University of Cologne

Doctoral Thesis

Distribution-Free Uncertainty Quantification under Adversarial Attacks and Graph Dependencies

Author:
Soroush H. Zargarbashi

Legal Name:
Sayed Soroush Haj Zargarbashi

Supervisor:
Prof. Dr. Aleksandar Bojchevski

Examiners:
Prof. Dr. Jilles Vreeken
Dr. Sebastian Stich

Dissertation for the degree of

Doctor of Natural Sciences (Dr. rer. nat.)

in the

Faculty of Mathematics and Natural Sciences

Date of submission: April 07, 2026

Abstract

With the vast deployment of machine learning models in safety-critical domains, rigorous uncertainty quantification is essential to ensure a reliable and trustworthy AI ecosystem. Building on top of conformal prediction (CP) — a powerful framework that provides distribution-free, finite-sample guarantees on predictive validity — this dissertation develops novel methods for graph-aware and robust application of conformal prediction.

First, we discuss conformal prediction for graph node-classification task. We show that the interdependence in GNN message passing does not break the conformal guarantee as long as the GNN is permutation equivariant. We show the validity of conformal prediction in a transductive setup where the entire graph is visible at once, and in node, or edge-exchangeable inductive setup where the graph is updated with new nodes and edges over time. Leveraging the neighborhood structure, we improve set size efficiency – we show that diffusion over the conformity score can reduce the size of the prediction set while maintaining the same guarantee.

Additionally, it is shown that AI models are vulnerable to various forms of noise and adversarial attacks, which can significantly degrade their performance and reliability. Same vulnerabilities can also affect the validity of conformal prediction sets – by a tiny perturbation in the input, the guarantee of CP can be completely broken, and the empirical coverage sharply drops to zero. We discuss the robustness of conformal prediction to worst-case natural noise and adversarial perturbations. We design provably robust CP sets that remain valid under worst-case perturbations in conformity scores, covering both evasion and poisoning attacks. We build our robust CP on randomized smoothing framework which provides robustness with black-box model access to larger magnitudes of perturbations. While randomized smoothing provides strong robustness guarantees, it can be computationally expensive. To address this, we introduce single-sample robust conformal predictors that certify the CP procedure directly rather than per-score, achieving comparable robustness with significantly lower computational cost. Our work advances the state-of-the-art in robust conformal prediction both in set size and computational efficiency, paving the way for more reliable and trustworthy AI systems in safety-critical and real-time applications.

Zusammenfassung

Mit dem breiten Einsatz von Machine-Learning-Modellen in sicherheitskritischen Anwendungsbereichen ist eine rigorose Unsicherheitsquantifizierung essenziell, um ein verlässliches und vertrauenswürdiges KI-Ökosystem sicherzustellen. Zudem wurde gezeigt, dass KI-Modelle gegenüber verschiedenen Formen von Rauschen und adversarialen Angriffen anfällig sind, was ihre Leistung und Zuverlässigkeit erheblich beeinträchtigen kann. Aufbauend auf Conformal Prediction (CP) – einem leistungsfähigen Rahmenwerk, das verteilungsfreie Garantien endlicher Stichproben für die prädiktive Validität liefert – entwickelt diese Dissertation neuartige Methoden für robuste und effiziente Unsicherheitsquantifizierung.

Wir behandeln CP-basierte Methoden für die Aufgabe der Knotenk klassifikation in Graphen. Dabei zeigen wir, dass die Abhängigkeiten im Message Passing von Graph Neural Networks (GNNs) die konformen Garantien nicht verletzen, solange das GNN permutationsäquivariant ist. Wir betrachten sowohl das transduktive Szenario, in dem der gesamte Graph auf einmal sichtbar ist, als auch induktive Szenarien mit Knoten- oder Kanten-Austauschbarkeit, bei denen der Graph im Zeitverlauf um neue Knoten und Kanten erweitert wird. Unter Ausnutzung der Nachbarschaftsstruktur verbessern wir die Effizienz der Vorhersagemengen; wir zeigen, dass eine Diffusion über den Konformitätsscore die Größe der Vorhersagemenge verringern kann, ohne die gleiche Garantie zu verlieren.

Orthogonal dazu untersuchen wir die Robustheit von Conformal Prediction gegenüber natürlichem Worst-Case-Rauschen und adversarialen Störungen. Wir entwerfen nachweisbar robuste CP-Mengen, die auch unter Worst-Case-Perturbationen der Konformitätsscores gültig bleiben, und decken dabei sowohl Evasion- als auch Poisoning-Angriffe ab. Unsere robuste CP basiert auf dem Framework des Randomized Smoothing, das bei Black-Box-Zugriff auf das Modell Robustheit gegenüber größeren Störungsstärken bietet. Da Randomized Smoothing jedoch mit hohen Rechenkosten verbunden sein kann, führen wir Single-Sample-robuste konforme Prädiktoren ein, die direkt das CP-Verfahren statt einzelner Scores zertifizieren und damit vergleichbare Robustheit bei deutlich geringerem Rechenaufwand erreichen.

Unsere Arbeit erweitert den Stand der Technik in der robusten Conformal Prediction sowohl hinsichtlich der Größe der Vorhersagemengen als auch der rechnerischen Effizienz und ebnet damit den Weg für verlässlichere und vertrauenswürdigere KI-Systeme in sicherheitskritischen und echtzeitnahen Anwendungen.

Acknowledgements

Writing this thesis has been an opportunity not only to reflect on the research itself, but also on the journey that led here. Looking back, I feel that my PhD has been successful—at least in part—and I am deeply satisfied with what I have been able to accomplish. While effort has certainly played a role, I am equally aware of how much of this journey has been shaped by fortunate circumstances. Early achievements foster confidence, encourage risk-taking, and gradually build a reputation; in turn, these open the door to further opportunities. In many ways, progress feeds on itself.

I began my PhD without a clear vision and with limited experience in research. On my first day, I did not yet understand what it meant to pursue high-quality scientific work. My supervisor, Prof. Dr. Aleksandar Bojchevski, played a defining role in shaping this path. With patience and clarity, he guided me, from developing an intuition for identifying meaningful problems to mastering the craft of writing and communicating research. Over time, this guidance allowed me to grow into an independent researcher, capable of initiating, developing, and completing projects. I believe my supervisor has shaped my research journey more effectively than the responsibilities of a mentor; he created an environment for me that was conducive to growth, learning, and success. More than anything, he was always supportive, encouraging, and patient through all my mistakes and failures. For this, I am profoundly grateful.

I have also been fortunate to work alongside exceptional collaborators, whose insight and rigor have significantly strengthened this work. I would like to thank Sadegh Akhondzadeh, Alireza Javanmardi, Dr. Siu Lun Chau, Simone Antonelli, Prof. Dr. Willem Waegeman, Jimin Cao, and Prof. Dr. Eyke Hüllermeier, among others, for their valuable contributions and thoughtful feedback. I am equally thankful for the many insightful discussions with great minds such as Prof. Dr. Jilles Vreeken and Prof. Dr. Rebekka Burkholz, and Guiliana Thomanek, whose perspectives have left a lasting impression on my thinking.

Beyond academia, this journey is inseparable from the people who supported me along the way. I often recall a moment early in my marriage, during a settled life in Esfahan, when my wife started a difficult conversation. She confronted me about setting aside my ambitions and urged me to pursue the path I truly wanted. What followed were intense months of applications and interviews—long days that began with work, continued with applications, over and over again. Upon getting accepted for the PhD program, she tolerated the 4 challenging years full of deadlines, not having any settlement for year, to many other unbearable experiences. Through all of this, she stood by me with unwavering support and determination. Without her encouragement, this path might never have unfolded. For that, I am endlessly grateful — thank you, Sarah! I'm more than happy to celebrate this with you!

I would also like to express my sincere gratitude to my parents. From childhood, my father nurtured my curiosity with patience and care, and he was always a role model of hard work and dedication. I still remember his PhD graduation ceremony. The next day, I, as a four-year-old, made my own graduation cap. That memory has stayed with me to this day. (As a funny story, I even once accidentally deleted his thesis from his computer while he was working on it—fortunately, passwords are more common now, and perhaps they can stop karma from striking me!) My mother has always been a constant source of support and kindness. She always had time for the curious little me. I am equally thankful to my brother Sepehr for his encouragements; he is truely kind and supportive. From my very first days in Germany, I was fortunate to be surrounded by a close circle of friends who made this journey both meaningful and enjoyable. It

would be incomplete not to acknowledge them: Prof. Dr. Saber Talari, Dr. Ati Motalebi, Maryam Meghdadi, Maryam Talebi, Omid Yazdizadeh, Sadegh Akhondzadeh, Kamal Barati, Amin Shahnazari, Ahmad Babadi, Mahnoosh, Kayvan, Pegah, Sadra, Saleh, Zahra, and many others. As a future AI scientist, I should also say thanks to attention, which seems to be all we need!

Finally, I would like to thank CISPA Helmholtz Center for Information Security, and the University of Cologne for providing a supportive and inspiring research environment. I am grateful for the resources, opportunities, and community that have been instrumental in shaping this work.

I would end this with a phrase from the well-known Persian Poet, Hafez:

On this turquoise dome, traced in lines of gold, they have written: "Of all that is, only the kindness of the noble-hearted shall remain."

Soroush H. Zargarbashi, Cologne, April 2026

Contents

List of Figures

List of Tables

1 Introduction

In the recent years, machine learning has been massively deployed in everyday applications, mainly replacing human decision-making with an automated decision-making. This capacity for automation is genuinely transformative — it democratizes access to expert-level decision-making, reduces costs, and enables applications that were previously unimaginable. Yet this same scalability carries a profound danger: when an automated system is wrong, it is wrong at scale. A single miscalibrated medical screening algorithm deployed across a national health system may confidently dismiss thousands of patients as healthy, potentially leaving the critical treatment window unnoticed; an overconfident credit-scoring model may systematically deny loans to entire demographic groups with no indication that its predictions should be questioned; a self-driving vehicle that assigns near-certain confidence to a misclassified obstacle may take no evasive action at all. In all of these cases, the harm is not merely that the model made a mistake — mistakes are inevitable — but that it made a mistake without knowing it, offering no signal to the humans or systems downstream that caution was warranted.

As studied by Guo et al. (2017) and many others, modern machine learning models are often poorly calibrated, meaning that their confidence scores do not reflect the true likelihood of correctness. This motivates the need for explicit uncertainty quantification methods that can provide reliable estimates. Many existing approaches, such as Bayesian inference, ensemble methods, and Monte Carlo dropout, attempt to quantify uncertainty by modeling a (second-order) distribution over the model's output, allowing for ambiguity in the prediction. These methods often require massive computational resources, both during training and inference; they need retraining the model, and rely on Monte-Carlo sampling to estimate uncertainty. Therefore, it is not surprising that they are often not used in practice, especially in large-scale or latency-sensitive applications. Moreover, most of these methods do not provide finite-sample guarantees on the reliability of their output.

A potential alternative is conformal prediction (CP), which provides distribution-free, finite-sample guarantees on predictive validity. CP constructs prediction sets that are guaranteed to include the true label with a user-specified probability. As CP can be applied on any black-box model as a post-hoc procedure, recent years have seen a surge of interest in developing it for various setups, or with more complicated objectives and constraints. This guarantee is particularly appealing because it does not depend on the underlying model's performance. The performance of the model, and the choice of uncertainty scores derived from it, only affect side metrics such as the efficiency of the prediction (i.e., the size of the prediction sets or how far from uniform is the distribution of coverage among points). The prediction set itself is a suitable representation of uncertainty in risk-averse settings Kiyani et al. (2025), and there are studies showing that even human decision-making can be improved by providing conformal prediction sets (Cresswell et al., 2024).

To apply conformal prediction, we need a holdout set of calibration data that is exchangeable with the test data. Exchangeability is a weaker assumption than i.i.d. as it allows for dependencies. CP works with any conformity score – any score that captures the alignment between the input, and any potential output. The choice of conformity score does not affect the validity of the CP guarantee, but it does affect other metrics. An easy example for the score function is the model's predictive probability (softmax) over labels. With the score function, and the calibration data, CP computes a threshold (also called conformal quantile), and at inference time, it includes in the prediction set

all labels whose conformity score is above the threshold.[1]

This thesis contributes to this growing literature in two main directions: (1) developing conformal prediction methods for graph-structured data, and (2) developing robust conformal prediction methods that remain valid under noise and adversarial attacks.

1.1 Conformal Prediction for Graphs

Graph neural networks (GNNs) are now central to many applications where relational structure matters, however the uncertainty quantification remains challenging for them. Unlike standard i.i.d. data, node classification involves interconnected instances, and the prediction for one node is influenced by others through message passing. This makes direct reuse of many i.i.d.-based uncertainty techniques difficult (Stadler et al., 2021). This interdependence makes it important for us to divide the graph node classification task into two settings: the transductive setting, where the full graph is available at inference time (including training, validation, calibration and test nodes), and the inductive setting, where new nodes or edges can arrive after calibration, inducing implicit distribution shifts in node embeddings and conformity scores – this distinction might not have the same importance for i.i.d. data since there is no dependency between the data points.

At the outset of this thesis, a fundamental question was whether conformal prediction could be applied to graph-structured data. While exchangeability is a weaker assumption than i.i.d. the answer to this question remained non-trivial. In the first dedicated study, Clarkson (2022) concluded that the dependencies in GNNs break the exchangeability assumption, and therefore the coverage guarantee. They proposed a workaround through an approach from Barber et al. (2022) applying CP beyond exchangeability. Their approach however suffered from severe sparsity issues – in empirical evaluations, their method could not return any prediction set (due to runtime issue from empty calibration sets) for up to 79% of the test nodes (see Fig. 1.1).

We show that under node, or edge-exchangeability, which is a common assumption, the coverage guarantee of CP applies to GNNs without any modification. This applies to both transductive setting that the graph is entirely available at calibration time, or the inductive setting where the graph evolves over time. The former holds due to permutation equivariance in GNNs, and the latter due to the symmetricity of message passing-originated shift between calibration and test data. We further show that a homophily structure can be used to improve the efficiency of CP sets, through a diffusion-based score refinement. Our diffused adaptive prediction sets (DAPS) produce smaller prediction sets, while preserving the coverage guarantee (see Fig. 1.2). Interestingly in many cases, the diffusion does not increase the model (top label) accuracy, which suggests that the improvement in efficiency is not simply due to better model performance, but rather due to a denoised uncertainty score. We further provide theoretical, and synthetic empirical supporting that assumption.

1.1.1 Outline of Corresponding Sections

In § 3, we establish the validity of CP for GNNs under node-exchangeable calibration set, and we show how to leverage the homophily structure to improve the efficiency of CP sets. In § 4, we establish the validity of CP for inductive GNNs under node,

[1]Note that in this work (since the main setup is classification) we use "conformity" score instead of non-conformity. The setups are equivalent upon a change in the scores sign.

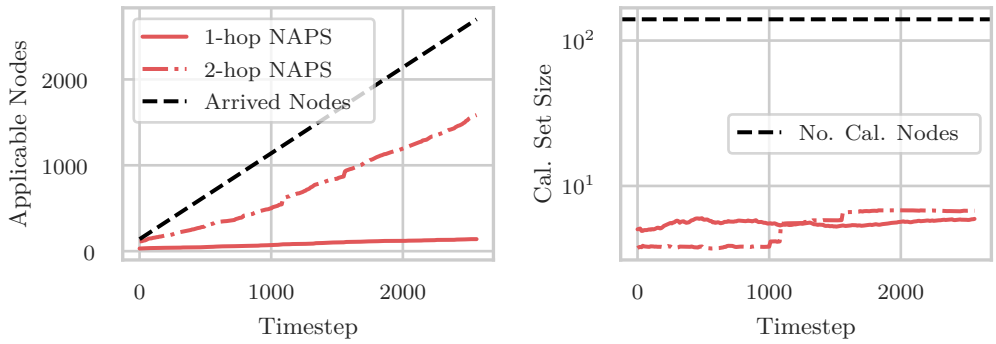

Figure 1.1: [Left] The proportion of nodes without a prediction set due to empty calibration on `CoraML` dataset. The plot shows the inapplicability of k-hop NAPS with $k \in \{1, 2\}$. [Right] The average size of each node's filtered calibration set. Note that this plot excludes non-applicable nodes. Originally shown in Fig. 1.1.

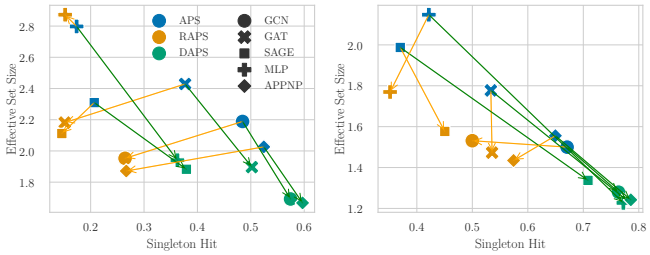

Figure 1.2: Comparing efficiency and singleton hit ratio for all models on `CoraML` (left) and `CoauthorPhysics` (right) datasets over coverage guarantee taken adaptively to the model's accuracy. Figure originally shown in Fig. 3.3.

and edge-exchangeable calibration set. We show that with recalibration, the coverage guarantee can be preserved even when there is a shift in the score distribution. We show that such shift is symmetric over the calibration and test set, therefore recalibration, or weighted recalibration (for the edge-exchangeable case) can restore the coverage guarantee.

The corresponding publications are as following:

Conformal Prediction Sets for Graph Neural Networks — ICML 2023

S. H. Zargarbashi, S. Antonelli, A. Bojchevski

`Reference to Section: § 3`

We prove that in permutation equivariant GNNs, the exchangeability which is a key requirement for the validity of CP, is preserved. This means that the standard CP procedure can be applied to GNNs without any modification. We propose DAPS (Diffusion-based Adaptive Prediction Sets), which refines conformity scores by propagating them over the graph's homophily structure via a diffusion operator. DAPS produces significantly smaller prediction sets than standard CP while preserving the same distribution-free coverage guarantee, with no need for model retraining.

The paper was presented as a poster at ICML 2023. I am the corresponding author of this publication.

Contributions: I was responsible implementation, and experiments, and the writing of the manuscript. All the theoretical development was in collaboration with my co-authors, specially with my supervisor, *Prof. Dr. Aleksandar Bojchevski*. The seed initial idea of trying to baseline diffusion was originally made by my supervisor. For many of the experiments, *Simone Antonelli* provided very valuable help both in implementation and in the analysis of the results. Same as all publications, my coauthors (specially *Aleksandar Bojchevsk*i) have contributed meaningfully to the manuscript, proofs, examining the results, and the code.

Supplementary: Supplementary Material: § A

Conformal Inductive Graph Neural Networks ICLR 2024

S. H. Zargarbashi, A. Bojchevski

Reference to Section: § 4

In sequential inductive settings, new nodes or edges arriving after calibration can implicitly shift GNN embeddings and conformity scores, violating the exchangeability assumption that underpins CP validity. We characterize the precise conditions — node- and edge-exchangeability of the underlying graph model — and we show that in these two scenarios, the shift in the distribution is symmetric between the calibration and test phases. For node-exchangeability, we prove that a re-calibration can hold the guarantee. For edge-exchangeability, we derive weights for which a weighted calibration can recover the nominal coverage probability.

The paper was presented at ICLR 2024. I am the corresponding author of this publication.

Contributions: Working on this research question was originally proposed by *Aleksandar Bojchevski*. I was responsible for the theoretical development, implementation, and experiments, and the writing of the manuscript. All theoretical developments, writings, and evaluation were in collaboration with my supervisor.

Supplementary: Supplementary Material: § B

1.2 Robust Conformal Prediction

The coverage guarantee in CP relies on the exchangeability assumption. This means that the distribution of the calibration data is the same as that of the test data. However, this assumption breaks down in presence of noise and adversary. Same as the model's top-class prediction, in CP, an adversary can decrease the empirical coverage from the nominal level (e.g. 90%) drastically down to zero, just by introducing an unnoticeable perturbation to the test input. This threat can be both at inference time (evasion attack – perturbing test inputs), at calibration time (poisoning attack – perturbing calibration data) or both. This motivates the need for robust conformal prediction methods that can provide coverage guarantees even under worst-case perturbations. Robust conformal prediction (RCP) was first introduced by Gendler et al. (2021) extending the coverage guarantee to the worst-case perturbation. The extension of the guarantee is proposed using randomized smoothing framework – a powerful method for certifying robustness of black-box models against adversarial attacks through augmenting the potentially perturbed input with random noise of a standard class. Later RCP was extended to use other robustness certificates, including verification-based certificates (Jeary et al.,

2024), Lipschitz-based certificates (Massena et al., 2025), etc. By the time, randomized smoothing remained the most promising method as (i) it works for any black-box model without limitation over the architecture, size, etc, (ii) it can be applied to any data type, including continuous, discrete, and sparse data, and (iii) it provides strong robustness guarantees against perturbations of a very larger magnitude compared to other methods.

An important shortcoming of randomized smoothing is the computational cost. To certify the robustness of a single test point, randomized smoothing requires Monte-Carlo estimation which is done through hundreds of model forward passes over each single input. This overhead is prohibitive for large-scale or latency-sensitive applications. In conformal prediction this translates to a trade-off between robustness and efficiency: the guarantee remains valid, but reducing the Monte-Carlo samples results in larger, less informative prediction sets. This trade-off becomes more severe for datasets with many classes. Therefore, the initial works on randomized smoothing RCP could not return non-trivial prediction sets (allowing all classes) for datasets like ImageNet even with 2000 forwards per point.

