some of the theoretical and empirical drawbacks of these prior methods, which are restricted to classification tasks with ℓ_2-norm bounded guarantees and are overly conservative in practice.

Thanks to our verification-based approach, VRCP is the first to extend adversarially robust conformal prediction to regression tasks and the first to go beyond ℓ_2-norm bounded guarantees. In Section 4, we introduce two versions of VRCP that apply verification at calibration and inference time, respectively. Further, in Section 5, we empirically validate our theoretical guarantees and demonstrate a direct improvement over previous work in terms of prediction set efficiency (i.e., average set size) compared to prior work. Fig. 1 shows an extract of our results on CIFAR10, demonstrating that VRCP yields more informative (tighter) prediction regions, a trend that we observe experimentally across all our benchmarks.

2 Preliminaries

We denote with $\mathbb{R}_+$ the set of positive real numbers. Vectors $x \in \mathbb{R}^d$ are shown in bold italic and scalars $x \in \mathbb{R}$ in italic typeface. We denote the norm used to make $\mathbb{R}^d$ a normed vector space by $\|\cdot\|$. This could for instance be ℓ_1, ℓ_2, or ℓ_∞-norm. Whenever a specific norm is intended, we indicate it using an index, e.g., $\|\cdot\|_2$ indicates the ℓ_2-norm. We denote the ϵ-ball around a point $x \in \mathbb{R}^d$ with respect to the used norm by $B_\epsilon(x)$.

2.1 Conformal Prediction

We provide a brief overview of the inductive (or split) vanilla CP approach. Suppose we have a dataset $\mathcal{D}$ containing pairs $(\boldsymbol{x}, y)$ sampled i.i.d. from an (unknown) data-generating distribution over a feature space $X \subseteq \mathbb{R}^d$ and label space Y such that $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$.

We partition the dataset into disjoint training and calibration sets $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{cal}}$, letting $n = |\mathcal{D}_{\text{cal}}|$. We fit a predictor f on $\mathcal{D}_{\text{train}}$ and define a score function $S : (X \times Y) \rightarrow \mathbb{R}$ as some notion of prediction error, such as $S(\boldsymbol{x}, y) = \|f(\boldsymbol{x}) - y\|$ when f is a regressor, or $S(\boldsymbol{x}, y) = 1 - f(\boldsymbol{x})_y$ when f is a classifier with $f(\cdot)_y$ being y's predicted likelihood.

After applying the score function to all calibration points, we construct the score distribution as $F = \delta_\infty/(n+1) + \sum_{i=1}^n \delta_{s_i}/n+1$, where δ_s is the Dirac distribution with parameter s, $s_i = S(\boldsymbol{x}_i, y_i)$ and δ_∞ represents the unknown score (potentially infinite) of the test point.

Given a miscoverage/error rate α and a test point $(\boldsymbol{x}_{n+1}, y_{n+1})$, we define the prediction set $C(\boldsymbol{x}_{n+1})$ by including all labels that appear sufficiently likely w.r.t. the score distribution: $C(\boldsymbol{x}_{n+1}) = \{y \in Y : S(\boldsymbol{x}_{n+1}, y) \leq Q_{1-\alpha}(F)\}$, where $Q_{1-\alpha}(F)$ is the $1 - \alpha$ quantile of F. This set satisfies the marginal coverage guarantee in Eq. (1) if the test point and the calibration points are exchangeable.

2.2 Adversarial Attacks

Neural networks have been shown to be vulnerable to *adversarial attacks*, i.e., small changes to their input that jeopardise the prediction (Szegedy et al., 2014; Biggio and Roli, 2018). This notion can be formally defined as maximising an adversarial objective function (e.g., the loss of the true label) subject to $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\| \leq \epsilon$. Alternatively, it can be defined as finding an adversarial example $\tilde{\boldsymbol{x}} \in \mathbb{R}^m$, such that $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\| \leq \epsilon$ and $\|f(\tilde{\boldsymbol{x}}) - y\| \geq \delta$ for a given neural network $f : \mathbb{R}^d \rightarrow \mathbb{R}^n$.

2.3 Neural Network Verification

Various approaches have been proposed to verify the robustness of NNs against adversarial attacks. These approaches can be divided into complete and incomplete algorithms. Given a neural network f, a verifier is *complete* if it allows computing exact bounds $f^\perp$ and $f^\top$ for the image $f(B_\epsilon(\boldsymbol{x})) = \{f(\boldsymbol{x}') : \boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})\}$, i.e., such that

$$f^\perp = \min_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})} \{f(\boldsymbol{x}')\}, \quad f^\top = \max_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})} \{f(\boldsymbol{x}')\}, \tag{3}$$

where $\min$ and $\max$ are computed coordinate-wise for vector-valued NNs. A verifier is *incomplete*, but *sound*, if it computes bounds that are valid but not exact, i.e., such that:

$$f^\perp \leq \min_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})} \{f(\boldsymbol{x}')\}, \quad f^\top \geq \max_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})} \{f(\boldsymbol{x}')\}. \tag{4}$$

Our results are verifier-agnostic, meaning that they are valid for any verifier that can produce exact bounds (as in Eq. (3)) or conservative bounds (as in Eq. (4)), depending on the completeness or incompleteness of the verifier used. The fastest and simplest way to compute the bounds in Eq. (4) is to propagate the bounds on the input $B_\epsilon(\boldsymbol{x})$ through the network to compute the output bounds. Several methods based on this approach have been proposed (Gowal et al., 2018; Wang et al., 2018; Zhang et al., 2018a; Batten et al., 2024; Lopez et al., 2023). At the expense of fast computation speed, these methods may result in loose bounds in Eq. (4). On the other hand, several complete methods (Pulina and Tacchella, 2010; Katz et al., 2017; Hosseini and Lomuscio, 2023) for NN verification have been put forward. Even though these methods compute exact bounds, their downside is their high computational cost.