In this mainstream of research, we develop a sequence of methods with four key improvements: (i) we applied tighter robustness certificates leading to smaller prediction sets while preserving the same level of robustness, (ii) we extended the framework to cover both label, and feature poisoning attacks, in addition to evasion attacks, and (iii) we reduced the computational cost of robustness certification first by an order of magnitude, and then all the way down to the same cost as a single model forward. (iv) While the initial RCP method Gendler et al. (2021) provided guarantee using confidence certificates (certificates in the continuous space), we derived RCP methods using binary certificates, allowing to use the much broader toolbox of binary certificates for free.

1.2.1 Outline of Corresponding Sections

In § 5, we use CDF information of the smooth score distribution to derive tighter bounds, leading to smaller prediction sets at the same robustness level. We further extend the robust CP framework to calibration-time poisoning attacks, both in feature and label space. While the framework results in significantly smaller prediction sets, still it requires a significant number of samples. The empirical evaluation in this section is conducted with 10^4 samples, and still on the ImageNet dataset, which has 1000 classes, the method could not return non-trivial prediction sets with finite sample correction. Therefore, in similar setups the method can only be applied in asymptotically valid setup.

We continue in § 6 to show that by replacing continuous certificates with binary ones, we can reduce the number of samples required to obtain the same set size. Empirically we can obtain similar set sizes with one order of magnitude less samples (see Fig. 1.3) This binarization also allows us to use any existing binary certificates for free. The remaining challenge at the time of this work was to find the lowest possible sample-rate for an acceptable set size.

In § 7 we show that a single forward pass of the model on one noise-augmented test point suffices to certify the robustness of the CP. To achieve that we use an alternative approach to robust CP. So far all methods where using some certified bounds over the score functions. We instead applied the certification to the coverage probability of the test point itself. By showing that the randomized smoothing certificate is convex to the input probability, we used the Jensen's inequality to directly apply the certificate,

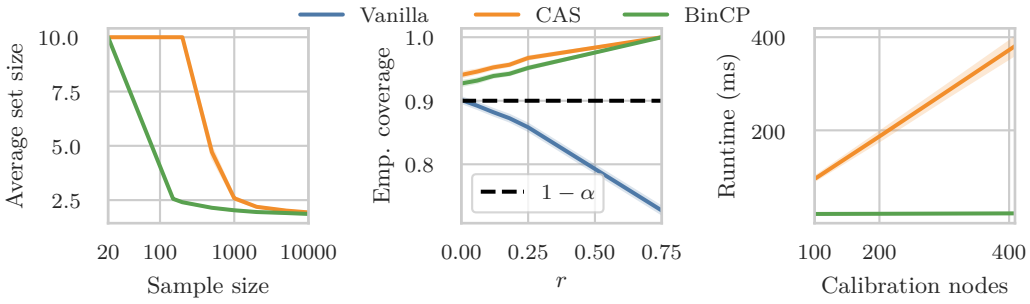

Figure 1.3: [Left] Average set size with different MC sample rates, [Middle] empirical coverage of vanilla and robust CPs under attack, and [Right] runtime of robust CP as a function of calibration datapoints (after computing the MC samples which is the number of lower bound computations). Originally shown in Fig. 6.1.

once over the known value $1-\alpha$. As this value is given in prior without any need for estimation, we can obtain robust prediction sets with a single forward pass of the model both during calibration and test-time.

Notably our single-sample robust CP method has two issues: (i) while it provides similar prediction sets to state-of-the-art with 150 sample per point, it can not be further improved. A property that previous sample-heavy approaches had. (ii) The approach is stochastic by nature meaning that out of bad random events, some inputs might end up with larger prediction sets. This can not be remedied by re-runing the random sample as it becomes multiple hypothesis testing issue. In § 8 we propose two updates to our single sample RCP approach. Via Monte-Carlo estimation in the calibration time (which is a one-time preprocessing) we can reduce the size of the robust prediction sets, while maintaining the same test-time cost. Additionally, we propose a multi-sample version of the method, which works with significantly fewer samples than previous sample-heavy approaches, but reduces the stochasticity of the method.

The corresponding publications are as following:

Robust Yet Efficient Conformal Prediction Sets ICML 2024

S. H. Zargarbashi, M. S. Akhondzadeh, A. Bojchevski

Reference to Section: § 5

Standard robust CP via randomised smoothing uses mean-based certified bounds on conformity scores, which discard distributional information and yield loose certified intervals. This directly translates to larger prediction sets. We propose CAS that leverages the *entire CDF* of the smoothed score distribution, leading to substantially tighter bounds and smaller prediction sets at the same robustness level. While previous works only target evasion, we provide robustness to feature and / or label corruptions as well. Our method is the first to provide robustness guarantees against all three types of perturbations.

This paper is presented at ICML 2024. I am the corresponding author of this publication.

Contributions: The motivation of this project was originally proposed by *Aleksandar Bojchevski*. I was responsible for the theoretical development, implementation, and experiments, and the writing of the manuscript. All developements, writings, and theory are done in collaboration with my

supervisor, and my co-author *Aleksandar Bojchevski*. *M. S. Akhondzadeh* scaled the experiments to ImageNet dataset, and provided valuable help in development of the code, in addition to great insights in all stages of the project.

Supplementary: Supplementary Material: § C

Robust Conformal Prediction with a Single Binary Certificate

ICLR 2025

S. H. Zargarbashi, A. Bojchevski

Reference to Section: § 6

Prior robust CP methods require *continuous* certificates of the conformity score, which limits the applicable robustness toolbox and imposes high Monte Carlo costs – compared to binary functions, continuous certificates need more MC samples to attain same tightness. We propose BinCP a binarization technique over the score space which allows to use binary certificates. Replacing continuous certificates with binary ones significantly reduces the required sample rate for the same set size. We reduce the MC sampling budget by an order of magnitude. Moreover, the binarization allows us to use any existing binary certificates for free, allowing to adapt for various threat models, and smoothing distributions. Empirically, we attain the same set size of the state of the art using ~2,000 samples only with a budget of ~150 samples.

This paper is presented at ICLR 2025. I am the corresponding author of this publication.

Contributions: I am the corresponding author of this publication, responsible for the theoretical development, implementation, experiments, and the writing of the manuscript. All stages of the project inlcluding the theoretical development, writing the manuscript, and testing the implementations were done in collaboration with my supervisor, *Aleksandar Bojchevski*.

Supplementary: Supplementary Material: § D

One Sample is Enough to Make Conformal Prediction Robust

NeurIPS 2025

S. H. Zargarbashi, M. S. Akhondzadeh, A. Bojchevski

Reference to Section: § 7

We propose RCP1, a robust CP method that requires only a single Monte Carlo sample per calibration and test point. By applying the randomized smoothing certificate directly to the coverage probability of the test point, and leveraging the convexity of the certificate, we can use Jensen's inequality to obtain robustness certificate on the known value $1 - \alpha$ without any estimation. The prediction sets obtained by RCP1 are comparable to those of the state-of-the-art robust CP method with 150 samples per point, but with a test-time cost of a single forward pass. This is the first robust CP method that eliminates the Monte Carlo overhead at test time while using randomized smoothing certificates.

This paper is presented at NeurIPS 2025. I am the corresponding author of this publication.

Contributions: I was responsible for the theoretical development, implementation, experiments, and the writing of the manuscript. All stages of the project inlcluding the theoretical development, writing the manuscript, and testing the implementations were done in collaboration with my

supervisor, *Aleksandar Bojchevski*. *M. S. Akhondzadeh* provided valuable help in the implementation, in addition to scaling the experiments to ImageNet dataset, vision transformer models, and providing great insights in all stages of the project.

Supplementary: Supplementary Material: § E

Front-Loaded Robust Conformal Prediction: Heavy Calibration, Minimal Test-Time Cost

ICML

2026 (Under review)

S. H. Zargarbashi, M. S. Akhondzadeh, A. Bojchevski

Under review — preprint available soon

We propose Flor$^{(1)}$, a front-loaded robust CP method that maintains the same test-time cost as a single model forward pass (exactly as RCP1). By performing a heavy Monte Carlo estimation during the calibration phase, we circumvent the over-conservativeness of the Jensen's inequality used in RCP1, leading to significantly smaller prediction sets. Moreover, we propose a multi-sample version of the method, which works with significantly fewer samples than previous sample-heavy approaches (e.g. BinCP), but reduces the stochasticity of the single sample approaches. Flor$^{(k)}$ works with test-time sample rates as low as $\sim$21, and 51 samples, while maintaining significanly lower variance and smaller prediction sets than RCP1.

This paper is under review for ICML 2026. I am the corresponding author of this publication.

Contributions: I am the corresponding author of this publication, responsible for the theoretical development, implementation, experiments, and the writing of the manuscript. All stages of the project inlcluding the theoretical development, writing the manuscript, and testing the implementations were done in collaboration with my supervisor, *Aleksandar Bojchevski*, and my co-author *M. S. Akhondzadeh*, who provided valuable help in the implementation.

Supplementary: Supplementary Material: § F

Publications

This thesis is based on the following publications. Additional work produced during the doctoral studies is listed thereafter.

Publications included in this thesis

[1] **S. H. Zargarbashi**, S. Antonelli, and A. Bojchevski. *Conformal Prediction Sets for Graph Neural Networks.* In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, PMLR 202, pp. 12292–12318, 2023.

[2] **S. H. Zargarbashi** and A. Bojchevski. *Conformal Inductive Graph Neural Networks.* In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[3] **S. H. Zargarbashi**, M. S. Akhondzadeh, and A. Bojchevski. *Robust Yet Efficient Conformal Prediction Sets.* In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, PMLR 235, pp. 17123–17147, 2024.

[4] **S. H. Zargarbashi** and A. Bojchevski. *Robust Conformal Prediction with a Single Binary Certificate.* In *The Thirteenth International Conference on Learning Representations (ICLR)*,

2025.

[5] **S. H. Zargarbashi**, M. S. Akhondzadeh, and A. Bojchevski. *One Sample is Enough to Make Conformal Prediction Robust.* In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[6] **S. H. Zargarbashi**, M. S. Akhondzadeh, and A. Bojchevski. *Front-Loaded Robust Conformal Prediction: Heavy Calibration, Minimal Test-Time Cost.* In *Under review for 43rd International Conference on Machine Learning (ICML)*, 2026.

Additional publications These publications are not included in this thesis, but they were produced during the doctoral studies. They are listed here for completeness and to provide a broader context of the research activities during the PhD.

[7] A. Javanmardi, **S. H. Zargarbashi**, S. M. A. R. Thies, W. Waegeman, A. Bojchevski, and E. Hüllermeier. *Optimal Conformal Prediction under Epistemic Uncertainty.* Preprint, arXiv:2505.19033, 2025.

[8] M. S. Akhondzadeh, **S. H. Zargarbashi**, J. Cao, and A. Bojchevski. *EvA: Evolutionary Attacks on Graphs.* In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

[9] S. L. Chau, **S. H. Zargarbashi**, Y. Sale, and M. Caprio. *Quantifying Epistemic Predictive Uncertainty in Conformal Prediction.* Under Review for *the 43rd International Conference on Machine Learning (ICML)*, 2026.

[10] M. S. Akhondzadeh, **S. H. Zargarbashi**, S. Antonelli, and A. Bojchevski. *CATS: Conformalized Adaptive Test-Time Scaling.* Under Review for *the 43rd International Conference on Machine Learning (ICML)*, 2026.

2 Background

This chapter lays the mathematical foundations for the methods developed in the thesis. We cover three core topics: *conformal prediction* (§ 2.1) a post-hoc method producing valid prediction sets with finite-sample guarantees; *graph neural networks and exchangeability* (§ 2.2) which is a family of models for graph-structured data; and *randomized smoothing* (§ 2.3), which is a well-known framework for robustness certificates against adversarial perturbations.

Note that since this thesis is in cumulative format – i.e., it consists of several self-contained papers – the notation might differ slightly across chapters. We have made an effort to unify notation as much as possible, but some variations are inevitable due to the different contexts. Each chapter contains a self-contained notation summary.

2.1 Conformal Prediction

2.1.1 Setup and Exchangeability

In most parts of this work we focus on the standard classification setting. Let $\mathcal{X}$ be the feature space and $\mathcal{Y} = \{1, \ldots, K\}$ the finite label space. We observe a data sequence

$$(X_1, Y_1),\ (X_2, Y_2),\ \ldots,\ (X_{n+1}, Y_{n+1}), \tag{1}$$

and assume it is *exchangeable*, meaning that for every permutation π of $\{1, \ldots, n+1\}$,

$$\big((X_1, Y_1), \ldots, (X_{n+1}, Y_{n+1})\big) \overset{d}{=} \big((X_{\pi(1)}, Y_{\pi(1)}), \ldots, (X_{\pi(n+1)}, Y_{\pi(n+1)})\big). \tag{2}$$

Intuitively exchangeability means that the order of the data points does not matter: the joint distribution is invariant under permutations of the indices. More importantly, exchangeability is implied by, but strictly weaker than, the i.i.d. assumption – it allows for many forms of structured dependence so long as the indexing of the variables carries no information about their values.

In the *split* (or *inductive*) conformal setting, the first n points form a held-out calibration set $\mathcal{D}_{\text{cal}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ that is disjoint from the data used to train the underlying predictor, and $(\boldsymbol{x}_{n+1}, y_{n+1})$ is the test point whose label is to be predicted. Notably for the case of graph node-classification (specially in the transductive setting), the calibration set can also be seen by the model during training (without their labels). The only requirement is that the model does not behave differently on the calibration nodes compared to the test nodes or have any bias in its knowledge that is asymmetric between these two sets.

Important notation and terminology. Note that there are several other conformal prediction setups that balance computational and statistical efficiency; examples of these frameworks are JackKnife+ (Barber et al., 2019b), CV+, and *full* (or *transductive*) setting, where for each test point and each potential label the model is retrained assuming that label is correct. Notably our methods are not limited to the choice of this setup but throughout the thesis we only focus on the split conformal setting, which is the most computationally efficient and widely used framework. Note that the terms transductive and inductive are used both in the context of conformal prediction and graph neural networks. In this thesis we only use them to specify the graph neural network training and evaluation setting, and by the term conformal prediction we always refer to the split conformal setting, unless otherwise specified.

2.1.2 Conformity Scores

Conformal prediction constructs a prediction set $C_\alpha(\boldsymbol{x}_{n+1}) \subseteq \mathcal{Y}$ for the test input $\boldsymbol{x}_{n+1}$ that contains the true label y_{n+1} with probability at least $1-\alpha$. The prediction sets are defined through a *conformity score* that measures how well a candidate label y conforms to the input $\boldsymbol{x}$ according to a pre-trained model f. The *conformity score* is any measurable function $s : \mathcal{X} \times \mathcal{Y} \longrightarrow \mathbb{R}$, that assigns a real-valued score to each input-label pair. Note that in the many works in the literature, the term *nonconformity score* is often used instead, which is simply the negative of the conformity score. The rest of the setup is also valid for both definition; we only take this definition as it is more intuitive for the classification task.

The Conformal Guarantee. Given a calibration set $\mathcal{D}_{\text{cal}}$ and a target miscoverage level $\alpha \in (0,1)$, define the *conformal threshold*

$$q_\alpha \;=\; \text{Quantile}\Big(\alpha;\; \big\{s(\boldsymbol{x}_i, y_i)\big\}_{i=1}^{n} \cup \{+\infty\}\Big), \tag{3}$$

where $\text{Quantile}(\alpha; \mathcal{A})$ denotes the $\lceil \alpha(|\mathcal{A}|)\rceil$-th order statistic of the multiset $\mathcal{A}$. The *conformal prediction set* for a test input $\boldsymbol{x}_{n+1}$ is then

$$C_\alpha(\boldsymbol{x}_{n+1}) \;=\; \big\{y \in \mathcal{Y} : s(\boldsymbol{x}_{n+1}, y) \geq q_\alpha\big\}. \tag{4}$$

The fundamental guarantee of conformal prediction is as follows.

Theorem 2.1 (Split conformal coverage (Vovk et al., 2005))**.** *Let s be any measurable conformity score function. If $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n+1}$ is exchangeable, then*

$$\Pr\big[y_{n+1} \in C_\alpha(\boldsymbol{x}_{n+1})\big] \;\geq\; 1-\alpha. \tag{5}$$

Moreover, if the scores are almost surely distinct, then the upper bound also holds:

$$1-\alpha \;\leq\; \Pr\big[y_{n+1} \in C_\alpha(\boldsymbol{x}_{n+1})\big] \;\leq\; 1-\alpha+\frac{1}{n+1}. \tag{6}$$

Proof sketch. Consider the rank statistic $R = |\{i \leq n+1 : s(\boldsymbol{x}_i, y_i) \leq s(\boldsymbol{x}_{n+1}, y_{n+1})\}|$. Under exchangeability, R is uniformly distributed over $\{1, \ldots, n+1\}$. The event $\{y_{n+1} \notin C_\alpha(\boldsymbol{x}_{n+1})\}$ equals $\{R \leq \lfloor \alpha(n+1) \rfloor\}$, which has probability at most α. □

Several aspects of Theorem 2.1 deserve emphasis. First, the guarantee is *distribution-free*: no parametric assumption on the data distribution is required. Second, it is *model-agnostic*: the choice of f, and therefore of s, does not affect the validity of Eq. 5. This means that CP can be used as a post-hoc wrapper around any pre-trained model and return guaranteed prediction sets for free. The model and score enter only through *efficiency*, i.e., the average size of $C_\alpha(\boldsymbol{x})$; a more informative score yields smaller sets while preserving the same coverage level – intuitively with a fixed score function, if the accuracy of the underlying model improves, the average set size also reduces.

Usually, the score is computed from a pre-trained model f. In following, we discuss two common choices of conformity scores for classification.

Threshold Prediction Sets (TPS). The simplest choice sets $s(\boldsymbol{x}, y) = \pi_y(\boldsymbol{x})$, where $\pi(\boldsymbol{x}) = (\pi_1(\boldsymbol{x}), \ldots, \pi_K(\boldsymbol{x})) \in \Delta^K$ is the softmax output of f. While TPS is both simple and yields small sets on average, it's coverage distribution is biased towards easy inputs: it over-covers easier inputs, and under-covers harder ones, even if the softmax probabilities are well-calibrated.

Importantly, the coverage guarantee in conformal prediction is *marginal* over the data distribution – averages over all future test points. A more desirable property is *conditional* coverage, which requires that the coverage guarantee holds similarly for each individual test point. Formally

$$\Pr[y_{n+1} \in C_\alpha(\boldsymbol{x}_{n+1}) \mid \boldsymbol{x}_{n+1}] \;\geq\; 1 - \alpha \tag{7}$$

Without unrealistic assumptions, this property is impossible to achieve as shown by Angelopoulos et al. (2024), but it motivates the design of scores that are more adaptive to the difficulty of the instance.

Adaptive Prediction Sets (APS). Romano et al. (2020) propose

$$s_{\text{APS}}(\boldsymbol{x}, y) \;=\; -\Big(\rho(\boldsymbol{x}, y) + u \cdot \pi_y(\boldsymbol{x})\Big), \quad \rho(\boldsymbol{x}, y) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x})\, \mathbf{1}\big[\pi_k(\boldsymbol{x}) > \pi_y(\boldsymbol{x})\big], \tag{8}$$

where $u \sim \text{Uniform}(0, 1)$ added to break ties. If the oracle label distribution is provided, then APS returns the smallest sets that satisfy conditional coverage for each instance (even without a calibration set by directly setting $q_\alpha = \alpha - 1$); this property does not hold for TPS even after calibration. Given a noisy and uncalibrated probability distribution we use the calibration set to account for the discrepancy. Intuitively for each label, the score is capturing the cumulative softmax probabilities the label and all labels with higher probability mass. After the calibration, the quantile answers the question of "how much probability mass should be accumulated to achieve the desired coverage level?" marginal to the data distribution. The addition of the uniform random variable u has another effect: it allows for exact $1 - \alpha$ coverage. Given the true distribution let for label y to be at the border of $1 - \alpha$ (e.g. = 90%) cumulative probability mass. Without y the set adds up to 89% and with y it adds up to 91%. The addition of u allows for a smooth interpolation between these two cases, so that the set is exactly at the desired coverage level. Note that the negative sign is used to convert the score from non-conformity to conformity (the original definition in Romano et al. (2020) does not have the negative sign). Conclusively, this score function spreads the coverage more evenly across the data distribution, it usually results in larger sets compared to TPS, but it is more reliable.

2.1.3 Notes about Conformal Prediction

Non-trivial evaluation The coverage is theoreticaly lower *upper* bounded. Notably CP solves a threshold optimization that results in the smallest average set size that satisfies the coverage requirement. This means that in some cases where the nominal $1 - \alpha$ is lower than model's accuracy (e.g. setting $1 - \alpha = 0.9$ for a model with 95% accuracy), the conformal procedure can return singleton sets for most test points and leave the rest with empty sets. Therefore, comparing the choice of score function at a trivial nominal coverage-levels is not meaningful. This is why in some experiments (like § 3.5) we choose α adaptive to the model's accuracy, and we compare different datasets and models at different coverage levels.

Coverage distribution The guarantee in Theorem 2.1 is probabilistic. Formally $\Pr[y_{n+1} \in C_\alpha(\boldsymbol{x}_{n+1})]$ itself is random over the choice of the calibration set and it comes from a Beta distribution $\text{Beta}(n+1-\lfloor\alpha(n+1)\rfloor, \lfloor\alpha(n+1)\rfloor)$. Here the variable n which is the size of the calibration set controls the variance of the coverage distribution. Increasiong the size of this calibration set makes the coverage probability more concentrated around the nominal level $1-\alpha$. In practice, we are more interested in smaller calibration sets as we are mostly in label-scarce settings. In our graph experiments the size of the calibration set is around 150 nodes, and in image classification examples it can increase up to $500 \sim 1000$ samples. As we report our results over multiple random splits, the effect of this parameter is negligible.

2.1.4 Conformal Risk Control

Conformal prediction is a special case of a more general framework called conformal risk control (CRC) (Angelopoulos et al., 2022). Let a process be assigned with a conservativeness parameter $\lambda \in \Lambda$ and a risk function $L : \mathcal{X} \times \Lambda \to \mathbb{R}$. Intuitively, increasing the parameter λ increases the conservativeness of the process, and therefore reduces the risk. Given a calibration set $\mathcal{D}_{\text{cal}}$, CRC finds a parameter $\hat{\lambda}$ such that the expected risk over the future test points is at most below a user-specified tollerance level α.

Theorem 2.2 (Conformal risk control (Angelopoulos et al., 2022))**.** *For any nominal tollerance level α, let*

$$\hat{\lambda} = \inf\{\lambda \in \Lambda : \frac{1}{n+1}\left(\sum_{i=1}^{n} L(X_i, \lambda) + b\right) \leq \alpha\} \tag{9}$$

is the empirical risk on the calibration set. If $\{X_i\}_{i=1}^{n+1}$ is exchangeable, and $L(\cdot, \lambda)$ is monotone non-increasing in λ, right-continuous, upper bounded by some constant b, and $L(\cdot, \max\Lambda) \leq \alpha$ then

$$\mathbb{E}\big[L(X_{n+1}, \hat{\lambda})\big] \;\leq\; \alpha. \tag{10}$$

Setting the risk function to miscoverage of the true label, i.e $L((\boldsymbol{x}, y), \lambda) = \mathbf{1}[y \notin C_\lambda(\boldsymbol{x})]$, derives the standard conformal prediction guarantee, however, this framework allows for more general risk functions, such as the false positive rate of a certain class in image segmentation or the coverage under noise.

2.2 Graph Neural Networks and Exchangeability on Graphs

2.2.1 Graph Neural Networks

In graph node-classificaiton, the task is to assign a label to each node in the graph which is assigned with a feature vector $\boldsymbol{x}_v$, and is connected to other nodes. Formally, a graph $G = (\boldsymbol{X}, \boldsymbol{A})$ consists of a node feature matrix $\boldsymbol{X} \in \mathbb{R}^{N \times d}$ (a concatenation of the node features $\boldsymbol{x}_v$) and an adjacency matrix $\boldsymbol{A} \in \{0,1\}^{N \times N}$, where N is the number of nodes and d is the feature dimension. We denote the node set by $\mathcal{V} = \{1, \ldots, N\}$ and write $\mathcal{N}(v) = \{u : A_{uv} = 1\}$ for the neighbourhood of node v.

Graph neural networks (GNNs) are a family of models that in addition to usual feature processing layers, they also contain graph-specific layers that propagate information across the graph structure. This mechanism is often called *message-passing*, since it can be interpreted as nodes sending messages to their neighbors and updating their

representations based on the received messages. Formally, message passing at layer ℓ consists of two steps: first, each node v aggregates the representations of its neighbors from the previous layer $\ell - 1$ into a message $\boldsymbol{m}_v^{(\ell)}$; then, it updates its own representation $\boldsymbol{h}_v^{(\ell)}$ based on the aggregated message and its previous representation:

$$\boldsymbol{m}_v^{(\ell)} = \mathrm{AGG}^{(\ell)}\Big(\big\{\boldsymbol{h}_u^{(\ell-1)} : u \in \mathcal{N}(v)\big\}\Big), \tag{11}$$

$$\boldsymbol{h}_v^{(\ell)} = \mathrm{UPDATE}^{(\ell)}\Big(\boldsymbol{h}_v^{(\ell-1)},\ \boldsymbol{m}_v^{(\ell)}\Big), \tag{12}$$

with $\boldsymbol{h}_v^{(0)} = \boldsymbol{x}_v$ (the v-th row of $\boldsymbol{X}$). The functions $\mathrm{AGG}^{(\ell)}$ and $\mathrm{UPDATE}^{(\ell)}$ are learned; AGG is typically a permutation-invariant aggregator (e.g. sum, mean, or max). Same as other networks, at the last layer, a node classifier applies a linear head $\boldsymbol{W} \in \mathbb{R}^{K \times d_L}$ followed by softmax:

$$\pi_v = \mathrm{softmax}\big(\boldsymbol{W}\,\boldsymbol{h}_v^{(L)}\big) \in \Delta^K. \tag{13}$$

Notably by increasing the number of layers capable of message passing, each node will process information from a larger neighborhood. The term "receptive field" is used to describe the set of nodes that can influence a given node's representation after a certain number of layers.

While the literature of GNNs is vast and growing, we do not focus on the architecture design of the GNNs in this thesis. As our approach uses the models as black-box, the only takeaway from this section is the permutation equivariance property of GNNs. Nevertheless, we briefly describe one canonical GNN architecture for better intuition and clarity.

Graph Convolutional Network (GCN). The GCN (Kipf and Welling, 2017) instantiates equation 11–equation 12 as

$$\boldsymbol{H}^{(\ell)} = \sigma\big(\hat{\boldsymbol{A}}\,\boldsymbol{H}^{(\ell-1)}\,\boldsymbol{W}^{(\ell)}\big), \qquad \hat{\boldsymbol{A}} = \tilde{\boldsymbol{D}}^{-1/2}\,\tilde{\boldsymbol{A}}\,\tilde{\boldsymbol{D}}^{-1/2}, \tag{14}$$

where $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ is the adjacency matrix with self-loops, $\tilde{\boldsymbol{D}}_{vv} = \sum_u \tilde{A}_{vu}$ is the corresponding degree matrix, and σ is a pointwise nonlinearity. Multiplying by $\hat{\boldsymbol{A}}$ propagates the representation of each node to its one-hop neighborhood.

Permutation equivariance. A key structural property of GNNs for our purposes is *permutation equivariance*: if the node indices are permuted by π, then the output predictions are permuted by the same π. Formally, if $\boldsymbol{P}_\pi$ denotes the corresponding permutation matrix, then

$$f(\boldsymbol{P}_\pi \boldsymbol{X},\ \boldsymbol{P}_\pi \boldsymbol{A} \boldsymbol{P}_\pi^\top) \;=\; \boldsymbol{P}_\pi\, f(\boldsymbol{X}, \boldsymbol{A}). \tag{15}$$

Message-passing GNNs with permutation-invariant aggregators are permutation equivariant by construction. This property is the bridge connecting GNN predictions to the exchangeability requirement of conformal prediction.

2.2.2 Semi-Supervised Node Classification Settings

In semi-supervised node classification, the node set $\mathcal{V}$ is partitioned into labeled nodes $\mathcal{V}_l$ and unlabeled nodes $\mathcal{V}_u$, with $\mathcal{V} = \mathcal{V}_l \cup \mathcal{V}_u$. The labeled nodes are further split into a development set $\mathcal{V}_d$(training and validation) and a calibration set $\mathcal{V}_c$, so that $\mathcal{V}_l = \mathcal{V}_d \cup \mathcal{V}_c$. We consider three settings that differ in how much of the graph

is observed during training. In semi-supervised node classification, the model might have access to the features of a subset of (unlabeled) nodes. While our framework and methods are indifferent to the setup where the model is trained, our main attention is on the graph where the calibration happens, if the test nodes are visible at the time of the calibration.

Transductive setting. The model f is trained on the full graph $G = (\boldsymbol{X}, \boldsymbol{A})$ using labels only from $\mathcal{V}_d$. All nodes in $\mathcal{V}_c \cup \mathcal{V}_u$ are visible during training (their features and edges are available), but their labels are withheld. This is the standard *semi-supervised* setting for citation networks. In short, the graph over which we calibrate is the same graph over which we run the inference and compute the prediction sets.

Sequential inductive setting. Training uses only $\mathcal{V}_d$; calibration is performed on the graph induced by $\mathcal{V}_d \cup \mathcal{V}_c$. Test nodes then arrive sequentially (one-by-one or in batches), potentially entering the receptive fields of calibration nodes and thereby altering their embeddings. It is not necessary for all the test-nodes to be unseen during the calibration. Even if a small subset of the test nodes are not in the graph during the calibration, still the effect is the same. Notably, in this setup, the prediction of the already existing nodes can also differ with or without the unseen test nodes. This makes the setup more challenging for conformal prediction.

Simultaneous inductive setting. The model is trained on the subgraph induced by $\mathcal{V}_d$. After training, the nodes $\mathcal{V}_c \cup \mathcal{V}_u$ and all their edges are revealed at once, and calibration proceeds on the enlarged graph. In short this setup is similar to the transductive setting from the CP's perspective, since between the calibration and inference, the graph does not change. The main difference is in the accuracy and the scores of the model, which will only affect the efficiency of the prediction sets, but not their validity.

In all setups we assume that the calibration nodes are randomly sampled from the node set $\mathcal{V}$. All GNN architectures we consider are also permutation equivariant. In the (sequential) inductive setup however, there is another important element: how the updates in the graph structure occur?

Node-exchangeable and edge-exchangeable graph models. While we deffer the detailed discussion to § 4.2, we briefly discuss the notion of exchangeability in graph updates. We assume the generation of the graph to start from an empty graph, and through each update, new nodes or edges apprear. Therefore we assume that the calibration is done after t_{cal} steps where calibration nodes are entirely visible – still there is the chance that the edge structure related to these nodes are not final at the time of calibration. We call a sequence of updates *node-evolving* if at each step, the new node (randomly selected from the ultimate graph) appears with all the edges it has in the current set of visible nodes. Therefore, the status of edge between two existing nodes will not change in future updates. In contrast, in an *edge-evolving* sequence of updates, at each step, a new edge appears between two nodes. The new edge can either connect two existing nodes, or introduce new node(s) to the graph. A node-evolving sequence is called *node-exchangeable* if the distribution of updates is invariant under permutations of the node labels. Similarly, an edge-evolving sequence is called *edge-exchangeable* if the distribution of updates is invariant under permutations of the edge labels. Real-life examples of node-evolving sequences include citation networks, where new papers (nodes) appear with all their citations (edges) at once. An example of edge-evolving

sequence is a social network, where new friendships (edges) can form between existing users (nodes) over time.

2.3 Robustness and Randomized Smoothing

A common assumption in machine learning is that the test model learns patterns from the training data, and during test, it matches the learned patterns to the test input to make a prediction. This suggests that adding a very small magnitude of noise to the test input should not change the prediction. However, it has been shown that for many models, especially deep neural networks, this assumption does not hold: an adversary can add a small (imprecetible) perturbation to the test input and change the prediction to an arbitrary target label. This phenomenon is known as *adversarial attack*. Adversarial attacks can represent a real threat in safety-critical applications, assuming the potential adversaries using the system as well, or they can be a very good proxy to the worst-case natural noise.

2.3.1 Threat Models

In classification task, the adversary's goal is to cause misclassification by perturbing the test input. The motivation can be either to cause misclassification to any wrong label (untargeted attack) or to a specific target label (targeted attack).

Tailoring the threat model to the representation of the output, in conformal prediction however, the goal of the adversary is to cause miscoverage, i.e., to cause the true label to be outside the prediction set. This is a more general goal than misclassification. The ultimate goal is to reduce the empirical coverage from the nominal guaranteed level $1 - \alpha$ to a much lower level. Practically, the vanilla conformal prediction can be very vulnerable with its coverage dropping to nearly zero with a very small magnitude of perturbation (see Fig. 7.1).

The adversary can achieve this goal by perturbing the test input (evasion attack) post-calibration, introducing noise in the calibration set (poisoning attack), or both. During the calibration phase, the adversary can either perturb the features of the calibration data, or flip their labels. While in the evasion attack, the adversary can only perturb the features of the test input. In either cases, the constraint for the adversary is to keep the perturbation small, so that it is not easily detectable. In label-poisoning attach that translates to a small number of label flips, while in feature perturbations it translates to keeping the perturbed input within a small "perturbation ball" around the clean input.

Perturbation ball. The adversary perturbs the test input $\boldsymbol{x}_{n+1}$ within a *threat ball* $\mathcal{B}(\boldsymbol{x}) \subseteq \mathcal{X}$. For continuous data, the usual choice is the ℓ_2 ball of radius r:

$$\mathcal{B}_r(\boldsymbol{x}) = \left\{\tilde{\boldsymbol{x}} \in \mathcal{X} : \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq r\right\}. \tag{16}$$

For sparse binary data (e.g. graph adjacency vectors), Bojchevski et al. (2020) suggest an asymmetric bit-flip ball:

$$\mathcal{B}_{r_a, r_d}(\boldsymbol{x}) = \left\{\tilde{\boldsymbol{x}} \in \{0,1\}^d : \sum_{j=1}^{d} \mathbf{1}[\tilde{x}_j = x_j + 1] \leq r_a,\ \sum_{j=1}^{d} \mathbf{1}[\tilde{x}_j = x_j - 1] \leq r_d\right\}, \tag{17}$$

where r_a bounds the number of zero-bits that may be activated (additions) and r_d bounds the number of one-bits that may be deleted (deletions). For the evasion attack

we assume that every test input is potentially perturbed.

Poisoning attack. The adversary can at most perturb k calibration inputs. For a clean calibration set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, define the dataset-level threat set

$$\mathfrak{B}_{k,\mathcal{B}}(\mathcal{D}) = \left\{ \tilde{\mathcal{D}} = \{(\tilde{x}_i, y_i)\}_{i=1}^n : \tilde{x}_i \in \mathcal{B}(x_i),\ \sum_{i=1}^n \mathbf{1}[\tilde{x}_i \neq x_i] \leq k \right\}. \tag{18}$$

Under label poisoning, the adversary may instead flip up to k calibration labels. Combined evasion-and-poisoning attacks can be modelled by applying both threat sets simultaneously.

Inverse threat ball. While addressing robustness, there are two perspectives to consider: if we are given the clean point and we want to know search for the worst-case perturbation within the threat ball, or if we are given a potentially perturbed point and we want to search for the clean point that can be perturbed to the observed one. The former is the usual perspective in the literature of adversarial robustness, while in CP, based on the framework of robustness we might need to take the latter perspective as well.

In the second view, during test time, one observes a potentially perturbed input $\tilde{x} \in \mathcal{B}(x)$ without knowing the clean x. To reason about the clean point from the observed perturbation, define the *inverse ball*

$$\mathcal{B}^{-1}(\tilde{x}) = \{x : \tilde{x} \in \mathcal{B}(x)\}, \tag{19}$$

the smallest set guaranteed to contain the clean input given any perturbation in $\mathcal{B}$. For symmetric balls such as ℓ_p norms, $\mathcal{B}^{-1} = \mathcal{B}$. For asymmetric balls like sparsity aware threat model, $\mathcal{B}^{-1}_{r_a,r_d} = \mathcal{B}_{r_d,r_a}$: the roles of additions and deletions are swapped.

Later in § 5, we discuss two equivalent frameworks to apply robust CP. One bounds the worst-case scores given the clean calibration points, and the other upper bounds the score of the invisible clean point given the observed perturbed test point. The first framework works with the usual threat ball $\mathcal{B}$, while the second framework works with the inverse ball $\mathcal{B}^{-1}$.

2.3.2 Robust Conformal Prediction (RCP)

Gendler et al. (2021) formalise the notion of adversarially robust coverage.

Definition 2.3 (Robust $1-\alpha$ coverage)**.** Prediction sets $\{C_\alpha(\cdot)\}$ have *adversarially robust* $1-\alpha$ *coverage* with respect to threat ball $\mathcal{B}$ if

$$\Pr\left[y_{n+1} \in C_\alpha(\tilde{x}_{n+1}) \text{ for all } \tilde{x}_{n+1} \in \mathcal{B}(x_{n+1})\right] \ \geq\ 1-\alpha, \tag{20}$$

where the probability is over the joint draw of $\mathcal{D}_{\text{cal}} \cup \{(x_{n+1}, y_{n+1})\}$ under the exchangeable distribution. Intuitively this guarantee asserts that the coverage probability of RCP over the potentially perturbed test input $\tilde{x}_{n+1}$ is higher than the coverage probability of the vanilla CP over the clean test input x_{n+1}. Note that this statement is conditional to each test input. While the marginal coverage guarantee of CP is above $1-\alpha$, this statement concludes the robust coverage guarantee of RCP is above $1-\alpha$ as well.

This definition is only meaningful for deterministic prediction sets. In many cases (e.g. APS), the prediction sets are defined randomly (e.g. by the random variable u to break ties, or randomized augmentation of the input). For random sets the adversary can does not access to the internal randomness of the prediction set, and therefore the goal changes to reduce the expected coverage over the randomness of the prediction set. With the redefinition of the threat model, the robust coverage guarantee can be rephrased as follows:

$$\mathbb{E}_{\mathcal{D}_{n+1}}[\min_{\tilde{\boldsymbol{x}}_{n+1}\in\mathcal{B}_r(\boldsymbol{x}_{n+1})} \Pr_u[y_{n+1} \in C_r(\tilde{\boldsymbol{x}}_{n+1}; u)]] \geq \Pr_{\mathcal{D}_{n+1},u}[y_{n+1} \in C_0(\boldsymbol{x}_{n+1}; u)] \geq 1-\alpha \tag{21}$$

The key idea for achieving robust coverage is to replace exact conformity scores with *certified bounds* on their worst-case values within the threat ball.

Definition 2.4 (Certified score bounds)**.** For a conformity score s and threat ball $\mathcal{B}$, define:

$$\underline{s}(\boldsymbol{x}, y; \mathcal{B}) \;\leq\; \inf_{\tilde{\boldsymbol{x}}\in\mathcal{B}(\boldsymbol{x})} s(\tilde{\boldsymbol{x}}, y), \tag{22}$$

$$\overline{s}(\boldsymbol{x}, y; \mathcal{B}) \;\geq\; \sup_{\tilde{\boldsymbol{x}}\in\mathcal{B}(\boldsymbol{x})} s(\tilde{\boldsymbol{x}}, y). \tag{23}$$

These bounds can be used in two complementary ways to achieve robust coverage.

Theorem 2.5 (Robust conformal prediction (Gendler et al., 2021; Zargarbashi et al., 2024))**.** *Assume $\mathcal{D}_{\text{cal}} \cup \{(\boldsymbol{x}_{n+1}, y_{n+1})\}$ is exchangeable. Let $\underline{s}$ and $\overline{s}$ be valid certified bounds as in Definition 2.4.*

1. ***Calibration-time robustness.*** *Define*

$$q_\alpha^{\downarrow} = \text{Quantile}\Big(\alpha;\ \big\{\underline{s}(\boldsymbol{x}_i, y_i; \mathcal{B})\big\}_{i=1}^{n} \cup \{+\infty\}\Big), \tag{24}$$

and the prediction set $C_\alpha^{\text{cal}}(\tilde{\boldsymbol{x}}) = \{y : s(\tilde{\boldsymbol{x}}, y) \geq q_\alpha^{\downarrow}\}$. Then

$$\Pr\big[y_{n+1} \in C_\alpha^{\text{cal}}(\tilde{\boldsymbol{x}}_{n+1}) \text{ for all } \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})\big] \geq 1-\alpha. \tag{25}$$

2. ***Test-time robustness.*** *Let q_α be the standard conformal threshold equation 3 computed from clean calibration scores. Define $C_\alpha^{\text{test}}(\tilde{\boldsymbol{x}}) = \{y : \overline{s}(\tilde{\boldsymbol{x}}, y; \mathcal{B}^{-1}) \geq q_\alpha\}$. Then*

$$\Pr\big[y_{n+1} \in C_\alpha^{\text{test}}(\tilde{\boldsymbol{x}}_{n+1}) \text{ for all } \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})\big] \geq 1-\alpha. \tag{26}$$

Both constructions are *conservative*: the robust sets C^{cal} and C^{test} are always supersets of the clean conformal set C_α. The conservativeness of RCP results in larger prediction sets, (and higher coverage probability). The ultimate goal in this setup is to maintain the robust coverage guarantee while keeping the prediction sets as small as possible – this also means that the objective is to keep the robust coverage probability higher, but as close as possible to the nominal level $1-\alpha$. The challenge of robust CP ultimately reduces to constructing *tight* certified bounds $\underline{s}$ and $\overline{s}$, and using these bounds in the most efficient way.

2.3.3 Randomized Smoothing

Randomized smoothing (Cohen et al., 2019) provides a general framework for certified robustness with only black-box access to the model or in our case the score function. By certified robustness, we mean to guarantee that the model's prediction will not change, or to guarantee that the model's score will remain within a certified interval under any perturbation within the threat ball (in classification) The core idea is to define a *smoothed* score by adding smoothing noise $\xi : \mathcal{X} \to \mathcal{X}$ (from a standard class of random noises), and using the push-forward distribution as the new output distribution. The final prediction of this framework can be any statistics of the smoothed distribution, such as the mean or the CDF – in case of classification it can be majority vote or any other thresholded statistic. Intuitively by moving from the clean $\boldsymbol{x}$ to the perturbed $\tilde{\boldsymbol{x}}$, there is still a large overlap between the two smoothed input distributions $\xi(\boldsymbol{x})$ and $\xi(\tilde{\boldsymbol{x}})$ which results in the smoothed scores changing slowly. As a result, we can propose tight certified bounds only considering the black-box statistics, and agnostic to the mechanics of the score function or the model.