3 Related Work

Adversarially Robust Conformal Prediction Gendler et al. (2021) introduced an algorithm called Randomly Smoothed Conformal Prediction (RSCP) that integrates randomised smoothing (Duchi et al., 2012; Cohen et al., 2019; Salman et al., 2019) with CP to provide robust coverage under adversarial attacks. RSCP replaces the CP score function $S(\boldsymbol{x}, y)$ with a smoothed score $\widetilde{S}(\boldsymbol{x}, y)$ obtained by averaging the values of $S(\boldsymbol{x} + \boldsymbol{v}, y)$ over n_{MC} realisations of a Gaussian noise

vector $\boldsymbol{v} \sim \mathcal{N}(0, \sigma^2 I)$, for a given smoothing level σ. To correct for potential ℓ_2-norm bounded ϵ perturbations at inference time, the critical value computed over the smoothed scores is inflated by ϵ/σ. This method produces empirically sound results, but the provided formal guarantees were found to be invalid in a later work (Yan et al., 2023), as discussed below.

Provably Robust Conformal Prediction Yan et al. (2023) address the issue with the robustness guarantee of Gendler et al. (2021) by correctly bounding the estimation error caused by the Monte-Carlo sampling used when generating the smoothed scores. The bound introduces an additional hyperparameter β such that they now find the $Q_{1-\alpha+2\beta}$ of smooth calibrated scores and inflate by Hoeffding's bound $\sqrt{-\ln(\beta)/2n_{\mathrm{MC}}}$ before correcting by ϵ/σ. Furthermore, the smoothed scores of the test points are decreased by an empirical Bernstein bound. This further inflation of the critical value and deflation of smooth scores for each test point often cause their amended algorithm, so-called RSCP+, to generate trivial prediction sets.

To address this issue, the authors introduce two methods to improve the efficiency of RSCP+. Firstly, they use robustly calibrated training (RCT), a training-time regularisation technique that penalises NN parameters that contribute to high scores for the true label. Our approach assumes that the underlying classifier is given; hence, we do not evaluate RCT in our experiments.

Secondly, they implement a post-training transformation (PTT), which aims to decrease the values of the smoothed calibration scores that lie between $Q_{1-\alpha}$ (the critical value of the base scores) and $\widetilde{Q}_{1-\alpha}$ (that of the smoothed scores). To this purpose, they transform the CDF of the smoothed scores $\widetilde{S}$ by composing learned ranking and sigmoid transformations with hyperparameters b and T using a holdout set $\mathcal{D}_{\mathrm{hold}}$ sampled i.i.d from $\mathcal{D}_{\mathrm{cal}}$. PTT however is not theoretically guaranteed to improve the average set sizes computed by RSCP+ and, in many cases, its efficacy is largely dependent on how representative the sampled holdout set is of the calibration set. We demonstrate the effect of PTT's sample dependence empirically in Section 5.1.

Probabilistically Robust Conformal Prediction Ghosh et al. (2023) also focus on the adversarial setting but maintain a relaxed form of robust coverage, where input perturbations $\boldsymbol{\delta}$ are drawn from a specific distribution and only a proportion of such perturbations are sought to be covered. In contrast, we do not make assumptions about the noise distribution, and we account for any ϵ-bounded perturbation.

All the works[2] discussed here rely on randomised smoothing Duchi et al. (2012) and as such are limited to the ℓ_2-norm. In contrast, our VRCP approach relies on NN verifiers, can be used with any ℓ_p-norms supported by the verification method, and does not require smoothing hyperparameters or holdout sets.

4   Verifiably Robust Conformal Prediction (VRCP)

In this section, we formally introduce two variants of VRCP. Both methods allow us to construct adversarially robust prediction sets at inference time.

The first variant, *VRCP via Robust Inference (VRCP–I)*, employs NN verification at inference time to compute a lower bound of the score for the given test input (best-case score), thereby obtaining more conservative regions. The calibration procedure is computed as in standard CP. The second variant, *VRCP via Robust Calibration (VRCP–C)*, instead uses NN verification at calibration time to derive upper bounds for the calibration scores (worst-case), thereby obtaining a more conservative calibration threshold (critical value). This allows us to use the regular scores at inference time, without requiring NN verification.

4.1   Verifiably Robust Conformal Prediction via Robust Inference (VRCP–I)

Given a calibration set $\mathcal{D}_{\mathrm{cal}}$, a test input $\boldsymbol{x}_{n+1}$, and score function $S(\cdot, \cdot)$, we compute the prediction set for $\boldsymbol{x}_{n+1}$ as follows.

---

[2]We are aware of related contemporaneous work by Zargarbashi et al. (2024). However, at the time of submission, neither the manuscript nor the code were available.

1. For each $y \in Y$ we compute,

$$s^\perp(\boldsymbol{x}_{n+1}, y) \leq \inf_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}_{n+1})} S(\boldsymbol{x}', y). \tag{5}$$

2. The robust prediction set is then defined as

$$C_\epsilon(\boldsymbol{x}_{n+1}) = \{y : s^\perp(\boldsymbol{x}_{n+1}, y) \leq Q_{1-\alpha}(F)\}. \tag{6}$$

Below, we show that we are able to maintain the marginal coverage guarantee from Eq. (2) for any ℓ_p-norm bounded adversarial attack.

Theorem 1. *Let $\tilde{\boldsymbol{x}}_{n+1} = \boldsymbol{x}_{n+1} + \boldsymbol{\delta}$ for a clean test sample $\boldsymbol{x}_{n+1}$ and $\|\boldsymbol{\delta}\|_p \leq \epsilon$. The prediction set $C_\epsilon(\tilde{\boldsymbol{x}}_{n+1})$ defined in Eq. (6) satisfies $\mathbb{P}\left[y_{n+1} \in C_\epsilon(\tilde{\boldsymbol{x}}_{n+1})\right] \geq 1 - \alpha$.*

Proof. We obtain that

$$\begin{aligned}
\mathbb{P}\left[y_{n+1} \in C_\epsilon(\tilde{\boldsymbol{x}}_{n+1})\right] &= \mathbb{P}\left[s^\perp(\tilde{\boldsymbol{x}}_{n+1}, y_{n+1}) \leq Q_{1-\alpha}(F)\right] \\
&\geq \mathbb{P}\left[\inf_{\boldsymbol{x}' \in B_\epsilon(\tilde{\boldsymbol{x}}_{n+1})} S(\boldsymbol{x}', y_{n+1}) \leq Q_{1-\alpha}(F)\right] \qquad \text{by Eq. (5)} \\
&\geq \mathbb{P}\left[S(\boldsymbol{x}_{n+1}, y_{n+1}) \leq Q_{1-\alpha}(F)\right] \geq 1 - \alpha. \qquad \square
\end{aligned}$$

4.2 Verifiably Robust Conformal Prediction via Robust Calibration

Given a calibration set $\mathcal{D}_{\text{cal}}$, a test input $\boldsymbol{x}_{n+1}$, and score function $S(\cdot, \cdot)$, we compute the robustly calibrated prediction set for $\boldsymbol{x}_{n+1}$ as follows.