Smoothing scheme. A smoothing scheme is a random map $\xi : \mathcal{X} \to \mathcal{X}$. A straightforward *smoothed conformity score* can be defined as the mean of the original score s under the smoothing distribution:

$$\hat{s}(\boldsymbol{x}, y) = \mathbb{E}\big[s\big(\xi(\boldsymbol{x}), y\big)\big]. \tag{27}$$

For continuous data, a common choice is Gaussian smoothing $\xi(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{\epsilon}$ for $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ where $\sigma > 0$ controls the smoothing radius.

For sparse binary data, Bojchevski et al. (2020) propose a bit-flip smoothing scheme $\xi(\boldsymbol{x}) = \boldsymbol{x} \oplus \boldsymbol{\delta}$ where $\oplus$ denotes componentwise XOR and $\delta_j \sim \text{Bernoulli}(p_{x_j})$ independently for each coordinate, with flip probabilities p_0 (for bits that are currently 0) and p_1 (for bits that are currently 1).

Mean-based certified bound. Here, the certified robustness can be achieved by bounding the worst-case smoothed score within the threat ball – i.e. bounding $\sup_{\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})} \hat{s}(\tilde{\boldsymbol{x}}, y)$ and/or $\inf_{\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})} \hat{s}(\tilde{\boldsymbol{x}}, y)$. The original optimization problem for the certificate is not easy to solve. The score function (which is coming from the model) is often complex, and non-convex. To circumvent this issue, we solve the optimization problem over the class of all measurable functions (classifiers or score functions); i.e.

$$\overline{s}(\tilde{\boldsymbol{x}}, y; \mathcal{B}) = \max \tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x}) \hat{s}(\tilde{\boldsymbol{x}}, y) \leq \max_{h \in \mathcal{H}, \tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})} \mathbb{E}\big[h(\xi(\tilde{\boldsymbol{x}}), y)\big] \tag{28}$$

This inequality holds because the score function is itself a measurable function, and therefore a member of $\mathcal{H}$. However, without any additional constraint, the right-hand side results in a trivial bound. To get a non-trivial bound, we need to impose constraints that uses black-box information about the score function, and limits the search space of h to functions that are consistent with the observed measure.

Gendler et al. (2021) following Kumar et al. (2020) derive bounds on the worst-case smoothed mean by solving the following linear program over the class $\mathcal{H}$ of all measurable functions $h : \mathcal{X} \times \mathcal{Y} \to [a, b]$ (with $[a, b] = [0, 1]$ for probability scores):

$$\overline{s}_{\text{mean}}(\tilde{\boldsymbol{x}}, y) = \max_{h \in \mathcal{H}} \mathbb{E}\big[h(\xi(\tilde{\boldsymbol{x}}), y)\big] \quad \text{s.t.} \quad \mathbb{E}\big[h(\xi(\boldsymbol{x}), y)\big] = \hat{s}(\boldsymbol{x}, y). \tag{29}$$

For Gaussian smoothing under ℓ_2 threat, this problem has the closed-form solution

$$\overline{s}_{\text{mean}}(\tilde{\boldsymbol{x}}, y) = \Phi_\sigma\Big(\Phi_\sigma^{-1}\big(\hat{s}(\boldsymbol{x}, y)\big) + r\Big)\,, \qquad \underline{s}_{\text{mean}}(\tilde{\boldsymbol{x}}, y) = \Phi_\sigma\Big(\Phi_\sigma^{-1}\big(\hat{s}(\boldsymbol{x}, y)\big) - r\Big)\,, \tag{30}$$

where Φ_σ is the CDF of $\mathcal{N}(0, \sigma^2)$ and $r = \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2$.

Switching from the model-aware optimization to the general problem over $\mathcal{H}$ yields a useful property: many cases (due to the symmetries in the smoothing distribution and the threat ball) we can replace $(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ with canonical points (e.g. the origin and a point on the boundary of the threat ball), and solve the optimization only over the choice of h. Specifically

- For the ℓ_2 ball $\mathcal{B}_r(\boldsymbol{x})$, one may set $\boldsymbol{x} = \mathbf{0}$ and $\tilde{\boldsymbol{x}} = [r, 0, \ldots, 0]^\top$ without loss of generality.
- For the asymmetric bit-flip ball $\mathcal{B}_{r_a, r_d}(\boldsymbol{x})$, one may set $\boldsymbol{x} = [1, \ldots, 1, 0, \ldots, 0]^\top$ with r_d ones, and $\tilde{\boldsymbol{x}} = \mathbf{1} - \boldsymbol{x}$ with r_a ones.

Intuitively this means that certificates only on the scalar statistics of s under ξ, not on the specific coordinates of $\boldsymbol{x}$. We discuss this more in § C.3, and subsubsection D.4.1.

Binary certificates. An important alternative is to work with *binary* certificates. In confidence certificates, the goal is to certify that a continuous statistic (e.g. the smoothed mean) remains within a certain interval. In contrast, binary certificates certify that a binary event (e.g. the model's prediction returning a certain label) holds with high probability under the smoothing distribution. One example usecase for binary certificate can be through thresholding the score function: let $b(\boldsymbol{x}, y) = \mathbf{1}[s(\boldsymbol{x}, y) \geq \tau]$ for an adaptive threshold τ, and certify the probability

$$\underline{p}(\boldsymbol{x}, y; r) \;\leq\; \Pr\big[s(\xi(\boldsymbol{x}), y) \geq \tau\big] \;=\; \mathbb{E}[b(\xi(\boldsymbol{x}), y)]\,. \tag{31}$$

The certificate in this context is a lower bound on the probability that the binary function b returns 1 under the smoothing distribution. In classification, we call a point *certified* if the probability of returning the top-1 label remains above $1/2$ within the threat ball. For Gaussian smoothing the binary certificate is the classical result of Cohen et al. (2019):

$$\underline{p}(\boldsymbol{x}, y; r) = \Phi\Big(\Phi^{-1}(\Pr[s(\xi(\boldsymbol{x}), y) \geq \tau]) - \frac{r}{\sigma}\Big)\,. \tag{32}$$

General recipe for deriving randomized smoothing certificates. Yang et al. (2020) provide a general recipe for deriving randomized smoothing certificates for any additive smoothing distribution and threat ball. A similar approach is provided by Lee et al. (2019) which solves the problem through a greedy search (equivalent to the fractional knapsack problem). The high-level idea is to partition the input space into regions of constant likelihood ratio $\frac{\Pr[\xi(x)=z]}{\Pr[\xi(\tilde{x})=z]}$, and then solve the optimization problem over the class of functions that are constant within each region (any other function can be reduced to this class through marginalization). The resulting optimization problem is a finite (or inifinite)-dimensional linear program that admits a closed-form greedy solution: visit the regions in decreasing order of likelihood ratio and assign $h = 1$ until the budget $\hat{s}(\boldsymbol{x}, y)$ is exhausted, then assign $h = 0$ for all remaining regions (with a fractional assignment at the boundary region to meet the equality constraint exactly).

Monte Carlo estimation and finite-sample corrections. In practice, the smoothed statistics ($\hat{s}$, p_j, etc.) are estimated via Monte Carlo sampling:

$$\hat{p}_j = \frac{1}{M}\sum_{m=1}^{M} \mathbf{1}\Big[s(\xi^{(m)}(\boldsymbol{x}), y) \le b_j\Big], \qquad \xi^{(1)}, \dots, \xi^{(M)} \overset{\text{i.i.d.}}{\sim} \xi. \tag{33}$$

Finite-sample corrections (e.g. via the DKW, Hoeffding, and Bernstein inequalities or Clopper–Pearson intervals) convert the empirical estimates into lower/upper confidence bounds with high probability $1-\delta$, ensuring validity of the certificates even with finite sampling. For example in Eq. 29 we replace the exact smoothed mean $\hat{s}(\boldsymbol{x}, y)$ (which is intractable to compute) with a high-confidence bound. In case that we are computing the certified upper bound, we replace $\hat{s}(\boldsymbol{x}, y)$ with an upper confidence bound $\overline{s}(\boldsymbol{x}, y)$, and vice versa for the certified lower bound. This results in a certified bound that holds with probability at least $1-\delta$ over the randomness of the Monte Carlo sampling.

This is the reason why randomized smoothing certificates are often more expensive to compute. To obtain a tight certificate, we need to use a large number of samples M, which means to run the model over the randomized augmentations of the same input M times. This directly multiplies the cost of computing the score function by M. The trade-off is between the tightness of the certificate and the computational cost: decreasing the sample rate M results in looser certificates – or looser certified bounds, which in robust CP it translates to larger prediction sets. An important part of our contribution in this thesis (specially in § 6, § 7, and § 8) is to propose methods that can achieve tight certificates with a smaller number of samples, and therefore reduce the computational cost of robust CP.

2.3.4 Summary of Ingredients

The methods developed in subsequent chapters combine the above building blocks as follows.

1. **Conformal prediction** converts any calibration set and score function into prediction sets with marginal coverage guarantee $1-\alpha$ under exchangeability (Theorem 2.1). Conformal risk control extends this guarantee to risk-controlling parameter selection.

2. **GNNs** are a powerful class of models for node classification, leveraging the graph structure to learn effective representations. Most of the models from this family are permutation equivariant by design, which is crucial for applying conformal prediction to node classification.

3. **Robust conformal prediction** aims to maintain the coverage guarantee under adversarial perturbations of the test input (or calibration data). Mainly this is done by replacing the exact conformity scores with certified bounds on their worst-case values within the threat ball (Theorem 2.5).

4. **Randomized smoothing** provides model-agnostic certified bounds for both continuous and sparse binary inputs. By adding random noise to the input and analyzing the smoothed score, we can derive tight certified bounds on the worst-case smoothed score within the threat ball, without any assumptions on the model or the score function.

3 Conformal Prediction for Graph Neural Networks

Abstract

Despite the widespread use of graph neural networks (GNNs) we lack methods to reliably quantify their uncertainty. We propose a conformal procedure to equip GNNs with prediction *sets* that come with distribution-free guarantees – the output set contains the true label with arbitrarily high probability. Our post-processing procedure can wrap around any (pretrained) GNN, and unlike existing methods, results in meaningful sets even when the model provides only the top class. The key idea is to diffuse the node-wise conformity scores to incorporate neighborhood information. By leveraging the network homophily we construct sets with comparable or better efficiency (average size) and significantly improved singleton hit ratio (correct sets of size one). In addition to an extensive empirical evaluation, we investigate the theoretical conditions under which smoothing provably improves efficiency.

3.1 Introduction

From health to traffic forecasting, graph neural networks (GNNs) have become a fundamental building block in a variety of applications. Even though their versatility places them in the spotlight among other machine learning topics, they seldom provide reliable uncertainty estimates. Since test accuracy is not necessarily a trustworthy indicator of performance, it is essential to explicitly quantify the model uncertainty for different inputs, especially in safety-critical domains. Naively considering the predicted distribution over labels (e.g. from the softmax) does not produce a good estimate of the true conditional probability $p(y \mid \boldsymbol{x})$ since models are often overconfident and uncalibrated Guo et al. (2017); Hein et al. (2019). Most uncertainty quantification methods are computationally expensive, and/or require modifications to the model architecture or at least retraining of the model Hüllermeier and Waegeman (2021); Abdar et al. (2020). Moreover, most techniques rely on the i.i.d. assumption, which is clearly violated for node classification due to the interdependence between nodes. Hence, methodological contributions in this direction are often incompatible with graph-based models such as GNNs Stadler et al. (2021).

Conformal prediction (CP) is a promising paradigm for constructing prediction sets (or intervals in the case of regression) with a statistically sound coverage guarantee – the output set covers the true label with any user-specified probability. CP is distribution free and it relies on exchangeability, i.e. the only assumption is that every permutation of the instances (in our case the nodes) is equally likely. In other words, we assume that the indexing of the random variables is immaterial. This makes CP a prime candidate for uncertainty quantification on graphs since exchangeability relaxes the i.i.d. assumption. In § 3.3 we discuss in detail the settings (e.g. transductive vs. inductive) under which this assumption is satisfied for semi-supervised node classification. More generally, we prove that semi-supervised learning with (subset-) permutation-equivariant models preserves exchangeability. Interestingly, even in cases where exchangeability may be violated, it is still possible to provide strong guarantees while incurring a coverage penalty that is proportional to the degree of distribution shift Barber et al. (2022).

Although full conformal prediction has a significant computational cost, *split conformal prediction* is fast, easy to implement, and model and data-distribution independent Vovk et al. (2005); Shafer and Vovk (2008). Since it uses the model as a black box, there is no need to retrain or modify it. Along with a provable coverage guarantee, the sets are interpretable and can be used to communicate with non-expert stakeholders,

making CP readily applicable to different domains like medicine Vazquez and Facelli (2022), electricity market forecasting Kath and Ziel (2021), and robotics Luo et al. (2023). Contrary to full conformal, split conformal prediction sacrifices statistical efficiency for computational efficiency, while there are extensions that sit in the middle of this tradeoff, e.g. cross-conformal prediction Vovk (2015), and CV+/Jackknife+ Barber (2020); Barber et al. (2019b). We focus on the split conformal setting, but our diffusion-based approach can be extended to CV+/Jackknife+.

An important ingredient of CP is the conformity score $s(\boldsymbol{x}, y)$ which quantifies the agreement between an observation $\boldsymbol{x}$ and a candidate label y. While the coverage guarantee holds for any scoring function s, the output sets are more efficient (i.e. smaller on average) the closer s is able to track the true conditional label distribution (see § 3.4 for a detailed discussion). Our key insight is that the network structure for homophilous graphs contains valuable information which we leverage to refine the node-wise conformity scores. Specifically, our main contributions are:

- A method called Diffusion Adaptive Prediction Sets (DAPS) to smooth node-wise conformity scores resulting in prediction sets with comparable or better efficiency and significantly improved singleton hit ratio.
- Theoretical insights into when smoothing is beneficial, and a rigorous discussion of graphs and exchangeability.
- The first thorough empirical evaluation of conformal prediction for transductive node classification.

In contrast to existing baselines, our method produces meaningful sets even without access to the class distribution. This considerably expands its applicability, e.g. to cloud-based models that only provide a prediction. DAPS is effective and simple – which we argue is its biggest strength.

3.2 Background

First, we review the concept of standard conformal prediction (as applied to e.g. image classifiers) without considering any additional structure like network homophily. We also cover the current state-of-the-art conformity scores.

Let $\pi(\boldsymbol{x}) \in \triangle^K$ be the distribution over $K = |\mathcal{Y}|$ class labels predicted by some classifier f (pre-)trained on $\mathcal{D}_{\text{train}}$. For example, $\pi(\boldsymbol{x}) = \sigma(f(\boldsymbol{x}))$ where σ is the softmax applied on the last layer of a neural network f, but any classifier that outputs a distribution is applicable. Given access to calibration data $\mathcal{D}_{\text{cal}} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ we can construct a prediction set $C(\boldsymbol{x}_{n+1}) \subseteq \mathcal{Y}$ for an unseen test example $\boldsymbol{x}_{n+1}$ with the following coverage guarantee $\mathbb{P}[y_{n+1} \in C(\boldsymbol{x}_{n+1})] \geq 1 - \alpha$, where α is the user-specified significance level. The only assumption is that $\mathcal{D}_{\text{cal}} \cup (\boldsymbol{x}_{n+1}, y_{n+1})$ is exchangeable.

Theorem 3.1 (Vovk et al. (2005))**.** *Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n+1}$ be exchangeable. For any score function $s : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ and any significance level $\alpha \in (0, 1)$, define quantile*

$$\hat{q} := \text{Quantile}\left(\frac{\lfloor (n-1)(\alpha) \rfloor}{n}; \{s(x_i, y_i)\}_{i=1}^{n}\right)$$

and prediction sets as $C_\alpha(\boldsymbol{x}_{n+1}) = \{y : s(\boldsymbol{x}_{n+1}, y) \geq \hat{q}\}$. We have[2]

$$1 - \alpha + \frac{1}{(n+1)} \geq \mathbb{P}[y_{n+1} \in C(\boldsymbol{x}_{n+1})] \geq 1 - \alpha \tag{34}$$

[2]The upper bound holds when there are no ties between the scores, but in practice, ties are broken by adding random noise.

The conformity score function $s(\boldsymbol{x}, y)$ quantifies the agreement between an observation $\boldsymbol{x}$ and a candidate label y[3].

Theorem 3.1 provides a *marginal* coverage guarantee that holds true on average for all $\boldsymbol{x}$. It has been shown that without strong unrealistic assumptions, coverage guarantees conditional on a given $\boldsymbol{x}$ are impossible Vovk (2012); Barber et al. (2019a). Additionally, with $l = \lfloor (n+1)\alpha \rfloor$, Vovk (2012) shows that the coverage follows a Beta distribution

$$\mathbb{P}\left[y_{n+1} \in C_\alpha(\boldsymbol{x}_{n+1}) \mid \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n\right] \sim \textbf{Beta}(n+1-l, l)$$

This means that if we resample the calibration set, the empirical coverage on the test set will be centered on $1-\alpha$. Two conclusions are directly implied: (i) the number of calibration samples has an effect on the concentration (and the variance) of the coverage probability (and other metrics), and (ii) the coverage is also upper-bounded, which in trivial cases where α is smaller than the model's accuracy, may lead to systematic miscoverage (for details see subsubsection 3.5.1).

Conformity scores. An obvious idea for the conformity score is $s(\boldsymbol{x}, y) := \pi(\boldsymbol{x})_y$ where $\pi(\boldsymbol{x})_y$ is the predicted probability for class y, which is known as threshold prediction sets (TPS) Sadinle et al. (2018). However, this scoring method has the tendency to undercover hard examples and overcover trivial ones Angelopoulos and Bates (2021). Hence, a popular alternative is the *adaptive* prediction sets (APS) Romano et al. (2020) method. Assuming we have access to an oracle, let $p(y \mid \boldsymbol{x})$ be the ground-truth conditional label distribution. We can form C_α by including classes one by one, from the most likely class to the least likely, until the cumulative probability becomes $> 1-\alpha$. This is the motivation behind APS. In place of the oracle, APS uses the estimated $\pi(\boldsymbol{x})$ defining $s(\boldsymbol{x}, y) := -\left(\rho(\boldsymbol{x}, y) + u \cdot \pi(\boldsymbol{x})_y\right)$ where $\rho(\boldsymbol{x}, y) := \sum_{c=1}^K \pi(\boldsymbol{x})_c \mathbf{1}\left[\pi(\boldsymbol{x})_c > \pi(\boldsymbol{x})_y\right]$ is the sum of all classes predicted as more likely than y, and $u \in [0,1]$ is a uniform random value that breaks potential ties between different scores Stutz et al. (2021)[4].

One drawback of APS is that it results in large sets. To overcome this, Angelopoulos et al. (2021) propose a regularization approach, called *regularized* adaptive prediction sets (RAPS), penalizing labels that are less likely, and thus encouraging smaller sets. Formally, let $o(\boldsymbol{x}, y) := |\{c \in \mathcal{Y} : \pi(\boldsymbol{x})_y \geq \pi(\boldsymbol{x})_c\}|$ be the rank of y, the proposed score is $s(\boldsymbol{x}, y) = -\left(\rho(\boldsymbol{x}, y) + u \cdot \pi(\boldsymbol{x})_y + \nu \max\left(o(\boldsymbol{x}, y) - k, 0\right)\right)$, where ν and k are hyperparameters. Intuitively, the regularization term penalizes classes that are at the bottom of the rank (after k) proportionally to ν, so to be selected for the predictive set, a lower quantile is needed.

3.3 Graphs and Exchangeability

Let $G = (\boldsymbol{X}, \boldsymbol{A})$ be a graph where $\boldsymbol{X}$ is the matrix of node features and $\boldsymbol{A}$ the adjacency matrix. Let $\mathcal{V}_l$ and $\mathcal{V}_u$ be disjoint sets of labeled and unlabeled nodes and $\mathcal{V} = \mathcal{V}_l \cup \mathcal{V}_u$. In all settings we split $\mathcal{V}_l = \mathcal{V}_d \cup \mathcal{V}_c$ into a disjoint development (training + validation) $\mathcal{V}_d$ and a calibration $\mathcal{V}_c$ set. We discuss three settings, focusing mostly on the first.

Transductive case. [5] Here the model has access to the entire graph during training, calibration, and testing. We assume an arbitrary fixed graph and that the union of

[3] Conformity scores can equivalently be defined as measuring the nonconformity (disagreement).

[4] This randomization helps achieve an exact $1-\alpha$ coverage.