1. We compute the upper-bound calibration distribution as:

$$F^\top = \frac{\delta_\infty}{(n+1)} + \sum_{i=1}^n \frac{\delta_{s_i^\top}}{n+1}, \text{ where } s_i^\top \geq \sup_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}_i)} S(\boldsymbol{x}', y_i). \tag{7}$$

2. The robust post-calibration prediction set is then defined as

$$C_\epsilon(\boldsymbol{x}_{n+1}) = \{y : S(\boldsymbol{x}_{n+1}, y) \leq Q_{1-\alpha}(F^\top)\}. \tag{8}$$

Theorem 2. *Let $\tilde{\boldsymbol{x}}_{n+1} = \boldsymbol{x}_{n+1} + \boldsymbol{\delta}$ for a clean test sample $\boldsymbol{x}_{n+1}$ and $\|\boldsymbol{\delta}\|_p \leq \epsilon$. The prediction set $C_\epsilon(\tilde{\boldsymbol{x}}_{n+1})$ defined in Eq. (8) satisfies $\mathbb{P}\left[y_{n+1} \in C_\epsilon(\tilde{\boldsymbol{x}}_{n+1})\right] \geq 1 - \alpha$.*

Proof. We have that

$$\begin{aligned}
\mathbb{P}\left[y_{n+1} \in C_\epsilon(\tilde{\boldsymbol{x}}_{n+1})\right] &= \mathbb{P}\left[S(\tilde{\boldsymbol{x}}_{n+1}, y_{n+1}) \leq Q_{1-\alpha}\left(F^\top\right)\right] \\
&\geq \mathbb{P}\left[S(\tilde{\boldsymbol{x}}_{n+1}, y_{n+1}) \leq Q_{1-\alpha}\left(\left\{\sup_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}_i)} S(\boldsymbol{x}', y_i)\right\}_{i=1}^n \cup \{\infty\}\right)\right] \\
&\geq \mathbb{P}\left[\sup_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}_{n+1})} S(\boldsymbol{x}', y_{n+1}) \leq Q_{1-\alpha}\left(\left\{\sup_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}_i)} S(\boldsymbol{x}', y_i)\right\}_{i=1}^n \cup \{\infty\}\right)\right] \\
&\geq 1 - \alpha
\end{aligned}$$

Let $P^\top$ denote the distribution of $(\boldsymbol{x}^\top, y)$ where $\boldsymbol{x}^\top = \operatorname{argsup}_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})} S(\boldsymbol{x}', y)$ and $(\boldsymbol{x}, y) \sim P$. The final inequality above holds since it is equivalent to constructing a CP prediction set using n i.i.d realisations of $P^\top$ and evaluating it on $\boldsymbol{x}_{n+1} \sim P^\top$. The resulting set will include the true test output y_{n+1} with probability at least $1 - \alpha$. $\square$

4.3 Computation of score bounds

Classification In the classification setting, we use the score function proposed in (Lei et al., 2013; Gendler et al., 2021):

$$S(\boldsymbol{x}, y) = 1 - f(\boldsymbol{x})_y, \tag{9}$$

where $f(\boldsymbol{x})_y \in (0, 1)$ is the model-predicted likelihood for label y. In this setting, to compute $s^\perp$ and $s^\top$ (required by VRCP–I and VRCP–C, respectively), it suffices to use NN verification algorithms to derive the output bounds for $f(\boldsymbol{x})$. Specifically, in VRCP–I, for a test input $\boldsymbol{x}_{n+1}$ and for each $y \in Y$ we derive $s^\perp(\boldsymbol{x}_{n+1}, y)$ as

$$s^\perp(\boldsymbol{x}_{n+1}, y) = 1 - f(\boldsymbol{x}_{n+1})_y^\top, \tag{10}$$

where $f(\boldsymbol{x}_{n+1})_y^\top$ denotes the upper bound computed by the neural network verifier for the model-predicted likelihood of label $y \in Y$ and input $\boldsymbol{x}_{n+1}$.

In VRCP–C, for each calibration point $(\boldsymbol{x}_i, y_i)$ we compute $s^\top(\boldsymbol{x}_i, y_i)$ as

$$s^\top(\boldsymbol{x}_i, y_i) = 1 - f(\boldsymbol{x}_i)_{y_i}^\perp, \tag{11}$$

where $f(\boldsymbol{x}_i)_{y_i}^\perp$ denotes the lower bound of the model output for label y_i given input $\boldsymbol{x}_i$.