[5] In the conformal prediction literature, full conformal prediction is sometimes referred to as transductive, while split conformal is referred to as inductive. This is orthogonal to the use of the terms transductive and inductive in semi-supervised node classification where they indicate which part of the graph is seen

calibration and unlabeled nodes $\mathcal{V}_c \cup \mathcal{V}_u$ is exchangeable. The set of development nodes $\mathcal{V}_d$ may have arbitrary dependencies. In our experiments, we sample all of the labeled nodes $\mathcal{V}_l$ uniformly at random so the exchangeability of $\mathcal{V}_c \cup \mathcal{V}_u$ is satisfied by construction since any node has an equal chance to land in $\mathcal{V}_c$ or $\mathcal{V}_u$. It is plausible that in a real-world application, the labeling budget is randomly allocated, but any other exchangeable sampling strategy is permitted. Importantly, while the classifier has access to the node features and the neighborhood structure for all nodes in the calibration set $\mathcal{V}_c$, their labels are *not* revealed during training. In other words, the classifier itself (and thus the score) cannot distinguish between calibration $\mathcal{V}_c$ and unlabeled $\mathcal{V}_u$ nodes.

Simultaneous inductive case. This setting is identical to the transductive case except the classifier is trained only on the subgraph induced by $\mathcal{V}_d$. The rest of the graph, including the calibration and the unlabeled test nodes, are simultaneously revealed after training and before calibration.

Inductive case. The classifier is again trained only on the subgraph induced by $\mathcal{V}_d$. We calibrate on the extended subgraph induced by $\mathcal{V}_d \cup \mathcal{V}_c$. The rest of the unlabeled test nodes may arrive either one at a time or in batches.

Exchangeability. We show that the conformity scores from a transductive (and simultaneous inductive) semi-supervised GNN are exchangeable. With $s(v, y \mid \boldsymbol{X}, \boldsymbol{A}) = s(v, y)$ denote the score for node v and a candidate label y, which may depend on all other nodes. We omit $\boldsymbol{X}$ and $\boldsymbol{A}$ for brevity.

Proposition 3.2. *Assume that $\mathcal{V}_c \cup \mathcal{V}_u$ is exchangeable. Let $\pi(G) = \boldsymbol{\Pi} \in \triangle^{|\mathcal{V}| \times K}$ be a matrix where row v is the label distribution for node v predicted by any permutation equivariant GNN classifier $\pi(\cdot)$ trained on the entire graph G and only using labels for nodes in $\mathcal{V}_d$. Then the scores $s(v, y) = \boldsymbol{\Pi}_{vy}$ where $\boldsymbol{\Pi}_{vy}$ is the predicted probability for node v and class y, are exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$.*

All omitted proofs are provided in § A.3. The gist here is that exchangeability is preserved since the classifier is permutation equivariant and does not distinguish between calibration $\mathcal{V}_c$ and unlabeled $\mathcal{V}_u$ nodes. Proposition 3.2 implies that the APS and RAPS scores are also exchangeable in this setting. More importantly, it implies that conformal prediction is applicable to node classification and that the coverage guarantee must hold. Our experimental evaluation confirms this since we can always obtain the desired coverage. Proposition 3.2, can also be trivially extended to the general transductive semi-supervised setting (e.g. on images) as long as permutation equivariance is satisfied.

Beyond exchangeability. In the inductive case exchangeability is violated whenever the conformity scores for calibration nodes are affected by a change in the graph. Specifically, as soon as a test node becomes part of the receptive field of any calibration node (e.g. its 2-hop neighborhood for a 2-layer GNN). Nonetheless, even in this case conformal prediction can still provide coverage guarantees, incurring only a penalty on the coverage that is proportional to the magnitude of the distribution shift as measured by the total variation distance (see Barber et al. (2022) for more details). We consider

during training. In this paper, we mostly focus on split conformal prediction and GNNs that are trained in a transductive manner.

this setting in subsubsection A.5.10. Clarkson (2022) uses the same approach to adapt conformal prediction for inductive node classification assuming exchangeability to be violated in both inductive and transductive settings. Although this assumption is correct for the inductive scenario, our Proposition 3.2 shows that exchangeability is not violated in the transductive case.

Sparsity. We are often interested in the sparsely-labeled setting where we have access to a relatively small set of labeled nodes for training, validation, and now calibration. Using an unrealistically large validation set, e.g. larger than the training set, is one pitfall that can skew the evaluation results of GNNs (Shchur et al., 2018). Moreover, if we happen to have access to a large labeled set it is probably more effective to use it for training than validation. These concerns equally apply to the calibration set. Thus, under label scarcity one reasonable strategy is to split the labeled nodes into training, validation and calibration sets of the *same size*. Since most graphs are sparse themselves, sparsity leads to a serious issue for the NAPS approach proposed by Clarkson (2022). Adapting the weighted variant of CP from Barber et al. (2022), NAPS assigns a weight of one to adjacent nodes, and a weight of zero otherwise. Now, due to the two sources of sparsity, many of the unlabeled test nodes have no calibration nodes in their immediate neighborhood. This means we cannot form any valid prediction sets for them (since all weights are zero), which happens for up to 79% of nodes as we show in subsubsection 3.5.1. This problem persist regardless of whether we are in the inductive or transductive setting. Our approach does not suffer from the same issue and we can always obtain valid predictions for all nodes.

Graph generative models and exchangeability. There is a rich body of work on exchangeability and graph generative models such as the Stochastic Block Model (SBM) Holland et al. (1983). There we can make a distinction between node-exchangeable models like the SBM that generate either dense or empty graphs with probability one Lloyd et al. (2012), and edge-exchangeable models Cai et al. (2016a) that can exhibit sparsity. Since most real world graphs are sparse one can conclude that edge-exchangeability is a more reasonable assumption. However, this literature is orthogonal to our work since it concerns *generative* models, while our focus is on the transductive node classification setting where we assume an arbitrary given graph. Here, our exchangeability assumption is w.r.t. the node labels, regardless of how the features vectors and the graph structure is generated. Since we sample the set of calibration nodes uniformly at random, exchangeability is satisfied by construction.

3.4 Properties of Conformal Scores

Efficiency. On the surface, conformal prediction seems to sidestep the need for direct uncertainty quantification where the goal is to provide calibrated probability estimates, making it a convenient alternative. However, CP is highly dependent on the choice of the scoring function, which in turn depends on how well the unknown conditional label distribution is approximated by the model. Even if we can obtain a good approximation, all scoring functions are not created equal. As shown by Romano et al. (2020), assuming an oracle model that returns the true $p(y \mid \boldsymbol{x})$, the scores produced by APS provide the *smallest possible* sets that satisfy the conditional coverage guarantee. We conjecture an interesting implication of this result that, to the best of our knowledge, has not been discussed before: we can use efficiency to compare models. Let S_α^f be the average set size at significance α using APS and probabilities estimated by some model f. Similarly,

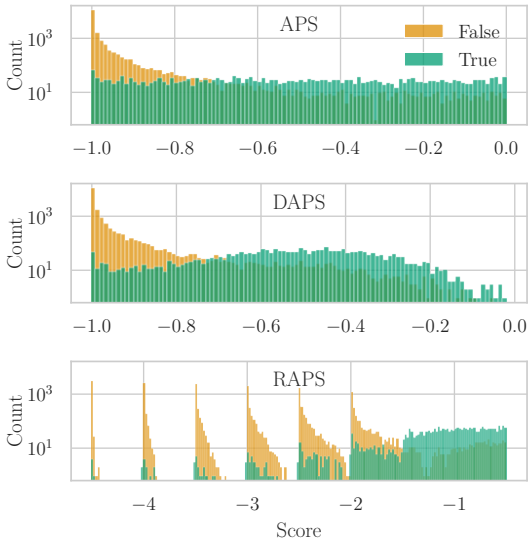

Figure 3.1: Histogram of conformity scores for ground-truth labels (True) and all other labels (False) on `CoraML` for `GCN`.

let S_α^g be the average set size for model g. If it holds $S_\alpha^f < S_\alpha^g$ for all α, i.e. CP on top of the model f always produces more efficient sets, then we can conclude that it is likely that f better approximates the oracle model. In § A.2 we discuss how to estimate the efficiency without having to explicitly compute it on the test set, and we use this insight to select the calibration hyperparameters.

Score distribution. To understand the effect of different scoring functions, we examine the distribution of conformity scores. In Fig. 3.1, we show the distribution for APS, RAPS and DAPS (introduced in subsubsection 3.4.1), where True denotes the scores for ground-truth labels in the calibration set and False denotes all other scores. We see that the penalty term in RAPS causes the distribution of low-ranked scores to concentrate on $K - k$ locations. This makes RAPS unstable since a small shift in the quantile threshold (e.g. due to sampling) can have a large effect on the outcome, causing RAPS to have a large variance, which we experimentally verify (see § 3.5). Moreover, Einbinder et al. (2022) show that for the oracle, the (conditional) distribution of true adaptive conformity scores is uniform. RAPS scores strongly deviate from this ideal case implying that it would be difficult to conclude how well the predicted sets capture the true $p(y \mid \boldsymbol{x})$. Diffused scores deviate from distribution mildly.

Evaluation metrics and the singleton hit ratio. In the conformal prediction literature, the most commonly used metric to compare different methods is efficiency (assuming valid coverage). We argue that another important metric is the singleton hit ratio defined as the fraction of examples with a *correct* prediction set of size one, i.e. singletons that contain the true label. For example, in a real-world application it is reasonable to automatically process all singleton predictions – simply predict the single class. However, a predicted set of size ≥ 1 might trigger inspection by a human expert who would have to decide how to handle the uncertain observation. Therefore, maximizing the singleton hit ratio would minimize the inspection effort in this example. As we will show in § 3.5, our approach significantly improves the singleton hit ratio even though it is not specifically designed to do so. It is also important to note that blindly optimizing for efficiency may not always be a good idea. Observations that are truly aleatorically

uncertain should indeed have larger sets.

3.4.1 Diffused Adaptive Prediction Sets

Our proposed DAPS exploits the graph structure by updating the node-wise conformal scores $s(v, y)$ based on neighborhood diffusion. We define the diffused score as

$$\hat{s}(v, y) = (1 - \lambda)s(v, y) + \frac{\lambda}{|\mathcal{N}_v|} \sum_{u \in \mathcal{N}_v} s(u, y) \tag{35}$$

where $\mathcal{N}_v$ is set of v's neighbors, and λ is a diffusion parameter. Practically, given the matrix of node-wise scores $\boldsymbol{H} \in \mathbb{R}^{|\mathcal{V}| \times K}$ the neighborhood diffused scores are $\hat{\boldsymbol{H}} = (1 - \lambda)\boldsymbol{H} + \lambda \boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{H}$ where $\boldsymbol{D}$ is the degree matrix.

We show that diffusion preserves exchangeability.

Proposition 3.3. *Let $\boldsymbol{H} \in \mathbb{R}^{|\mathcal{V}| \times K}$ be any matrix where row v is the conformal scores for all classes y for node v, and let $\boldsymbol{H}$ be exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$. Then the diffused scores $\hat{\boldsymbol{H}}$ are also exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$.*

3.4.2 Homophily and Theoretical Benefits of Diffusion

The predicted label distribution approximates the ground truth. To understand when diffusion is beneficial, we compare the approximation error before and after diffusion.

Theorem 3.4. *Let $\boldsymbol{\pi}_i$ be the model's approximation of the ground-truth conditional probability vector $\boldsymbol{p}_i$, and let the diffused distribution be $\hat{\boldsymbol{\pi}}_i = (1 - \lambda)\boldsymbol{\pi}_i + \frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \boldsymbol{\pi}_j$. Assume that the $G = (\boldsymbol{X}, \boldsymbol{A})$ is constructed such that $\boldsymbol{A}_{ij} = 1$ iff $||\boldsymbol{p}_i - \boldsymbol{p}_j|| \leq \Delta$ where $||\cdot||$ is the total variation norm. Diffusion improves the approximation error $\epsilon_i = ||\boldsymbol{\pi}_i - \boldsymbol{p}_i||$, i.e. $||\hat{\boldsymbol{\pi}}_i - \boldsymbol{p}_i|| < \epsilon_i$ if $\frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \epsilon_j + \Delta < \epsilon_i$.*

Assuming that the graph is constructed in a homophilous manner – here edges are formed only between nodes with ground-truth distributions closer than some Δ – Theorem 3.4 shows that diffusion helps whenever the average approximation error in the neighborhood plus a Δ penalty is smaller than the node's own error. Diffusion is beneficial since a better approximation of $\boldsymbol{p}_i$ leads to more efficient sets.[6]

Efficiency is affected by more than just the model accuracy. We can amplify or decrease the absolute probabilities without affecting the accuracy by e.g. rank-preserving transformations such as temperature scaling or uniform perturbations (see § A.4 for a discussion). Moreover, in § 3.5 we show that diffusion has a negligible (positive) impact on accuracy while significantly improving efficiency and singleton hit ratio. This shows that diffusion does not change the most likely label, but rather refines the distribution of labels.

Simulation. To illustrate the effect of diffusion we create synthetic data where the true label distribution is known. Then we simulate approximation errors by perturbing the nodes – each node is randomly perturbed with noise which either has a large

[6]The result in Theorem 3.4 is about diffusion in probability space, while Eq. 35 performs diffusion on the scores. We investigate both variants in subsubsection A.5.8. Moreover, it's easy (but notationally more cumbersome) to derive a similar result for the score diffusion.

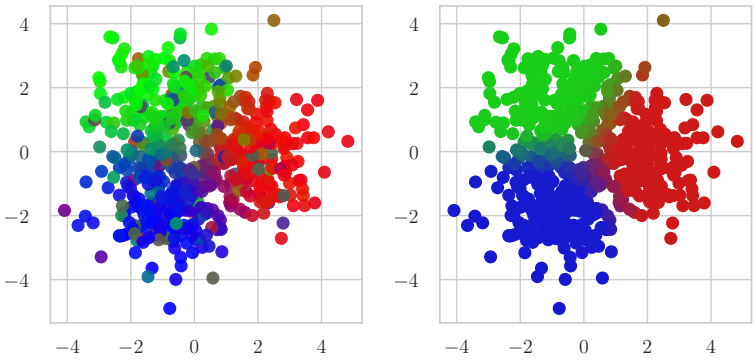

Figure 3.2: Diffusion (right) corrects the perturbed synthetic data (left). The RGB color shows the probability of each class.

magnitude with probability $p_s = 0.20$ or small magnitude with probability $1 - p_s$ (see § A.4 for more details). The graph is constructed for $\Delta = 0.4$. In Fig. 3.2 we see that the diffused probability vectors can correct the introduced perturbations. Intuitively, if the neighborhood of a node is mostly unperturbed, its probability vector can be reconstructed from its neighbors as long as their probability vectors are similar enough (i.e. we have homophily).

3.4.3 Generalizations of Neighborhood Diffusion

We generalize diffusion beyond the 1-hop neighborhood. For instance, we can incorporate the k-hop neighbors as $\hat{\mathbf{H}} = \lambda_0 \mathbf{H} + \sum_{i=1}^{k} \lambda_i (\boldsymbol{D}^{-1}\boldsymbol{A})^i \times \mathbf{H}$. We investigate the 2-hop variant with parameters λ_0 and λ_1. Inspired by label propagation (LP), we define another variant which we call *score propagation (SP)*, where each node propagates its scores to its neighbors. We define the iterative score update as $\hat{\boldsymbol{H}}^{(t)} = (1-\lambda)\hat{\boldsymbol{H}}^{(0)} + \lambda \hat{\boldsymbol{H}}^{(t-1)} \boldsymbol{D}^{-1}\boldsymbol{A}$, where $\hat{\boldsymbol{H}}^{(0)} = \boldsymbol{H}$ is the initial node-wise score, $\lambda \in (0, 1)$ can be interpreted as the teleport probability in the corresponding random walk, and $\boldsymbol{D}^{-1}\boldsymbol{A}$ is the degree-normalized adjacency matrix. Similar to LP, the close-form solution is $\hat{\boldsymbol{H}} = (1-\lambda)\left(\boldsymbol{I} - \lambda \boldsymbol{D}^{-1}\boldsymbol{A}\right)^{-1} \boldsymbol{H}$. In practice, we do not perform the matrix inverse and run $t = 10$ iterations which is enough for convergence. Note, since both of these transformations are permutation equivariant it is easy to see that they also preserve exchangeability like the 1-hop variant (see proof of Proposition 3.3). In § 3.5 we show that the 2-hop and SP variants improve over the 1-hop variant, however, for simplicity, in most experiments we focus on the latter. For code and computational complexity analysis see § A.1.

3.4.4 Hard Predictions

In some cases, the model outputs only the most likely label and there is no information on the predicted distribution of labels. For example, cloud-based models may provide such hard predictions on purpose for privacy protection since this makes membership inference attacks more difficult. In this case, $\pi(\boldsymbol{x})_y = 1$ for the predicted label, and 0 for all other labels. Therefore, the (empirical) distribution of scores computed by APS will degenerate with all scores concentrated on one of two locations, introducing many ties between the scores. While the coverage guarantee will still hold due to the built-in randomization with uniform noise, the resulting prediction sets will be less informative. Moreover, RAPS is not applicable at all since it penalizes scores based on rank, but here

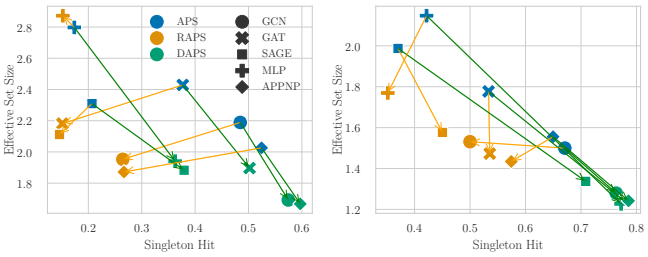

Figure 3.3: Comparing efficiency and singleton hit ratio for all models on `CoraML` (left) and `CoauthorPhysics` (right) datasets with adaptive coverage.

	RAPS		DAPS	
Dataset	Eff ↓	SH ↑	Eff ↓	SH ↑
`CoraML`	-0.24	-0.22	-0.50	0.09
`PubMed`	0.07	-0.13	0.02	-0.01
`CiteS`	0.08	-0.24	-0.40	0.10
`Co-CS`	-0.93	-0.09	-1.03	0.13
`Co-Phy`	0.03	-0.17	-0.22	0.09
`Amz-C`	-0.35	-0.16	-0.77	0.19
`Amz-P`	-0.02	-0.23	-0.48	0.16

Table 3.1: Performance relative to APS across all small datasets for `GCN` model. DAPS is best overall.

there is no rank information (all values are either 0 or 1). In contrast, DAPS will diffuse the scores based on the neighborhood and recover some of the missing information about the distribution of labels.

We provide some intuition on why diffusion works in this case. Bahri and Jiang (2021) have shown that even if we only have access to hard labels sampled from the true $p(y \mid \boldsymbol{x})$, the estimate $p_k(y \mid \boldsymbol{x})$ computed as the average label among the k nearest neighbors, approaches the true distribution at a minimax optimal rate for an appropriately chosen value of k. Assuming that the graph is constructed based on the k nearest neighbors in feature space, $p_k(y \mid \boldsymbol{x})$ coincides with the diffused $\hat{p}(y \mid \boldsymbol{x})$ for $\lambda = 1$. Thus, if all predictions are correct, $\hat{\pi}(y \mid \boldsymbol{x})$ also approaches the true distribution at the same optimal rate. Since the model is never perfectly accurate, there will be some estimation error, however, in practice we observe that diffusion indeed provides a significant performance boost (see § 3.5). We leave it for future work to theoretically characterize this setting in more detail.

3.5 Experimental Evaluation

We study the: (i) the impact of diffusion on efficiency and singleton hit ratio for semi-supervised node classification, (ii) the stability of all methods to random sampling and their sensitivity to hyperparameters, (iii) and the performance when we only have hard predictions. We compare our approach with the two strongest baselines APS and RAPS, which do not explicitly take the graph structure into account (although implicitly the graph is used to produce the probability vectors $\boldsymbol{\pi}_i$). For experimental evaluation, we put our main focus on the transductive case. We also provide some experiments for inductive and simultaneous inductive settings in subsubsection A.5.10. Moreover, in subsubsection A.5.2 we study the combination of the regularization from RAPS plus diffusion (although it's not recommended due to the instability of RAPS). In subsubsection A.5.4 we show that the margin score Wijegunawardana et al. (2020) also benefits from diffusion.

Models and datasets. We evaluate conformal prediction considering five different models: `GCN` Kipf and Welling (2017), `GAT` Veličković et al. (2018), `GraphSAGE` Hamilton et al. (2017), and `APPNP`Klicpera et al. (2019) as structure-aware models and `MLP` as a structure-independent model. We evaluate our approach on 10 datasets. The common citation graphs `CoraML` McCallum et al. (2004), `CiteSeer` Sen et al. (2008), `PubMed` Namata et al. (2012), `CoraFull`Bojchevski and Günnemann (2018), `Coauthor Physics`

and Coauthor CS Shchur et al. (2018). The co-purchase graphs Amazon Photos and Amazon Computers McAuley et al. (2015); Shchur et al. (2018). And two large graphs, OGBN Arxiv Wang et al. (2020) and OGBN Products Bhatia et al. (2016). CoraML* and CoraFull* are variants considering the largest connected component. Datasets statistics are in § A.7.

Evaluation procedure. We randomly split the nodes into train/validation/calibration/test sets. Since GNNs are sensitive to splits, especially in the sparsely labeled setting Shchur et al. (2018), we train 10 different models with different train/validation splits and report the average. We randomly select 20 nodes per class for training/validation. As described in § A.2, we split the calibration set into two sets, one for tuning parameters like λ, and one for actual calibration. To reflect a realistic scenario, the size of the calibration (and tuning) set is the same as as the training set size. Since APS is parameter-free, for fairness we increase its calibration set such that it uses the same total number of labels. We report average results over 100 calibration/test splits. The calibration/test labels are never used for model training. See § A.7 for details on the labeling budget.