Regression In the regression tasks, we follow the conformalized quantile regression (CQR) methodology proposed by (Romano et al., 2019). We train quantile regressors $f_{\alpha_{\text{low}}}$ and $f_{\alpha_{\text{high}}}$ to estimate the $\alpha_{\text{low}} = \alpha/2$ and $\alpha_{\text{high}} = 1 - \alpha/2$ quantiles of $y \mid \boldsymbol{x}$. In CQR, we use the following score function:

$$S(\boldsymbol{x}, y) = \max\{f_{\alpha_{\text{low}}}(\boldsymbol{x}) - y, y - f_{\alpha_{\text{high}}}(\boldsymbol{x})\}. \tag{12}$$

Unlike classification, where the label space is discrete, we cannot construct the region explicitly by enumerating all possible outputs y. Instead, the prediction region for a given test point $C(\boldsymbol{x}_{n+1})$ is constructed implicitly, by adjusting the quantile predictions by the critical value of the calibration distribution $Q_{1-\alpha}(F)$, as follows:

$$C(\boldsymbol{x}_{n+1}) = \left[f_{\alpha_{\text{low}}}(\boldsymbol{x}_{n+1}) - Q_{1-\alpha}(F), f_{\alpha_{\text{high}}}(\boldsymbol{x}_{n+1}) + Q_{1-\alpha}(F)\right] \tag{13}$$

In both VRCP–C and VRCP–I, the score function leverages an NN verifier to derive the bounds over the upper and lower quantiles of the model. In VRCP–C, we compute the worst-case calibration scores as:

$$s^\top(\boldsymbol{x}_i, y_i) = \max\{f_{\alpha_{\text{low}}}^\top(\boldsymbol{x}_i) - y_i, y_i - f_{\alpha_{\text{high}}}^\perp(\boldsymbol{x}_i)\}. \tag{14}$$

In VRCP–I for classification, for each output we check inclusion in C_ϵ by using the best-case score $s^\perp$. As explained above, explicit enumeration is infeasible for regression, and so we construct our robust region by replacing predicted quantiles in Eq. (13) with their conservative approximations, as follows:

$$C_\epsilon(\boldsymbol{x}_{n+1}) = \left[f_{\alpha_{\text{low}}}^\perp(\boldsymbol{x}_{n+1}) - Q_{1-\alpha}(F), f_{\alpha_{\text{high}}}^\top(\boldsymbol{x}_{n+1}) + Q_{1-\alpha}(F)\right] \tag{15}$$

The above-defined region is equivalent to enumerating all possible outputs y, and for each, considering the best-case score $s^\perp(\boldsymbol{x}_{n+1}, y) = \max\{f_{\alpha_{\text{low}}}^\perp(\boldsymbol{x}_{n+1}) - y, y - f_{\alpha_{\text{high}}}^\top(\boldsymbol{x}_{n+1})\}$. A proof is available in Appendix A.

A nice property of both VRCP–I and VRCP–C is that they guarantee that they can only increase the size of the prediction set for any input $\boldsymbol{x}$ compared to vanilla CP, thus will always attain at least as much coverage as the vanilla CP procedure. Moreover, as we show in Section 5, both algorithms do not trivially inflate the size of the prediction sets and maintain a similar distribution of set sizes. This is formalised in the Proposition 1, which is proved in Appendix A.

Proposition 1. *Let $C(\boldsymbol{x})$ and $C_\epsilon(\boldsymbol{x})$ be the prediction sets obtained using vanilla CP and VRCP (using VRCP–I or VRCP–C), respectively. For any input $\boldsymbol{x}$, we have that $C(\boldsymbol{x}) \subseteq C_\epsilon(\boldsymbol{x})$.*

5 Evaluation

We evaluate VRCP–I and VRCP–C on classification (image) and regression (RL) benchmarks, and compare them against the SotA approaches on each benchmark. For all the networks used, we did not perform adversarial training as we assume that the attack budget ϵ is unknown at training time. Nonetheless, both our approaches can benefit from adversarial training, as it results in models that are more verifiable and have tighter bounds for the same attack budget.[3]

---

[3]Code for the experiments is available at: https://github.com/ddv-lab/Verifiably_Robust_CP

5.1 Classification Experiments

We evaluate each method using a nominal coverage of $1 - \alpha = 0.9$ and report the 95% confidence intervals for coverage and average set sizes computed over 50 splits ($n_{\text{splits}} = 50$) of the calibration, holdout and test set.

Bounds   We use the verification library auto_LiRPA (Xu et al., 2020a) to compute the output bounds for $f(\boldsymbol{x})$ required in Eq. (10) and Eq. (11) for VRCP–I and VRCP–C respectively. In particular, we use two SotA GPU-parallelised incomplete NN verification algorithms, CROWN Zhang et al. (2018b) and α-CROWN Xu et al. (2020b). In brief, CROWN performs linear bound propagation and α-CROWN employs a branch-and-bound algorithm to tighten the CROWN bounds at the expense of slower verification times. Therefore, we use CROWN to compute the output bounds for the TinyImageNet model and α-CROWN for the smaller CIFAR10 and CIFAR100 models.

Our CIFAR10 model with α-CROWN takes ≈ 0.5s per image to compute bounds with $\epsilon = 0.03$, whereas our larger CIFAR100 model takes ≈ 7.2s with $\epsilon = 0.02$. Comparatively, computing the smoothed scores takes ≈ 0.09s per image to compute on both models under the same respective ϵ values. The largest model for the TinyImageNet dataset uses CROWN to compute bounds at a rate of ≈ 0.2s per image whereas the smoothed scores take ≈ 0.24s. All measurements are made with respect to the hardware details listed in Appendix B.

Attacks   We use the PGD attack algorithm (Madry et al., 2017), which is a popular white-box attack algorithm to generate adversarial inputs with respect to either the ℓ_2 or ℓ_∞-norm.

Models   For all datasets, we train a CNN model on training set images with random crop and horizontal flip augmentations. Full model details are outlined in the appendix.

Hyperparameters   RSCP+ based approaches use $\sigma = 2\epsilon$, $\beta = 0.001$ and those with PTT use $|\mathcal{D}_{\text{hold}}| = 500$, $b = 0.9$ and $T = 1/400$. For PGD, we choose a step size of $1/255$ and compute 100 steps for each attack. For CIFAR10 and CIFAR100 $|\mathcal{D}_{\text{train}}| = 50{,}000$ and for TinyImageNet $|\mathcal{D}_{\text{train}}| = 100{,}000$. For all datasets $|\mathcal{D}_{\text{cal}}| = 4{,}500$ and $|\mathcal{D}_{\text{test}}| = 5{,}000$.

Results   In Table 1, we benchmark both our methods against the baseline vanilla CP (which is agnostic of the attack), RSCP+ and RSCP+ with PTT. At inference time, images are attacked using PGD to generate ℓ_2-norm bounded attacks with $\epsilon = 0.02$ for CIFAR100 and TinyImageNet, and $\epsilon = 0.03$ for CIFAR10.

In all domains, the vanilla CP method fails to construct valid prediction sets with nominal marginal coverage, as expected. RSCP+ maintains robust marginal coverage but produces trivial prediction sets in all settings due to the highly conservative inflation of the threshold with respect to the calibration scores. Using PTT improves RSCP+'s performance but introduces significant variance in the set sizes: in many cases, PTT still produces trivial prediction sets and is heavily dependent on the sampled holdout set for RSCP+ to generate useful predictions.

Both of our methods have minimal sample dependence, as demonstrated by a very small variability in coverage and size over the 50 splits. We obtain prediction sets with substantially smaller average sizes than the other robust approaches, and hence, they provide more informative uncertainty estimates. VRCP–I provides slightly more efficient regions than VRCP–C. Still, it implies additional computational overhead at inference time because it requires computing bounds via NN verification for each test sample. In contrast, in VRCP–C, bounds are computed only once at calibration time. On the other hand, in an environment where we may want to change ϵ for different test points at inference time, VRCP–I would be a sound choice, while VRCP–C would require re-calibration.

Effect of increasing adversarial noise   Fig. 2b shows the impact of increasing ϵ across all evaluated robust methods. Our methods consistently produce smaller average set sizes with minor sample dependence, and simultaneously provide a more conservative marginal coverage than RSCP+ (PTT). We remark that, unlike RSCP+, we do not require a holdout set or any score function transformations.

Effect of increasing Monte-Carlo samples   Fig. 2a displays the influence of the n_{MC} hyperparameter on the RSCP+ based methods with respect to our CIFAR10 model. Whilst increasing samples

Table 1: Marginal Coverage and Average Set Sizes for different methods on CIFAR10, CIFAR100 and TinyImageNet. All results record a 95% confidence interval with $n_{\text{splits}} = 50$, $\alpha = 0.1$, $\sigma = 2\epsilon$, $n_{\text{MC}} = 1024$, $\epsilon = 0.03$ for CIFAR10 and $\epsilon = 0.02$ otherwise.

| Method | CIFAR10 | | CIFAR100 | | TinyImageNet | |
	Coverage	Size	Coverage	Size	Coverage	Size
Vanilla	0.878±0.002	1.721±0.008	0.890±0.002	6.702±0.058	0.886±0.002	38.200±0.252
RSCP+	1.000±0.000	10.000±0.000	1.000±0.000	100.000±0.000	1.000±0.000	200.000±0.000
RSCP+ (PTT)	0.983±0.008	8.357±0.780	0.925±0.010	26.375±9.675	0.931±0.013	90.644±20.063
VRCP–I	0.986±0.000	**4.451±0.011**	0.971±0.001	**22.530±0.107**	0.958±0.001	**72.486±0.311**
VRCP–C	0.995±0.000	5.021±0.010	0.983±0.000	23.676±0.131	0.965±0.001	77.761±0.352

(a) Varying n_{MC} for $n_{\text{splits}} = 50$, $\epsilon = 0.03$, $\alpha = 0.1$, and $\sigma = 2\epsilon$.

(b) Varying the value of ϵ for $n_{\text{splits}} = 50$, $\sigma = 2\epsilon$, $\alpha = 0.1$, and $n_{\text{MC}} = 1024$.
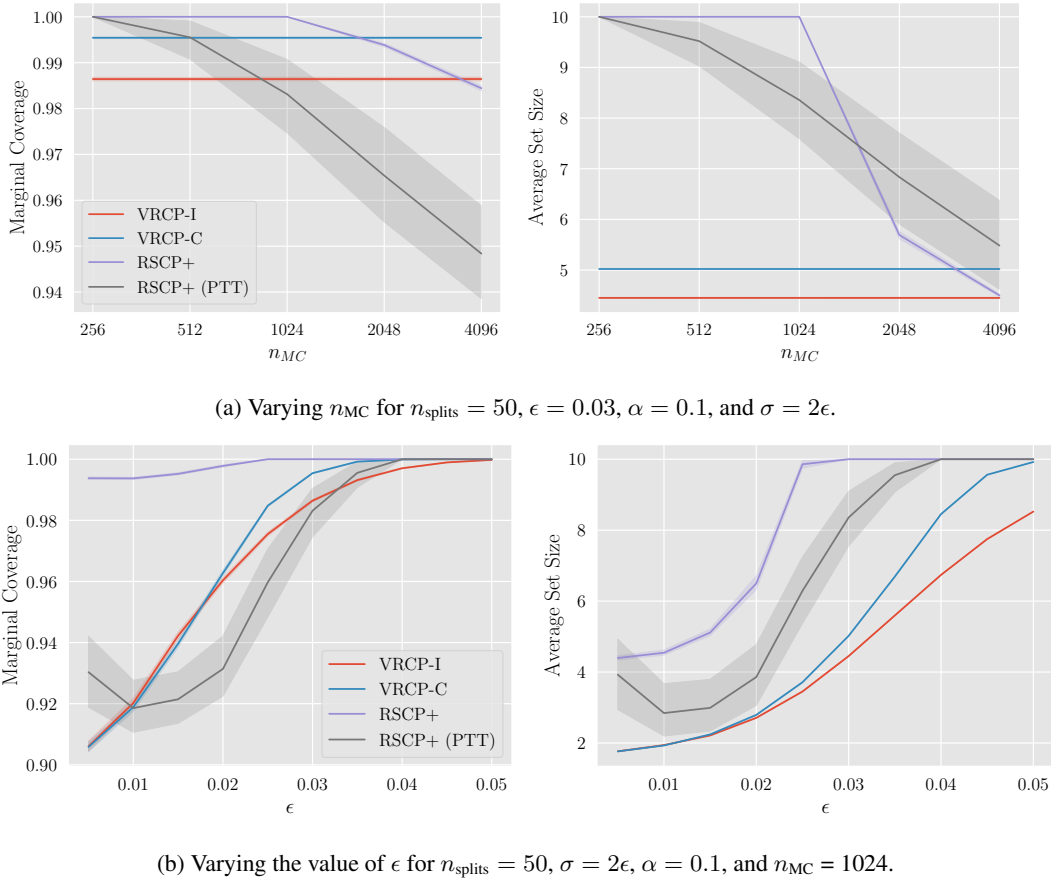
Figure 2: Marginal Coverage and Average Set Sizes on CIFAR100 with 95% confidence intervals.

improves the performances of randomised smoothing approaches, we incur a large computational overhead when computing the smoothed scores. In our experiments in Table 1 we fix $n_{\text{MC}} = 1024$ which is four times larger than the value for n_{MC} used in previous work (Gendler et al., 2021; Yan et al., 2023) as a trade-off between prediction quality and computation.