3.5.1 Comparing Conformal Prediction Sets for GNNs

Efficiency and singleton hit ratio. As discussed in § 3.4, efficiency (average set size) and the singleton hit ratio are two important metrics. We consider two settings: a fixed coverage with $\alpha = 0.08$ and an adaptive coverage, related to the actual accuracy of the model, which we discuss in subsubsection A.5.3. Fig. 3.3 shows that DAPS slightly improves efficiency while significantly increasing the singleton hit ratio compared to APS and RAPS. We also study the performance across different coverage guarantees. Fig. 3.4 shows the result for the large OGBN Products dataset, again seeing that DAPS performs best overall. We provide a comprehensive report on all datasets and models in subsubsection A.5.9 with similar conclusions. In subsubsection A.5.6 we also compare empty, singleton, and multi-sets.

On Fig. 3.4 (right) we observe a mild increase in the singleton hit ratio for values of $1 - \alpha$ close to the model's accuracy. Since the coverage is distributed as a Beta distribution, there is both an upper and a lower bound on the coverage meaning that in these conditions CP is forced to discard potential singleton sets to satisfy the upper bound. As the coverage gets closer to the accuracy, there is a potential for improvement in the singleton hit ratio. The inflection point around 78.4% (model's accuracy with diffusion) is followed by a trade-off between coverage and the singleton hit ratio.

Generalizations of diffusion. So far we focused on 1-hop diffusion since it is simple and computationally inexpensive, making it practical. We also evaluate the 2-hop and score propagation (SP) variants introduced in subsubsection 3.4.3. In Fig. 3.5, for a GCN model on CoraML, we see that both variants provide further improvements. We leave it as a future work to study what is the optimal form of diffusion.

Efficiency and accuracy. Model accuracy plays a significant role in conformal prediction. One might wonder whether the improvements from DAPS stem primarily from improved accuracy. Fig. 3.6 (right) shows this is not the case. In most cases DAPS does not increase the accuracy (when predicting $\arg\max_y \hat{s}(v, y)$), while significantly increasing efficiency and singleton hit ratio. We further investigate this phenomenon in

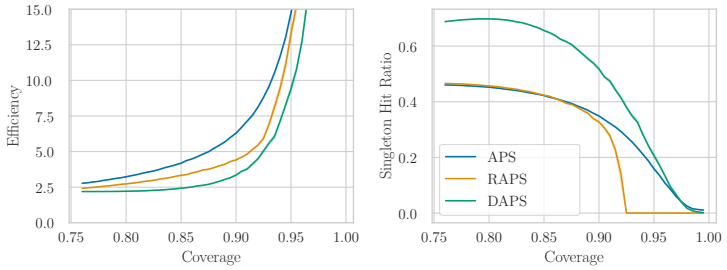

Figure 3.4: DAPS scales to large datasets such as `OGBN Products` and provides a strong improvement in both efficiency (left) and singleton hit ratio (right) for any coverage.

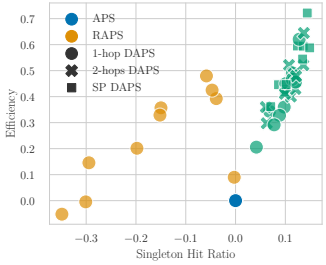

Figure 3.5: RAPS vs. DAPS variants relative to the APS baseline for `CoraML/GCN`.

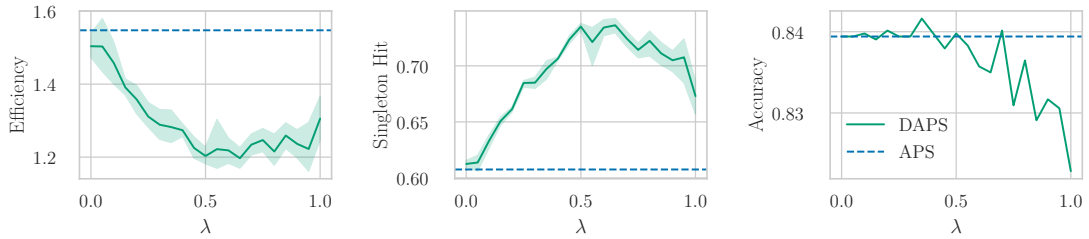

Figure 3.6: DAPS for different values of λ for a `GCN` model trained on `CiteSeer`. From left to right: the effect of diffusion on (i) efficiency (ii) singleton hit ratio (iii) accuracy. The change in accuracy is negligible while significantly improving the other two metrics.

subsubsection A.5.5, and find a large subset of λ values that lead to improvement in CP metrics, with the optimal value of around $\lambda = 0.5$ for most datasets and models.

Stability. GNNs are known to be sensitive to data splits, so we study the stability of all methods across varying conditions. We explore two settings: (i) we tune the parameters on a single given split and we evaluate the same parameters on all other splits, (ii) we tune the parameters for a single α and we evaluate on all other values of α. In Fig. 3.7 (left) we see that the variance of RAPS across both metrics is significantly larger compared to APS and DAPS (as visually shown by the rectangles). In Fig. 3.7 (right) we see the relative enhancement over APS. Each circle corresponds to a different α and its size shows the magnitude. Unlike RAPS, DAPS provides a consistent enhancement for all values.

Comparison with NAPS. As discussed in § 3.3 due to sparsity many test nodes are not adjacent to any calibration nodes. As a result, NAPS fails to return a prediction set for them. Table 3.2 shows that on `CoraML` with `GCN` under the default evaluation setting with $|\mathcal{V}_c| = 5.2\%$, NAPS is non-applicable and unable to make predictions for 79% of test nodes (see subsubsection A.5.11 for details). Even if we make the size of the calibration set unrealistically large, NAPS still fails for many test nodes. DAPS is always applicable regardless of the size of $\mathcal{V}_c$, and returns more efficient sets.

Hard predictions. We compare only with APS since, as we discussed in § 3.4, RAPS is not applicable when only hard predictions are given. If we naively use APS the coverage guarantee cannot be satisfied since we will still get ties despite the built-in

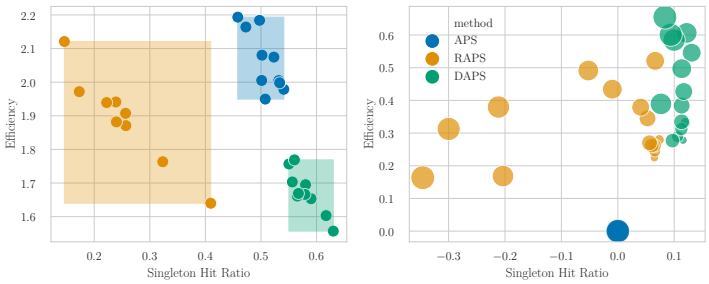

Figure 3.7: Stability for a `GCN` model on `CoraML` across different initial splits (left, absolute difference) and different coverage guarantees (right, enhancement relative to APS).

$\lvert\mathcal{V}_c\rvert$	NAPS		DAPS
	Eff ↓	N/A ↓	Eff ↓
5.2%	2.24	79%	1.81
10%	2.34	67%	1.82
20%	2.44	47.9%	1.81
50%	2.58	19.5%	1.80

Table 3.2: Comparison between DAPS and NAPS ($k = 1$) for different calibration set sizes in a transductive setting. N/A means "not applicable". Note that DAPS is always applicable to all test nodes.

randomization (see subsubsection A.5.1 for details). To make APS applicable we add a small constant ϵ to all classes and renormalize (increasing 0 to ϵ and decreasing 1 to $1 - |C|\epsilon$). For comparability, we do the same for DAPS even though it does not need it.

In Fig. 3.8 we see that APS catastrophically fails to provide useful sets. In contrast, DAPS still works reasonably well, and its performance is close to the soft baseline that has access to the full distribution. Moreover, we see that for all methods the empirical coverage matches the guaranteed coverage, which is a good sanity check against theoretical and implementation bugs. The variance is also on the expected order of magnitude (recall that the coverage is distributed as **Beta**$(n + 1 - l, l)$).

Limitations. Diffusion relies on homophily. In subsubsection A.5.9 we study graphs with lower homophily such as `OGBN Arxiv`, where as expected diffusion does not provide a significant boost, and has different trade-offs than RAPS. Moreover, during tuning we can always select $\lambda = 0$ to disable diffusion. More importantly, a general limitation of conformal prediction is that the guarantees only hold marginally (over all test nodes). Recall, that conditional coverage is impossible without additional strong assumption Barber et al. (2019a). Thus, we have to be careful when interpreting the

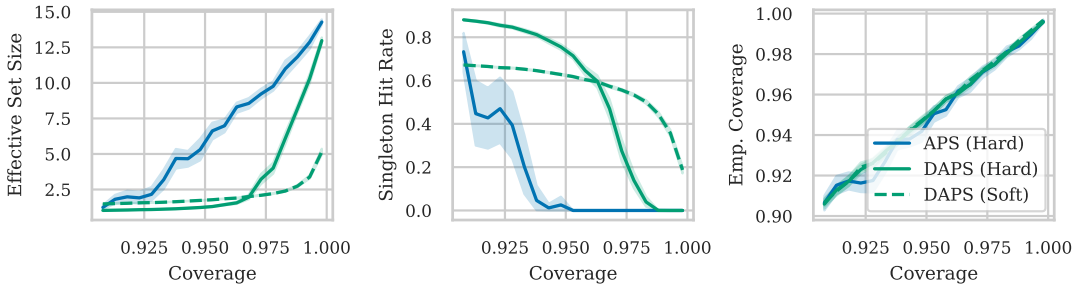

Figure 3.8: Comparison of APS and DAPS when using hard predictions on the `Coauthor-CS` dataset for a `GCN` model. We evaluate three metrics: efficiency (left), singleton hit ratio (middle), and empirical coverage (right). Dashed lines are recalls from CP approaches with access to the predicted softmax probabilities while solid line correspond to the same approach applied over the hard (one-hot) predictions.

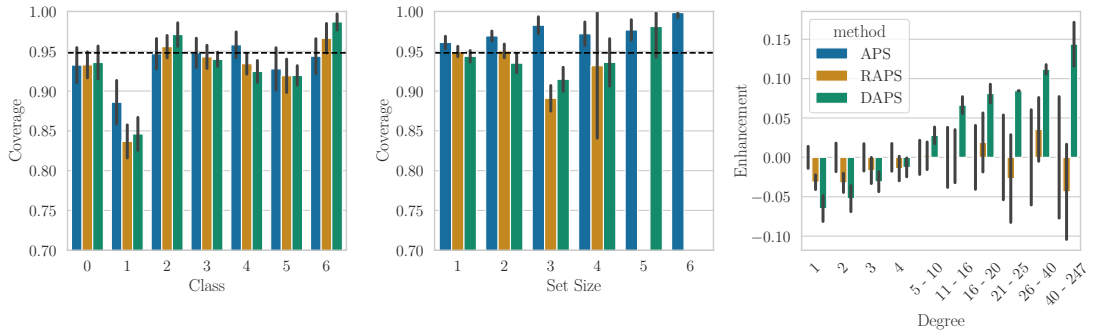

Figure 3.9: Empirical coverage conditional on the class label (left), prediction set size (middle), and node degree (right). On the middle plot the support for each set size is different per method. On the right plot the lines without any bar show the s.d. of APS from its mean.

results. Nonetheless, in practice we observe good empirical coverage for different subsets of nodes. In Fig. 3.9 we investigate coverage conditional on the class label, the set size and the node degree and see that diffusion again provides a strong benefit. In subsubsection A.5.7 we investigate empirical conditional coverage and find that APS and DAPS are comparable. Finally, coverage is also upper-bounded (see Theorem 3.1).

3.6 Related Work

Conformal prediction. First introduced by Vovk et al. (2005), CP provides distribution-free guarantees assuming only exchangeability Lei and Wasserman (2014); Shafer and Vovk (2008).

Most works provide guarantees on the marginal coverage of the true label, however, CP can be generalized to any user-defined risk function Angelopoulos et al. (2022). Improving efficiency is often the goal. APS, RAPS and our DAPS do not change the model, while Stutz et al. (2021) improve efficiency by simulating calibration during training. Einbinder et al. (2022) encourage uniformity via a loss to improve conditional coverage. Fisch et al. (2022) add a constraint on the false positive rate. There is also significant effort in generalizing CP beyond exchangeability. Gendler et al. (2022) address the adversarial setting. Tibshirani et al. (2019a) define a weighted CP to handle covariate shift. Finally, Barber et al. (2022) propose a general framework where their guarantee has a penalty term proportional to the degree of distribution shift.

Uncertainty quantification on graphs. There are few studies on uncertainty quantification for graph-based models such as GNNs Abdar et al. (2020). One challenge is the interdependence between the nodes which prevents us from e.g. directly applying methods designed for i.i.d. data. Stadler et al. (2021) explicitly model epistemic and aleatoric uncertainty by propagating node-wise estimates along the graph. In § A.6 we show that their approach is orthogonal and can be combined with our CP guarantees. They define three axioms for uncertainty quantification with structural dependency. DAPS aligns with the third, indicating that a node's aleatoric uncertainty should increase when connected to conflicting nodes or nodes with higher aleatoric uncertainty. Cai et al. (2016b) study calibration and temperature scaling, and Hsu et al. (2022) study edge-wise calibration. A few works study out-of-distribution detection Liu et al. (2023); Huang et al. (2022); Bazhenov et al. (2022).

Conformal prediction on graphs. Wijegunawardana et al. (2020) is the first to apply CP on graphs. They propose a margin-based score, which unlike DAPS, does not explicitly account for the graph structure. In subsubsection A.5.9 we show that their score also benefits from diffusion. Nonetheless, we argue that using APS as the base score is more suitable, since similar to TPS, the margin score may undercover hard examples. Recently, Clarkson (2022) introduces NAPS for the inductive case using the beyond-exchangeability technique from Barber et al. (2022), dismissing the transductive case as unsuitable. In § 3.3, we highlighted the major limitations of NAPS (see also subsubsection A.5.11 for a longer discussion). Alongside our main focus on the transductive scenario, we provide additional experiments on the inductive setting in subsubsection A.5.10. Finally, Kang et al. (2022) derives a variant of Jackknife+ for GCN. Different from existing works, in our score we explicitly leverage homophily while still providing a valid coverage guarantee.

3.7 Conclusion

We propose conformal prediction sets that explicitly account for the graph structure. The key insight is that diffusing the conformity scores along the graph leads to improved uncertainty quantification in presence of homophily. We discuss exchangeability for graphs and GNNs, and the theoretical conditions under which diffusion is beneficial. Our method, DAPS, performs on par or better than the baselines in efficiency and significantly better w.r.t. singleton hit ratio.

4 Conformal Inductive Graph Neural Networks

Abstract

Conformal prediction (CP) transforms any model's output into prediction sets guaranteed to include (cover) the true label. CP requires exchangeability, a relaxation of the i.i.d. assumption, to obtain a valid distribution-free coverage guarantee. This makes it directly applicable to transductive node-classification. However, conventional CP cannot be applied in inductive settings due to the implicit shift in the (calibration) scores caused by message passing with the new nodes. We fix this issue for both cases of node and edge-exchangeable graphs, recovering the standard coverage guarantee without sacrificing statistical efficiency. We further prove that the guarantee holds independently of the prediction time, e.g. upon arrival of a new node/edge or at any subsequent moment.

4.1 Introduction

Graph Neural Networks (GNNs) are used in many applications however without a reliable estimation of their output's uncertainty. We can not rely solely on the predicted distribution of labels $\pi(y \mid \boldsymbol{x})$ (e.g. from the softmax) as it is often uncalibrated. Therefore, it is crucial to find confidence estimates aligned with the true $p(y \mid \boldsymbol{x})$. There is a rich literature on uncertainty quantification methods many of which require retraining or modifications to the model architecture. Among them, almost none come with a guarantee, and many rely on the i.i.d. assumption. Given the interdependency structure (adjacency) we cannot easily adopt these methods for node classification. As a result, uncertainty quantification methods for GNNs are very limited (Stadler et al., 2021).

Conformal prediction (CP) is an alternative approach that uses the model as a black box and returns prediction *sets* guaranteed to cover the true label without any assumptions on the model's architecture or the data generating process. This guarantee is probabilistic and works for any user-specified probability $1 - \alpha$. To apply CP, we need a conformity score function $s : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ that quantifies the agreement between the input and the candidate label. Additionally, we need a held-out *calibration* set. The only assumption for a valid coverage guarantee is exchangeability between the calibration set and the test set. This makes CP applicable to non-i.i.d settings like transductive node classification – Zargarbashi et al. (2023a) and Huang et al. (2023a) showed that with a permutation-equivariant model (like almost all GNNs) CP obtains a valid coverage guarantee.

Given the dynamic nature of real-world graphs which evolve over time, *inductive* node classification is more reflective of actual scenarios than transductive. In the inductive setting, we are given an initial graph (used for training and calibration) which is progressively expanded by set of nodes and edges introduced to the graph over time. Importantly test nodes are not present at the training and calibration stage. For GNNs, updates in the graph cause an implicit shift in the embeddings of existing nodes and consequently the softmax outputs and conformity scores. Therefore, in the inductive setting, even assuming node/edge exchangeability (see § B.2 for a formal definition), the coverage guarantee is no longer valid. Intuitively, calibration scores are no longer exchangeable with test scores, as soon as new nodes or edges are introduced (see Fig. 4.1 right).

To address this issue, Clarkson (2023) adapt the *beyond exchangeability* approach applying weighted CP for inductive node-classification aiming to recover the guarantee up to a bounded (but unknown) error (Barber et al., 2022). Unfortunately, this approach has an extremely limited applicability when applied with a realistic (sparse) calibration

set. In sparse networks and limited labels, the method fails to predict *any* prediction set for a large number of nodes; and for the rest, the statistical efficiency is significantly low (see § B.3 for details).

We show that for a node or edge exchangeable graph sequence (without distribution shift), we can adapt CP w.r.t. the implicit shift in score space, recovering the desired coverage guarantee. Node-exchangeability assumes that the joint distribution of nodes is invariant to any permutation. In the inductive setting, this means that any node is equally likely to appear at any timestep. We show that under node-exchangeability, the effect of new coming nodes is symmetric to the calibration set and the existing nodes. Hence, computing the conformity scores conditional to the subgraph at any timestep recovers the guarantee. Previous works under the transductive setting (Zargarbashi et al., 2023a) and (Huang et al., 2023a) also assume node-exchangeability.

While real-world graphs are sparse, all node-exchangeable generative processes produce either empty or dense graphs (see § 4.3). Therefore, we extend our study to the edge-exchangeable setting which can generate sparse graphs. In subsubsection 4.4.2 we show that w.r.t. scores, any edge-exchangeable sequence is equivalent to a weighted node-exchangeable sequence. Therefore, through weighted CP (weighted quantile lemma from Tibshirani et al. (2019b)) we recover the $1-\alpha$ coverage. Importantly, the improvement of our method is orthogonal to whether there is a distribution shift over time. Therefore, to address shift, we can still apply the weighting scheme from Barber et al. (2022) on top.

We focus on inductive node-classification. We define node-exchangeable (NodeEx) CP, and we show that (i) for node-exchangeable sequences, NodeEx CP obtains $1-\alpha$ guarantee by calibrating over shifted conformal scores conditional to subgraphs in each timestep (Fig. 4.1 left), (ii) the $1-\alpha$ guarantee is achievable via weighted CP on edge-exchangeable sequences, and (iii) The guarantee holds independent of the prediction time – the prediction set for a node at any timestep after its appearance benefits from the same coverage guarantee (e.g. prediction upon arrival or all nodes at once both result in the same coverage). We justify our approach with both theoretical and empirical results.

4.2 Background: Conformal Prediction and Transductive GNNs

Here we recall the general guarantee of CP (e.g. for image dataset). Through this paper, we assume a continuous score function without ties. This can apply to any function via adding noise. With weights $w_i \in [0,1]$, and the sorting permutation $\tau^\star$, we define the weighted quantile $\mathbb{Q}(\cdot;\cdot)\cdot$ as

$$\mathbb{Q}\left(\alpha;\{s_i\}_{i=1}^N\right)\{w_i\}_{i=1}^N = \inf\left\{s_{\tau^\star(i)} : \frac{\sum_{j=1}^i w_{\tau^\star(j)}}{\sum_{j=1}^N w_j + 1} \geq \alpha\right\} \tag{36}$$

Theorem 4.1 (Vovk et al. (2005))**.** *Let* $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$*, and* $(\boldsymbol{x}_{n+1}, y_{n+1})$ *be exchangeable. With any continuous function* $s : \mathcal{X}\times\mathcal{Y} \mapsto \mathbb{R}$ *measuring the agreement between* $\boldsymbol{x}$*, and* y*, and user-specified significance level* $\alpha \in (0,1)$*, with prediction sets defined as* $C_\alpha(\boldsymbol{x}_{n+1}) = \{y : s(\boldsymbol{x}_{n+1}, y) \geq \hat{q}\}$ *where* $\hat{q} := \mathbb{Q}\left(\tilde{\alpha}; \{s(x_i, y_i)\}_{i=1}^n\right)$ 1*. We have* $\mathbb{P}\left[y_{n+1} \in C(\boldsymbol{x}_{n+1})\right] \geq 1-\alpha$*.*

The marginal coverage probability is upper-bounded by $1-\alpha+1/(n+1)$. With infinite samples, the coverage is distributed as a Beta$(n+1-l, l)$ with $l = \lfloor (n+1)\alpha \rfloor$ (Vovk, 2012), while with a fixed population $|\mathcal{D}| = n+m$, and an exchangeable calibration set $|\mathcal{D}_{\text{cal}}| = n$, the probability of coverage $\text{Cov}(\mathcal{D}\setminus\mathcal{D}_{\text{cal}}) := (1/m)\mathbf{1}[y_i \in C(\boldsymbol{x}_i)]_{i\in\mathcal{D}-\mathcal{D}_{\text{cal}}}$ derives from a collection of hyper-geometric distributions (Huang et al., 2023a).

Conformity scores. The guarantee from Theorem 4.1 holds regardless of how

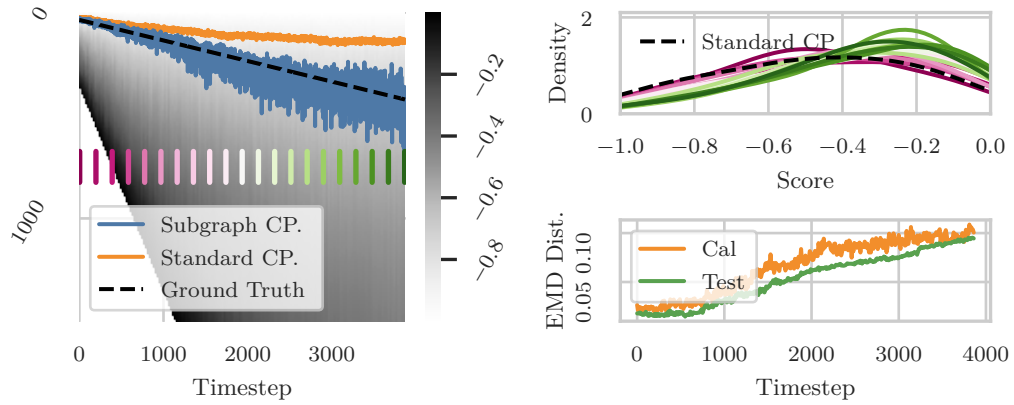

Figure 4.1: [Left] Each vertical line on the heatmap shows sorted true test scores at each timestep. The dashed line shows the true (unknown) α-quantile and the quantile from each approach is also shown alongside. NodeEx CP (ours) closely tracks the true quantile, while naive CP deviates over time. [Upper right] Distributions from selected timesteps marked by the same color on the heatmap. The distribution shift is observable over time with new nodes appearing. [Lower right] The earth mover distance (EMD) between naive CP calibration scores and shifted true scores, denoted as "Test"; and EMD between naive and NodeEx CP scores, denoted as "Cal". Details in subsubsection B.4.5.

the conformity score function $s : \mathcal{X} \times \mathcal{Y} \to R$ is defined. A wise choice of $s(\cdot, \cdot)$ is reflected in other metrics like the prediction set size. We use *adaptive* prediction sets (APS) with a score function defined as $s(\boldsymbol{x}, y) := -\big(\rho(\boldsymbol{x}, y) + u \cdot \pi(\boldsymbol{x})_y\big)$. Here $\rho(\boldsymbol{x}, y) := \sum_{c=1}^{K} \pi(\boldsymbol{x})_c \mathbb{1}\left[\pi(\boldsymbol{x})_c > \pi(\boldsymbol{x})_y\right]$ 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 (Stutz et al., 2021). Our method is independent of the choice of score function and we show its applicability on other (network-agnostic and network-aware) scores in subsubsection B.4.3.

CP beyond exchangeability. For cases where the calibration and test points are not exchangeable, there is a coverage gap Δ_α, i.e. $\mathbb{P}\left[y_{n+1} \in C_\alpha(\boldsymbol{x}_{n+1})\right] \geq 1 - \alpha - \Delta_\alpha$. Barber et al. (2022) shows that Δ_α has an upper bound that depends on how far the data deviates from exchangeability. This upper bound can be further controlled via weighted conformal prediction. For related works see § B.6.

4.2.1 Conformal Prediction for Transductive Node-Classification

Consider a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{A}, \boldsymbol{y})$ with $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ as node-features matrix, $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ as adjacency matrix, and $\boldsymbol{y} \in \mathbb{R}^n$ as labels. $\mathcal{V} = \mathcal{V}_{\text{tr}} \cup \mathcal{V}_{\text{cal}} \cup \mathcal{V}_u$ is the set of vertices (training, calibration, and test). Let f be a black-box permutation-equivariant model (e.g. a GNN) trained on labels of $\mathcal{V}_{\text{tr}}$, and s be a continuous score function with access to the outputs of f. The score function s may or may not use information about the graph structure (see subsubsection B.4.3). In the transductive setup the graph $\mathcal{G}$ is fixed, and training and calibration is performed with all observed nodes (including features and structure). The labels of $\mathcal{V}_{\text{tr}}$ and $\mathcal{V}_{\text{cal}}$ are used respectively for training and calibration, while all other labels remain unseen. Here when $\mathcal{V}_{\text{cal}}$ and $\mathcal{V}_u$ are exchangeable, standard CP can be applied (Zargarbashi et al., 2023a; Huang et al., 2023a). For a set with a fixed number of nodes $\mathcal{V}'$, we define the discrete coverage as $\text{Cov}(\mathcal{V}') = \frac{1}{|\mathcal{V}'|} \sum_{v_j \in \mathcal{V}'} \mathbf{1}\left[y_j \in \hat{C}(v_j)\right]$. The following theorem shows that under transductive calibration, CP yields valid prediction

sets.

Theorem 4.2 (Rephrasing of Theorem 3 by Huang et al. (2023a))**.** *For fixed graph $\mathcal{G}$, and a permutation-equivariant score function $s(\cdot,\cdot)$, with $\mathcal{V}_{\text{cal}} = \{(v_i, y_i)\}_{i=1}^{N}$ exchangeably sampled from $\mathcal{V} \setminus \mathcal{V}_{\text{tr}}$, and prediction sets $\hat{C}(v) = \left\{y : s(v,y) \geq \mathbb{Q}\left(\alpha; \{s(v_i, y_i)\}_{i=1}^{N}\right) 1\right\}$ we have*

$$\mathbb{P}\left[\text{Cov}(\mathcal{V}_u) \leq t\right] = 1 - \Phi_{HG}(i_\alpha - 1; M + N, N, \lceil Mt \rceil + i_\alpha) \tag{37}$$

where $i_\alpha = \lceil (N+1)(1-\alpha) \rceil$ is the unweighted α-quantile index of the calibration scores, and $\Phi_{HG}(\cdot; N, n, K)$ is the c.d.f. of a hyper-geometric distribution with parameters N (population), n number of samples, and K number of successful samples among the population.

The issue with inductive node-classification. The above approach does not directly translate to the inductive setting. As soon as the graph is updated, the implicit shift in the nodes embeddings results in a distribution shift w.r.t. the calibration scores that are computed before the update. This breaks the exchangeability as shown by Zargarbashi et al. (2023a) and in Fig. 4.1. To address this issue, Clarkson (2023) adopts weighted CP (Barber et al., 2022) with neighborhood-dependent weights (NAPS), limiting the calibration nodes to those inside an immediate neighborhood of a test node. Applying NAPS on sparse graphs or with small (realistic) calibration sets, leaves a significant proportion of test nodes with an "empty" calibration set. Hence, its applicability is limited to very special cases (see § B.3 for details). Moreover, NAPS does not quantify the coverage gap Δ_α. In contrast, when assuming either node or edge-exchangeability, the gap for our approach is zero regardless of the weights.

4.3 Inductive GNNs under Node and Edge Exchangeability

In the inductive scenario, the graph changes after training and calibration. Therefore, the model f is trained, and CP is calibrated only on a subgraph before the changes. We track these updates in a sequence of graphs $\mathcal{G}_1, \mathcal{G}_2, \ldots$ where $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ is a graph with a finite set of vertices and edges. We focus on progressive updates meaning $\forall t : \mathcal{V}_t \subseteq \mathcal{V}_{t+1}$ and $\mathcal{E}_t \subseteq \mathcal{E}_{t+1}$. At timestep t in a node-inductive sequence, the update adds a node v_t with all its connections to existing vertices. In an edge-inductive sequence, updates add an edge, which may or may not bring unseen nodes with it. W.l.o.g update sets are singular. Node-inductive and edge-inductive sequences are formally defined in § B.2.

We call a node-inductive sequence to be *node-exchangeable* if the generative process is invariant to the order of nodes, meaning that any permutation of nodes has the same probability. Analogously, a sequence is *edge-exchangeable* if all permutations of edges are equiprobable. For both cases, the problem is node-classification. Since split CP is independent of model training w.l.o.g. we assume that we train on an initial subgraph $\mathcal{G}_0$. We do not need to assume that $\mathcal{G}_0$ is sampled exchangeably. Formal definitions of node-exchangeability and edge-exchangeability are provided in § B.2.

For transductive GNNs, both Zargarbashi et al. (2023a) and Huang et al. (2023a) assume that the calibration set is sampled node-exchangeably w.r.t. the test set, however the entire graph is fixed and given. In contract, we assume that the sequence of graphs $\mathcal{G}_1, \mathcal{G}_2, \ldots$ itself, including calibration and test nodes, is either node- or edge-exchangeable. Transductive setting can be seen as a special case.

Sparsity. Any node-exchangeable graph is equivalent to mixture of *graphons* (Aldous, 1981; Cai et al., 2016a,b). A graphon (or graph limit) is a symmetric measurable function $W : [0,1]^2 \mapsto [0,1]$ and the graph is sampled by drawing $u_i \sim \text{Uniform}[0,1]$ for each

vertex v_i, and an edge between v_i, and v_j with probability $W(u_i, u_j)$. We know that graphs sampled from a graphon are almost surely either empty or dense (Orbanz and Roy, 2014; Cai et al., 2016b) – the number of edges grows quandratically w.r.t. the number of vertices. However, sparse graphs (with edges sub-quadratical in the number of vertices) are more representative to real-world networks. Therefore, we also consider edge-exchangeable graph sequences (Cai et al., 2016b) that can achieve sparsity.

4.4 Conformal Prediction for Exchangeable Graph Sequences

The first N nodes of the sequence (and their labels) are taken as the calibration set, leaving the rest of the sequence to be evaluated.[7] For easier analysis, we record the coverage of each node at each timestep. Let T_0 be the time when the last calibration node arrived. Up to timestep T, let matrix $\boldsymbol{C} \in \{0,1\}^{\mathcal{V}_T \times T}$ with $\boldsymbol{C}[i,t]$ indicate whether the prediction set for test node v_i at timestep t covers the true label y_i. The time index in $\boldsymbol{C}$ is relative to T_0 and w.l.o.g. we index nodes upon their appearance. Hence, in each column $\boldsymbol{C}[\cdot, t]$, the first $|\mathcal{V}_t|$ elements are in $\{0,1\}$ and the rest are N/A. Formally:

$$\boldsymbol{C}[i,t] = \begin{cases} \mathbf{1}\left[y_i \in \left\{y : s(v_i, y) \geq \mathbb{Q}\left(\alpha; \{s(v_i, y_i | \mathcal{G}_t)\}_{v_i \in \mathcal{V}_{\text{cal}}}\right) 1\right\}\right] & i \geq t \\ \text{N/A} & \text{o.w.} \end{cases} \tag{38}$$

Here we assume that one can evaluate a node at any timestep after its appearance. We define $\mathcal{V}^{(t)}_{\text{eval}} \subseteq \mathcal{V}_t$ as the set of nodes evaluated at timestep t. For an edge-exchangeable sequence, a node v is considered as active (existing) upon the arrival of the first edge connected to it. After T timesteps, we call the (recorded) sub-partition $\cup_{i=1}^{T} \mathcal{V}^{(i)}_{\text{eval}}$ as an evaluation mask $\mathbb{I}_T$. Any node appears at most once in $\mathbb{I}_T$ (we do not re-evaluate a node). We define the $\text{Cov}(\mathcal{V}^{(t)}_{\text{eval}}) = (1/|\mathcal{V}^{(t)}_{\text{eval}}|) \sum_{v_i \in \mathcal{V}^{(t)}_{\text{eval}}} \boldsymbol{C}[i,t]$ as the empirical coverage over the mask $\mathcal{V}^{(t)}_{\text{eval}}$. Similarly, $\text{Cov}(\mathbb{I}_T)$ is the empirical coverage for $\mathbb{I}_T$ (average over all records). We visualize $1 - \boldsymbol{C}$ under node exchangeability on Fig. 4.2.

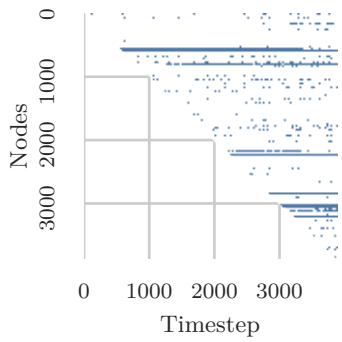

Figure 4.2: $1 - \boldsymbol{C}$ for Cora. Details in § B.2

4.4.1 Node-exchangeable Sequences

A node-inductive graph sequence is node-exchangeable if any permutation of node appearance is equally likely – at each step, any unseen node has the same probability of appearing (see § B.2).

With the arrival of new nodes, the distribution of embeddings (and consequently scores) shifts (as shown in Fig. 4.1). This is why calibrating prior to the update fails to guarantee coverage for new nodes. However, with node-exchangeability, calibration and evaluation scores computed conditional to a specific subgraph are still exchangeable. We extend Theorem 4.2 to inductive GNNs on node-exchangeable graph sequences for any subset of evaluated vertices.

[7] All results can be trivially generalized to the case where the calibration nodes are scattered in the sequence. This only affects the width of the coverage interval (the variance) as it is directly a function of the number of available calibration nodes when computing prediction sets.

Proposition 4.3. *At time step t, with graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ from a node-exchangeable sequence, and a calibration set $\mathcal{V}_{\text{cal}}$ consisting of the first n nodes in $\mathcal{G}_t$, for any $v_j \in \mathcal{V}_{\text{eval}}^{(t)} \subseteq \mathcal{V}_t \setminus \mathcal{V}_{\text{cal}}$ define*

$$C^{(t)}(v_j) = \left\{ y : s(v_j, y \mid \mathcal{G}_t) \geq \mathbb{Q}\left(\alpha; \{s(v_i, y_i \mid \mathcal{G}_t)\}_{i \in \mathcal{V}_{\text{cal}}}\right) 1 \right\} \tag{39}$$

Then $\mathbb{P}\left[y_j \in C^{(t)}(v_j) \mid v_j \in \mathcal{V}_{\text{eval}}^{(t)}\right] \geq 1 - \alpha$. Moreover with $m := |\mathcal{V}_{\text{eval}}^{(t)}|$,

$$\mathbb{P}\left[\text{Cov}(\mathcal{V}_{\text{eval}}^{(t)}) \leq \beta\right] = 1 - \Phi_{\text{HG}}(i_\alpha - 1; m + n, n, i_\alpha + \lceil \beta t \rceil) \tag{40}$$

We deffer proofs to § B.2. Here we provide an intuitive justification of the theorem. First, if $\mathcal{V}_{\text{eval}}^{(t)}$ includes all vertices the problem reduces to the transductive case, for which Theorem 4.2 applies with scores conditional on the current graph. Otherwise, $\mathcal{V}' := \mathcal{V}_t \setminus \mathcal{V}_{\text{eval}}^{(t)}$ is not empty. Here the effect of $\mathcal{V}'$ is symmetric to $\mathcal{V}_{\text{cal}}$ and $\mathcal{V}_{\text{eval}}^{(t)}$. For instance, consider a linear message passing $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{X}\boldsymbol{W}$ with weight matrix $\boldsymbol{W}$ (our results hold in general). From standard CP we know that for row-exchangeable matrix $\boldsymbol{X}$, $\boldsymbol{X}\boldsymbol{W}$ is also row-exchangeable (a linear layer is permutation equivariant). Hence, we just question the row-exchangeability of $\boldsymbol{A}\boldsymbol{X}$. Split $\boldsymbol{X}$ into $[\boldsymbol{X}_{\text{cal}}^\top \mid \boldsymbol{X}_{\text{eval}}^\top \mid \boldsymbol{X}_{\mathcal{V}'}^\top]^\top$, and similarly split $\boldsymbol{A}$ into nine blocks based on the endpoints of each edge. We have that $\boldsymbol{A}\boldsymbol{X}$ equals

$$\begin{bmatrix} \boldsymbol{A}_{\text{cal}\cdot\text{cal}} & \boldsymbol{A}_{\text{cal}\cdot\text{eval}} & \boldsymbol{A}_{\text{cal}\cdot\mathcal{V}'} \\ \boldsymbol{A}_{\text{eval}\cdot\text{cal}} & \boldsymbol{A}_{\text{eval}\cdot\text{eval}} & \boldsymbol{A}_{\text{eval}\cdot\mathcal{V}'} \\ \boldsymbol{A}_{\mathcal{V}'\cdot\text{cal}} & \boldsymbol{A}_{\mathcal{V}'\cdot\text{eval}} & \boldsymbol{A}_{\mathcal{V}'\cdot\mathcal{V}'} \end{bmatrix} \boldsymbol{X} = \underbrace{\begin{bmatrix} \boldsymbol{A}_{\text{cal}\cdot\text{cal}}\boldsymbol{X}_{\text{cal}} + \boldsymbol{A}_{\text{cal}\cdot\text{eval}}\boldsymbol{X}_{\text{eval}} \\ \boldsymbol{A}_{\text{evalcal}}\boldsymbol{X}_{\text{cal}} + \boldsymbol{A}_{\text{eval}\cdot\text{eval}}\boldsymbol{X}_{\text{eval}} \\ \cdots \end{bmatrix}}_{\text{Mat(1)}} + \underbrace{\begin{bmatrix} \boldsymbol{A}_{\text{cal}\cdot\mathcal{V}'}\boldsymbol{X}_{\mathcal{V}'} \\ \boldsymbol{A}_{\text{eval}\cdot\mathcal{V}'}\boldsymbol{X}_{\mathcal{V}'} \\ \cdots \end{bmatrix}}_{\text{Mat(2)}}$$

For the guarantee to be valid, the first $|\mathcal{V}_{\text{cal}}| + |\mathcal{V}_{\text{eval}}|$ rows should be exchangeable. For Mat(1), this already holds due to node-exchangeability and Theorem 4.2. With $\boldsymbol{X}_{\mathcal{V}'}$ being common in both blocks of Mat(2), we only need $\boldsymbol{A}_{\text{cal}\cdot\mathcal{V}'}$ and $\boldsymbol{A}_{\text{eval}\cdot\mathcal{V}'}$ to be row-exchangeable which again holds due to node exchangeability. Hence, the effect of $\mathcal{V}'$ is symmetric to calibration and evaluation sets. In other words, the shifted embedding still preserves the guarantee conditional to the graph at timestep t. The only requirement is to compute the conformal scores for all nodes including the calibration set and *dynamically update* the quantile threshold – the threshold depends on t. Another way to understand the theorem is that for any column in $\boldsymbol{C}$, the expectation of any subset of elements $\geq 1 - \alpha$. Next, we generalize this guarantee to any evaluation mask independent of time.

Theorem 4.4. *On a node-exchangeable graph sequence, with exchangeably sampled $\mathcal{V}_{\text{cal}}$ consisting of the first n nodes in $\mathcal{G}_t$, for any valid partition $\mathbb{I}_T = \cup_{i=1}^T \mathcal{V}_{\text{eval}}^{(i)}$ we have*

$$\mathbb{P}\left[y_i \in C^{(t)}(v_i) \mid v_i \in \mathcal{V}_{\text{eval}}^{(t)}\right] \geq 1 - \alpha \tag{41}$$

Moreover, it holds that $\mathbb{P}\left[\text{Cov}(\mathbb{I}_T) \leq \beta\right] = 1 - \Phi_{\text{HG}}(i_\alpha; |\mathbb{I}_T| + n, n, i_\alpha + \lceil |\mathbb{I}_T| \cdot \beta \rceil)$

Theorem 4.4 indicates that CP applies to inductive GNNs conditional to the subgraph in which each node is evaluated. We will leverage this insight in § 4.5. In conclusion, with a node-exchangeable graph sequence, for any node v_i appearing at timestep t, and evaluated at any timestep $t' \geq t$, it holds that $\mathbb{P}\left[y_i \in C^{(t')}(v_i)\right] \geq 1 - \alpha$. Two special

cases of this result are evaluation upon appearance (the diagonal of $\boldsymbol{C}$) and evaluation all at once (the last column of $\boldsymbol{C}$) which follows from Theorem 4.2. The latter is also equivalent to the simultaneous-inductive setting in Zargarbashi et al. (2023a). The user is flexible to delay the prediction to any time, still preserving the guarantee.

4.4.2 Edge-exchangeable Sequences

In an edge-inductive sequence at each timestep an edge is added, $\mathcal{G}_{t+1} = (\mathcal{V}_t \cup V(e_{t+1}), \mathcal{E}_t \cup e_{t+1})$.[8] This edge may introduce new nodes or connect existing ones. When all permutations μ of the edge-sequence are equally likely, $\mathbb{P}\left[(e_1, \dots e_m)\right] = \mathbb{P}\left[(\mu(e_1), \dots \mu(e_m))\right]$, the sequence is edge-exchangeable (see § B.2 for a formal definition). We address this setting using a special case of weighted quantile lemma (Tibshirani et al., 2019b) with weights defined by the frequency of elements.