Beyond ℓ_2-norm bounded attacks   Table 2 demonstrates that both of our methods generalise to other ℓ_p-bounded perturbations other than for when $p = 2$ which RSCP+ is limited to. In particular, we examine the ℓ_∞, where even a small ϵ can cause misclassification. We experiment using CIFAR10 and use $\epsilon = 0.001$. PGD is used to generate ℓ_∞-bounded adversarial examples.

Set size distribution   From Fig. 1 we can visually examine the sample dependency issue that the PTT introduces. In the splits where the holdout set allows the PTT to make an informative transformation, RSCP+ is able to make quite reasonable predictions, otherwise, RSCP+ just returns trivial sets. This is clearly an undesirable property and adds significant variance to the predictions.

Table 2: Marginal Coverage and Average Set Sizes for ϵ perturbations with respect to the ℓ_∞-norm on the CIFAR10 dataset. All results record a 95% confidence interval with $n_{\text{splits}} = 50$, $\alpha = 0.1$ and $\epsilon = 0.001$.

	CIFAR10	
Method	Coverage	Size
Vanilla	0.872±0.002	1.737±0.007
VRCP–I	0.947±0.001	**2.262±0.008**
VRCP–C	0.931±0.001	2.342±0.008

Both of our methods increase the spread of the average set sizes to account for the presence of adversarial examples whilst still maintaining a consistent distribution.

5.2   Regression Experiments

We evaluate our VRCP framework on regression tasks from the PettingZoo Multi-Particle Environment (MPE) library Terry et al. (2021) for deep reinforcement learning. In these environments, the world is a 2D space containing n agents (of which some may be adversarial) and m landmarks, which are defined as circles of fixed radii. The position of the landmarks is fixed, and agents traverse the space according to second-order motion laws. We evaluate our method on three tasks:

- **Adversary** The good agents must try to reach a specific goal landmark whilst avoiding the adversaries. We use 2 good agents, 1 adversary and 2 landmarks.

- **Spread** All agents collaborate and minimise the distance to each landmark. We set the number of agents and landmarks equal to 3.

- **Push** In this task, there is a single good agent, adversary and landmark. The task is for the adversary to maximise the distance between the landmark and the good agent.

In our experiments, for data-generation we select 5,000 random initial world configurations and, for each, simulate 25 Monte-Carlo trajectories of length $k = 5$. The regression task for all environments is to predict the upper and lower quantiles of the total cumulative reward over the k steps, given as input the initial world state. As in the classification experiments, we partition the dataset into the following partitions: $|\mathcal{D}_{\text{train}}| = 1,000$, $|\mathcal{D}_{\text{cal}}| = 2,000$ and $|\mathcal{D}_{\text{test}}| = 2,000$.

For computing the bounds, we use CROWN Zhang et al. (2018b) with ℓ_∞-bounded perturbations. To generate the adversarially perturbed test points, we use the Fast Gradient Sign Method as given in (Goodfellow et al., 2015).

As seen in Table 3, both VRCP methods recover the marginal coverage guarantees in the presence of adversarial perturbations, whereas vanilla CP fails drastically after $\epsilon = 0.02$. We note that the performance of VRCP–C and VRCP–I are similar, although VRCP–I tends to produce more conservative intervals (without sacrificing efficiency).

6   Limitations

VRCP's scalability depends on that of the underlying neural network verifier. We evaluated VRCP on small to medium-sized neural networks. For large networks, existing complete verification methods

Table 3: Marginal coverage and average interval lengths for each MPE regression task for various ϵ perturbations bounded by an ℓ_∞-norm. All results record a 95% confidence interval with $n_{\text{splits}} = 50$.

| | Perturbation | $\epsilon = 0.01$ | | $\epsilon = 0.02$ | | $\epsilon = 0.04$ | |
	Method	Coverage	Length	Coverage	Length	Coverage	Length
Adversary	Vanilla	0.871±0.006	0.480±0.006	0.834±0.007	0.484±0.006	0.745±0.009	0.490±0.006
	VRCP–I	0.928±0.004	0.605±0.006	0.951±0.003	0.673±0.006	0.985±0.002	0.855±0.006
	VRCP–C	0.910±0.005	0.534±0.006	0.923±0.005	0.606±0.006	0.966±0.003	0.806±0.005
Spread	Vanilla	0.864±0.005	0.595±0.005	0.834±0.005	0.602±0.005	0.768±0.006	0.612±0.005
	VRCP–I	0.929±0.004	0.690±0.006	0.958±0.003	0.769±0.006	0.991±0.001	0.992±0.006
	VRCP–C	0.908±0.005	0.663±0.006	0.935±0.004	0.762±0.005	0.977±0.002	1.054±0.006
Push	Vanilla	0.891±0.006	0.643±0.006	0.875±0.007	0.646±0.006	0.841±0.008	0.652±0.006
	VRCP–I	0.917±0.006	0.687±0.006	0.934±0.005	0.721±0.006	0.961±0.003	0.800±0.006
	VRCP–C	0.905±0.005	0.674±0.006	0.910±0.005	0.711±0.005	0.924±0.005	0.795±0.005

would become computationally infeasible, while incomplete methods would produce bounds that are too loose to be useful. However, it is important to note that since VRCP is agnostic of the specific verification tool used, it would directly benefit from any future advances in neural network verification. Thus, as neural network verification tools continue to evolve and improve, so does VRCP.

7 Conclusion

We introduced Verifiably Robust Conformal Prediction (VRCP), a novel framework that leverages conformal prediction and neural network verification to produce prediction sets that maintain marginal coverage under adversarial perturbations. We presented two variants: VRCP–C, which applies verification at calibration time, and VRCP–I, which applies verification at inference time.

Extensive experiments on classification and regression tasks demonstrated that VRCP recovers valid marginal coverage in the presence of ℓ_1, ℓ_2, and ℓ_∞-norm bounded adversarial attacks while producing more accurate prediction sets than existing methods. VRCP is the first adversarially robust CP framework supporting regression tasks and perturbations beyond the ℓ_2-norm, achieving strong results without relying on probabilistic smoothing or posthoc corrections. VRCP's theoretical guarantees and empirical performance showcase the potential of leveraging verification tools for uncertainty quantification of machine learning models under attack.

References

V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic learning in a random world*. Springer, 2005, vol. 29.

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

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

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR*, 2014.

B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 2154–2156.

S. Gowal, K. Dvijotham, R. Stanforth *et al.*, "On the effectiveness of interval bound propagation for training verifiably robust models," 2018, arXiv preprint arXiv:1810.12715.

S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Formal security analysis of neural networks using symbolic intervals," in *USENIX Security Symposium*. USENIX Association, 2018, pp. 1599–1614.

H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," in *Advances in Neural Information Processing Systems, NuerIPS*, 2018.

B. Batten, M. Hosseini, and A. Lomuscio, "Tight verification of probabilistic robustness in bayesian neural networks," in *The 27th International Conference on Artificial Intelligence and Statistics, AISTATS*, vol. 238, 2024.

D. M. Lopez, S. W. Choi, H. Tran, and T. T. Johnson, "NNV 2.0: The neural network verification tool," in *Computer Aided Verification - 35th International Conference, CAV*, ser. Lecture Notes in Computer Science, C. Enea and A. Lal, Eds., vol. 13965. Springer, 2023, pp. 397–412.

L. Pulina and A. Tacchella, "An abstraction-refinement approach to verification of artificial neural networks," in *Computer Aided Verification, 22nd International Conference, CAV*, ser. Lecture Notes in Computer Science, vol. 6174. Springer, 2010, pp. 243–257.

G. Katz, C. W. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *Computer Aided Verification - 29th International Conference, CAV*, ser. Lecture Notes in Computer Science, vol. 10426. Springer, 2017, pp. 97–117.

M. Hosseini and A. Lomuscio, "Bounded and unbounded verification of rnn-based agents in non-deterministic environments," in *International Conference on Autonomous Agents and Multiagent Systems, AAMAS*. IFAAMAS, 2023, pp. 2382–2384.

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

J. C. Duchi, P. L. Bartlett, and M. J. Wainwright, "Randomized smoothing for stochastic optimization," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 674–701, 2012.

H. Salman, J. Li, I. Razenshteyn, P. Zhang, H. Zhang, S. Bubeck, and G. Yang, "Provably robust deep learning via adversarially trained smoothed classifiers," *Advances in neural information processing systems*, vol. 32, 2019.

G. Yan, Y. Romano, and T.-W. Weng, "Provably robust conformal prediction with improved efficiency," in *The Twelfth International Conference on Learning Representations*, 2023.

S. Ghosh, Y. Shi, T. Belkhouja, Y. Yan, J. Doppa, and B. Jones, "Probabilistically robust conformal prediction," in *Uncertainty in Artificial Intelligence*. PMLR, 2023, pp. 681–690.

S. H. Zargarbashi, M. S. Akhondzadeh, and A. Bojchevski, "Robust yet efficient conformal prediction sets," in *41st International Conference on Machine Learning, ICML*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 21–27 Jul 2024, pp. 17 123–17 147.

J. Lei, J. Robins, and L. Wasserman, "Distribution-free prediction sets," *Journal of the American Statistical Association*, vol. 108, no. 501, pp. 278–287, 2013.

Y. Romano, E. Patterson, and E. Candes, "Conformalized quantile regression," *Advances in neural information processing systems*, vol. 32, 2019.

K. Xu, Z. Shi, H. Zhang, Y. Wang, K.-W. Chang, M. Huang, B. Kailkhura, X. Lin, and C.-J. Hsieh, "Automatic perturbation analysis for scalable certified robustness and beyond," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," *Advances in neural information processing systems*, vol. 31, 2018.

K. Xu, H. Zhang, S. Wang, Y. Wang, S. Jana, X. Lin, and C.-J. Hsieh, "Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers," *arXiv preprint arXiv:2011.13824*, 2020.

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

J. Terry, B. Black, N. Grammel, M. Jayakumar, A. Hari, R. Sullivan, L. S. Santos, C. Dieffendahl, C. Horsch, R. Perez-Vicente *et al.*, "Pettingzoo: Gym for multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 032–15 043, 2021.

I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

A Additional Proof Details

Here we prove Proposition 1 regarding the prediction sets obtained from VRCP–I and VRCP–C.

Proof of Proposition 1. To prove $C(\boldsymbol{x}) \subseteq C_\epsilon(\boldsymbol{x})$ for VRCP–I it suffices to observe that

$$C_\epsilon(\boldsymbol{x}) = \{y \in Y : s^\perp(\boldsymbol{x}, y) \leq Q_{1-\alpha}(F)\} \supseteq \bigcup_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}_{n+1})} \{y \in Y : S(\boldsymbol{x}', y) \leq Q_{1-\alpha}(F)\}$$

$$= \bigcup_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}_{n+1})} C(\boldsymbol{x}') \supseteq C(\boldsymbol{x}).$$

To prove the same for VRCP–C, we observe that since $Q_{1-\alpha}(F^\top) \geq Q_{1-\alpha}(F)$, we have that

$$C_\epsilon(\boldsymbol{x}) = \{y \in Y : S(\boldsymbol{x}, y) \leq Q_{1-\alpha}(F^\top)\} \supseteq \{y \in Y : S(\boldsymbol{x}, y) \leq Q_{1-\alpha}(F)\} = C(\boldsymbol{x}). \qquad \square$$

Next, we prove the validity of the VRCP–I region for the regression case, defined in Eq. (15).

Proof. It suffices to show that all $y \in C_\epsilon(\boldsymbol{x}_{n+1}) = [f^\perp - q, f^\top + q]$ satisfy $s^\perp(\boldsymbol{x}_{n+1}, y) = \max\{f^\perp - y, y - f^\top\} \leq q$ and all $y \notin C_\epsilon(\boldsymbol{x}_{n+1})$ do not. For simplicity of notation, we abbreviated $f^\top_{\alpha_{\text{high}}}(\boldsymbol{x}_{n+1})$ with $f^\top$, $f^\perp_{\alpha_{\text{low}}}(\boldsymbol{x}_{n+1})$ with $f^\perp$ and $Q_{1-\alpha}(F)$ with q.

Assume $y \in C_\epsilon(\boldsymbol{x}_{n+1})$. We divide the proof into two cases:

1. $s^\perp(\boldsymbol{x}_{n+1}, y) = f^\perp - y$, which implies that $y \in [f^\perp - q, \frac{f^\top + f^\perp}{2}]$. It suffices to show that $f^\perp - y \leq q$ for $y = f^\perp - q$, which is clearly satisfied.

2. $s^\perp(\boldsymbol{x}_{n+1}, y) = y - f^\top$, which implies that $y \in [\frac{f^\top + f^\perp}{2}, f^\top + q]$. It suffices to show that $y - f^\top \leq q$ for $y = f^\top + q$, which is clearly satisfied.

Finally, we show that $y \notin C_\epsilon(\boldsymbol{x}_{n+1})$ implies $s^\perp(\boldsymbol{x}_{n+1}, y) > q$: if $y < f^\perp - q$, we have that $s^\perp(\boldsymbol{x}_{n+1}, y) = f^\perp - y > q$. Similarly, if $y > f^\perp + q$, we have that $s^\perp(\boldsymbol{x}_{n+1}, y) = y - f^\top > q$. $\qquad \square$

B Model Details

All experimental results were obtained from running the code provided in our GitHub repository on a server with 2x Intel Xeon Platinum 8360Y (36 cores, 72 threads, 2.4GHz), 512GB of RAM and an NVIDIA A40 48GB GPU. All pre-trained models as well as the training scripts are also provided in the GitHub repository. In summary, the models' train and test performances are provided in Tables 4 and 5.

Table 4: Train and test accuracies (%) for the classifications models on CIFAR10, CIFAR100, and TinyImageNet datasets. It should be noted that the model's accuracy has no effect on VRCP's validity and only affects the efficiency of the prediction sets (more accurate models, tighter prediction regions)

Metric	CIFAR10	CIFAR100	TinyImageNet
Train Top-5	98.77	90.49	78.44
Train Top-1	77.80	67.12	52.81
Test Top-5	98.27	82.87	55.72
Test Top-1	76.52	55.73	29.65

B.1 Classification

CIFAR10 We use 2 convolution layers with average pooling and dropout, followed by 2 linear layers. ReLU activations across all layers.

CIFAR100 We use 1 convolution layer with average pooling, 2 further convolution layers with average pooling and dropout followed by 2 linear layers. ReLU activations across all layers.

TinyImageNet   We use 4 convolution layers with dropout followed by 2 linear layers with dropout. Leaky ReLU activation function with $a = 0.1$

For all models we train using images augmented with random crop with 4 pixels of padding and random horizontal flip. We standardise the TinyImageNet models with $\mu = 0.5$ and $\sigma = 0.5$ overall 3 RGB channels.

As previously mentioned, we do not make any assumptions during training about the perturbations we expect to see at inference time. As such, unlike the existing SotA methods, we do not train on smoothed or adversarially attacked images.

All models are trained for 200 epochs with a batch size of 128 using the stochastic gradient descent optimiser with momentum set to 0.9. We also employ a weight decay of 5×10^{-4} and a cosine annealing learning rate scheduler.

B.2   Regression

For the MPE datasets, we train Deep Q-Net policies for the RL tasks for the sole purposes of generating the appropriate datasets and provide these policies in the GitHub repository.

The model used for the quantile regressors is a simple linear architecture consisting of 3 layers, separated with ReLU activation functions and dropout. We trained the model to estimate the $\alpha/2$ and $1 - \alpha/2$ quantiles, where $\alpha = 0.1$, as in the other experiments.

The exact parameters for the RL policies can be found in the config files within the GitHub repository, however have little bearing on the efficiency of our results, being used only for the data-generating process. The quantile regressors are each trained for 400 epochs, with a learning rate of 10^{-5}, dropout of 0.1 and a decay of 10^{-5}.

Table 5: Train and test loss for the regression models in the adversary, spread, and push environments.

Metric	Adversary	Spread	Push
Train	0.066	0.075	0.075
Test	0.051	0.053	0.068

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Claims made in the abstract and introduction are supported by our results in Section 5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Covered in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Covered in Section 4.

   Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All relevant details provided in Section 5, Appendix B and in our GitHub repository.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All code used for experiments is available in our GitHub repository.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.

- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Hyperparameters for experiments are all provided in Section 5 and model details are provided in Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All experiments in Section 5 provide a 95% confidence interval taken over 50 splits.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources discussed in Section 5 and Appendix B

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have reviewed the CoE and deemed that our work is fully aligned with the statements.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no significant societal impact of the work provided outside of the fact that we provide a robust approach to defend against adversarial attacks.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our approach provides defences for adversarial attacks and our methods are not exploitable by design.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All credit is given to authors of any assets used as a citation in the paper in the README file of the GitHub repository. Relevant licenses are also included in the same repository.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The model details are included in Appendix B and documentation on how to reproduce results and train the models is given in the README file in the GitHub repository.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Not relevant to our work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Not relevant to our work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.