Lemma 4.5. *Let $\mathcal{X} = \{x_1, \cdots, x_m\}$ be exchangeable random variables and $f : 2^{\mathcal{X}} \mapsto \mathbb{R}$ be a mapping defined on subsets of $\mathcal{X}$. For any partitioning $\cup_{i=1}^{n+1} \mathcal{X}_i = \mathcal{X}$ and $z_i := f(\mathcal{X}_i)$ we have:*

$$\mathbb{P}\left[z_{n+1} \leq \mathbb{Q}\left(\beta; \{z_i\}_{i=1}^{n} \cup \{\infty\}\right) \left\{ \frac{1}{|\mathcal{X}_i|} \right\}_{i=1}^{n+1} \right] \geq \beta$$

We introduce the edge-exchangeable (EdgeEx) CP and prove its guarantee by showing that at any timestep t, an edge exchangeable sequence, is decomposed into weighted node exchangeable subsequences with weights equal to $1/\deg(v)$. Then, on each subsequence weighted CP maintains a valid guarantee. In this setup, there are no isolated nodes, any node can be evaluated upon appearance.

Theorem 4.6. *At each timestep t, given graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ from an edge-exchangeable sequence and with a calibration set $\mathcal{V}_{\text{cal}}$, define $q = \mathbb{Q}\left(\alpha; \{s_i\}_{v_i \in \mathcal{V}_{\text{cal}}}\right) \left\{1/\deg(\mathrm{v_i})_{v_i \in \mathcal{V}_{\text{cal}}}\right\}$, and for any $v_j \in \mathcal{V}_{\text{eval}}^{(t)}$ define $C^{(t)}(v_j) := \left\{y : s(v_j, y) \geq q\right\}$. We have $\mathbb{P}\left[y_j \in C^{(t)}(v_j)\right] \geq 1 - \alpha$.*

Via Theorem 4.6, EdgeEx CP is guaranteed to provide $1 - \alpha$ coverage. The result holds for each timestep t, conditional to that timestep.

4.5 Node Exchangeable and Edge Exchangeable CP

Building upon the theory in § 4.4, for node-exchangeable graph sequences we define node-exchangeable (NodeEx) CP with coverage guarantee conditional to the subgraph at each timestep. Recall that the shift in scores upon changes in the graph is symmetric for calibration and evaluation nodes. Hence, NodeEx recomputes the calibration scores with respect to the additional context. In an edge-exchangeable sequence via Theorem 4.6, EdgeEx CP – weighted NodeEx CP with $w_i = 1/\deg(v_i)$, results in a similar valid guarantee. Any further details about NodeEx also generalizes to the EdgeEx via Theorem 4.6.

With both settings at each timestep t we have seen a set of nodes $\mathcal{V}_t$. Let $\mathcal{V}_{\text{eval}}^{(t)} \subset \mathcal{V}_t$ be the set of nodes evaluated at timestep t. This set can contain any node from the past as long as they are not already evaluated. Specifically, we cannot use the prediction sets of a particular node multiple times to compare and pick a specific one (see subsubsection B.4.1 for a discussion). We always assume that the calibration set is the

[8] Similar results apply for updates with more than one edge.

union of all nodes appeared up to timestep $t = T_0$ and the test set contains the remaining nodes. For a sequence of timesteps and corresponding evaluation sets $\left\{(t_i, \mathcal{V}^{t_i}_{\text{eval}})\right\}^{T}_{i=T_0+1}$ where $\cup^{T}_{i=T_0+1}\mathcal{V}^{t_i}_{\text{eval}} = \mathcal{V}_T - \mathcal{V}_{\text{cal}}$, NodeEx CP (see § B.1 for algorithm) returns valid prediction sets. We use EdgeEx CP for edge-exchangeable sequences.

CP on node-conditional random subgraphs. Another application of Proposition 4.3 is to produce subgraph conditional prediction sets even for transductive node-classification. For each node v_{test}, we define K subgraphs each including $\{v_{\text{test}}\}$, $\mathcal{V}_{\text{cal}}$ and randomly sampled $\mathcal{V}' \subset \mathcal{V}_{\mathcal{G}}$. Then, we average the scores and return prediction sets. In another approach, we can run K concurrent CPs, each resulting in a prediction set. Then we combine them with a *randomized* voting mechanism to a final prediction set. The resulting prediction sets include true labels with $1 - \alpha$ probability. This approach works for any number of subgraphs. See subsubsection B.4.1 for further explanations.

Relation to full conformal prediction. In our paper, we technically adopted the split conformal[9] approach where model training is separate from calibration due to its computational efficiency. An alternative is full conformal prediction, where to obtain a score for a (calibration) node $v_i \in \mathcal{V}_{\text{cal}}$ and any candidate label y' w.r.t. a test node v_{test} we train a model from scratch on $\mathcal{V}_{\text{cal}} \cup \{(v_{\text{test}}, y')\}$ and consider its prediction for v_i. This involves training $|\mathcal{V}_{\text{cal}}| + 1$ models for each y' and each test node, which is extremely expensive but can use all[10] available labels. Transductive node classification can be seen as a middle ground between split and full conformal as explained by Huang et al. (2023a). Our approach is similarly in between – instead of retraining the model from scratch we simply update the embeddings (and thus scores) by performing message-passing with the newly arrived nodes/edges.

4.6 Experimental Evaluation

4.6.1 Experimental setup and discussion of metrics

We evaluate our approach for both node and edge exchangeable sequences where we have three options to make predictions: (i) upon node arrival, (ii) at a given fixed timestep shared for all nodes, (iii) at an arbitrary node-specific timestep (we choose at random). We compare our approach with naive CP where the coverage guarantee will not hold. Later we discuss why both over- and under-coverage are invalid. For completeness, we compare NodeEx CP with NAPS in § B.3 in addition to a detailed overview of its drawbacks. However, as discussed in subsubsection 4.2.1, NAPS is not a suitable baseline. We use APS (Romano et al., 2020) as the base score function while other scores are tested in § B.4.

Models and datasets. We consider 9 different datasets and 4 models: `GCN` Kipf and Welling (2017), `GAT` Veličković et al. (2018), and `APPNP`Klicpera et al. (2019) as structure-aware and `MLP` as a structure-independent model. We evaluate our NodeEx, and EdgeEx CP on the common citation graphs `CoraML` McCallum et al. (2004), `CiteSeer` Sen et al. (2008), `PubMed` Namata et al. (2012), `Coauthor Physics` and `Coauthor CS` Shchur et al. (2018), and co-purchase graphs `Amazon Photos` and `Computers` McAuley et al. (2015); Shchur et al. (2018) (details in § B.5). Results for Flickr (Young et al., 2014), and Reddit2 (Zeng et al., 2019) (with `GCN` model and their original split) datasets are in subsubsection B.4.5. Here, the model only affects the efficiency and not the validity.

[9]Sometimes called inductive conformal prediction. However, this is an orthogonal use of the word inductive.

[10]This is unlike split conformal that needs to split the available labels into training and calibration sets.

Evaluation procedure. For any of the mentioned datasets, we sample 20 nodes per class for training and 20 nodes for validation with stratified sampling. For the node-exchangeable setup, the calibration set has the same size as the training. For the edge-exchangeable sequence, the same number of *edges* are sampled. Therefore, each round of simulation has a potentially different number of calibration nodes for the edge-exchangeable setup. First, we take a random sample of nodes as train/val set and train the model on the resulting subgraph $\mathcal{G}_0$. Then, for the node-exchangeable sequence, we first sample a calibration set randomly from the remaining nodes. Then, at each timestep, we add a random unseen node to the graph (with all edges to the existing nodes) and predict the class probability given the updated subgraph. We use the updated conformal scores to create prediction sets for *all* existing test nodes until time t and record the empirical coverage results in column t of the coverage matrix $\boldsymbol{C}$. Similarly, for the edge-exchangeable sampling, we sample calibration *edges*, and take both ends as calibration nodes. The remaining edges are sampled one at a time. The rest of the procedure is the same as node-exchangeable setting and results in an analogous $\boldsymbol{C}$.

Challenges with small calibration sets. With a limited number of samples in the calibration set, there will be an additional error in the coverage due to the discrete quantile function with a low sample rate. In practice, we take the $\lfloor n/(n+1) \cdot \alpha \rfloor$ index of a discrete array as the conformal threshold which is often not exactly equal to $1-\alpha$ quantile in the continuous domain. Therefore, we expect $1/(|\mathcal{V}_{\text{cal}}|+1)$ error around $1-\alpha$. This error is in addition to the variance of the Beta or Hyper-geometric distribution and will converge to 0 with increasing calibration set size.

Distance from target coverage. Our main evaluation criteria is the distance of empirical coverage w.r.t. the target $1-\alpha$. For all datasets we set the desired coverage to 90%, and we report the (absolute) distance w.r.t. this value (see § B.5). Each reported result is an average of 10 different random node-exchangeable sequences and 15 different random edge-exchangeable sequences. Note that due to homophily, we observe a higher empirical coverage in the non-exchangeable case which can wrongly be interpreted as being better. To address any potential confusion, first the guarantee is invalid if it breaks either the lower or the upper-bound. In real-world deployment we do not have ground-truth labels to calculate empirical coverage, and if the guarantee is broken (in either direction) the output is unreliable – nullifying the main goal of CP. Second, a higher empirical coverage also results in larger set sizes making the prediction sets less useful (see also Fig. 4.4).

Efficiency and singletons. In addition to coverage, which is the main goal of our work, we also briefly study the efficiency (average set size) and the singleton hit – fraction of correct sets of size 1. As we will see in subsubsection 4.6.2, our NodeEx and EdgeEx CP improve these metrics as a byproduct. Some scoring functions such as RAPS (Angelopoulos et al., 2020) and DAPS (Zargarbashi et al., 2023a) directly target these metrics and can be used on top of our approach. We explore DAPS in subsubsection B.4.3.

4.6.2 Evaluation of empirical coverage, set size and singleton hit

Table 4.1 shows the deviation from the coverage guarantee for different datasets when the label of each node is predicted *upon its arrival*. As shown in the table, while naive CP shows a significant shift from the $1-\alpha = 0.9$ coverage, NodeEx CP maintains the empirical coverage close to the desired value. In Fig. 4.3 (upper left) we show the temporal evolution of coverage. We show the coverage for all nodes that arrived until

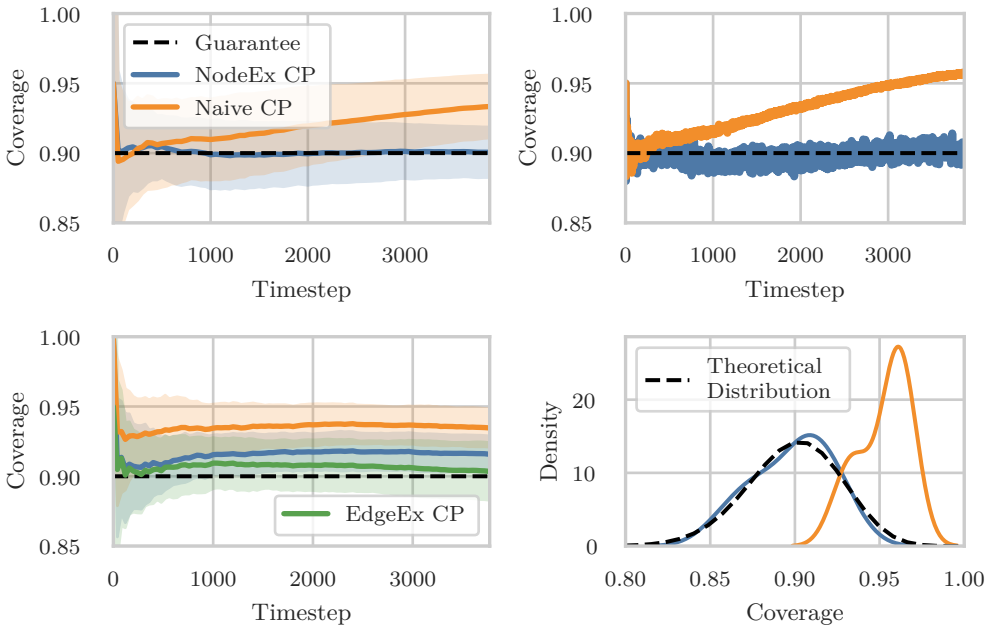

Figure 4.3: [Upper left] Coverage over time under node exchangeability when predicting upon node arrival (diagonals of $\boldsymbol{C}$). [Upper right] Coverage when we instead predict at a fixed time (columns of $\boldsymbol{C}$). [Lower right] Same as upper right but under edge-exchangeability. [Lower left] The empirical distribution of coverage when we predict at node-specific times (fixed entries of $\boldsymbol{C}$), compared to the theoretical distribution. Sample size results in a slight *expected* shift. The transparent lines show a particular sequence, and the thick solid lines shows the average over 10 (15) sequences.

time t and see that the guarantee is preserved for each timestep t. As mentioned before, the goal is to achieve near $1-\alpha$ empirical coverage, the over-coverage of standard CP might misleadingly appear as a better result. Not only does over-coverage come at the cost of less efficiency as shown in Fig. 4.4 (see subsubsection B.4.2 for details), but the empirical coverage of non-guaranteed prediction sets is not predictable apriori.

Fig. 4.3(lower left) shows the same experiment for edge-exchangeability. While NodeEx CP guarantees coverage under node-exchangeability, the weights specified in our EdgeEx CP are necessary for the guarantee to hold on an edge-exchangeable sequence. Standard CP fails again.

As explained in § 4.4 any node can be evaluated at *any timestep* after its appearance. Fig. 4.3 (upper right) shows the result when predicting at a fixed time (instead of upon arrival). Additionally, Fig. 4.3 (lower right) shows the empirical distribution of coverage for node-specific times (random subsets of nodes, each node predicted at a random time after appearance). For NodeEx CP, the empirical distribution matches the theoretical one, while the coverage of naive CP substantially diverges. Unsurprisingly, under node exchangeability, in Fig. 4.4 we see that our NodeEx CP improves both additional metrics – has smaller sets and larger singleton hit ratio. The same holds for other settings.

In all experiments, we consider a sparse (and thus realistic) calibration set size, e.g. for `PubMed` we sample 60 nodes for calibration, which is a significantly lower number compared to other tasks like image classification with 1000 datapoints for same purpose (Angelopoulos et al., 2020). The calibration size controls the concentration (variance) around $1-\alpha$ as reflected by the transparent lines on Fig. 4.3. Increasing the set size

reduces the variance but the mean is always $1 - \alpha$.

Limitations. We identified three main limitations. First, the guarantee is marginal. Second, real-world graphs may not satisfy node- or edge-exchagability. This can be partially mitigated by the beyond exchangeability framework. Finally, the guarantee does not hold for adversarially chosen evaluation sets $\mathcal{V}_{\text{eval}}$. In other words, the choice of which nodes is evaluated at which timesteps must be prior to observing the prediction set. We provide a longer discussion on each of these limitations in § B.1. Our main focus is on the validity. However, we also reported other metrics such as the average set size, singleton hit ratio and other score functions in § B.4.

		Node Exch.		Edge Exch.		
Dataset	Acc	Sub. CP	Std. CP	W. Sub. CP	Sub. CP	Std. CP
Cora-ML	0.800	**0.280**	5.860	**1.929**	3.883	6.860
PubMed	0.777	**1.254**	4.649	**1.241**	3.405	5.315
CiteSeer	0.816	**0.039**	4.150	**0.335**	1.572	3.460
Coauth-CS	0.914	**0.397**	4.082	**3.024**	4.662	7.835
Coauth-Phy.	0.940	**0.555**	2.689	**2.240**	4.378	6.123
Amz-Computers	0.788	**0.263**	6.373	**2.687**	5.727	7.036
Amz-Photo	0.868	**0.127**	3.483	**2.546**	4.130	6.613

Table 4.1: Average deviation from guarantee (in percentage (%), for `GCN` model and 1 train/val split).

4.7 Conclusion

We adapt conformal prediction to inductive node-classification for both node and edge exchangeable graph sequences. We show that although introducing new nodes/edges causes a distribution shift in the conformity scores, this shift is symmetric. By recomputing the scores conditional to the evaluation subgraph, we recover the coverage guarantee. Under edge-exchangeability, we need to also account for the node degrees to maintain validity. Importantly, our approach affords flexibility – the guarantee holds regardless of prediction time which can be chosen differently for each node.

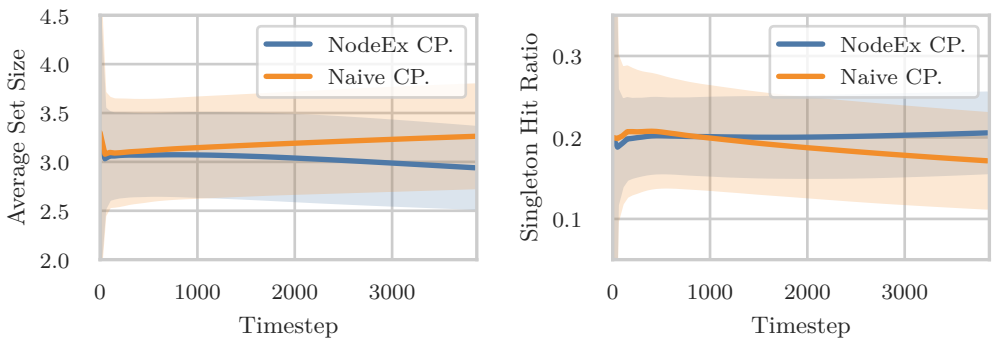

Figure 4.4: [Left] Average set size (lower is better) and [Right] singleton hits ratio (higher is better) of naive CP vs. NodeEx CP for `CiteSeer` and `GCN`. Our approach improves both metrics.

5 Robust Yet Efficient Conformal Prediction Sets

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).

5.1 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 often require retraining or modifications in the model architecture (Abdar et al., 2020). 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 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. (2023a).

Most models suffer a significant performance drop when fed noisy or manipulated data, even for indistinguishable (label-preserving) perturbations (Silva and 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 § 5.2). Unlike heuristic defenses which are easily overcome by new attacks (Athalye et al., 2018; Mujkanovic et al., 2023), certificates provide worst-case guarantees that the prediction does not change. How can we extend certificates to conformal prediction sets?

Given calibration data and a score function $s : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ that captures conformity between inputs and potential labels, the prediction sets $C_\alpha(\boldsymbol{x}) = \{y : s(\boldsymbol{x}, y) \geq q_\alpha\}$ include all labels with scores above a calibrated threshold q_α. CP guarantees that $\mathbb{P}\left[y_{\text{true}} \in C_\alpha(\boldsymbol{x})\right] \geq 1 - \alpha$ for a clean $\boldsymbol{x}$ and any user-specfied α. To certify robustness, we can define *conservative* sets that ensure that the coverage remains above the nominal $1 - \alpha$ 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 consider 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).[11] We then propose an alternative approach that bounds 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.

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. 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.

5.2 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 an arbitrary coverage probability $1-\alpha$, for any test point $\boldsymbol{x}_{n+1}$, CP defines a prediction set $C_\alpha(\boldsymbol{x}_{n+1}) \subseteq \mathcal{Y}$ that is guaranteed to cover the true label y_{n+1} with the predetermined probability.

Theorem 5.1 (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 sets defined as $C_\alpha(\boldsymbol{x}_{n+1}) = \{y : s(\boldsymbol{x}_{n+1}, y) \geq q_\alpha\}$ have coverage probability*

$$\mathbb{P}\left[y_{n+1} \in C_\alpha(\boldsymbol{x}_{n+1})\right] \geq 1-\alpha \tag{42}$$

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

This theorem was extended to graphs (Zargarbashi et al., 2023a; Huang et al., 2023a) 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 predicted probabilities (softmax) (Sadinle et al., 2018) $s(\boldsymbol{x}, y) = \pi(\boldsymbol{x}, y)$. TPS tends to over-cover easy examples and under-cover hard ones (Angelopoulos and 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 (for an extended introduction to CP see § C.1).

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

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

$\boldsymbol{x}$ 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.3, 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 $C_\alpha(\tilde{\boldsymbol{x}})$ which breaks the guarantee in Eq. 42. Here we assume that CP has been calibrated with a clean (unperturbed) calibration set.

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. Thus, the goal of the adversary is to instead perturb the *calibration* set in order to decrease the empirical coverage – breaking the guarantee (see formal definition in subsubsection 5.3.2).

5.3 Robust Prediction Sets

5.3.1 Robustness to Evasion Attacks

Definition 5.2 (Robust coverage (Gendler et al., 2021))**.** The prediction sets $C_\alpha(\cdot)$ have adversarially robust $1 - \alpha$ coverage if for any $(\boldsymbol{x}_{n+1}, y_{n+1})$ exchangeable with $\mathcal{D}_{\text{cal}}$

$$\mathbb{P}\left[y_{n+1} \in C_\alpha(\tilde{\boldsymbol{x}}_{n+1}) \mid \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})\right] > 1 - \alpha \tag{43}$$

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, $\forall \tilde{\boldsymbol{x}} \in \mathcal{B}_r(\boldsymbol{x}) : s_{\text{rscp}}(\tilde{\boldsymbol{x}}, y) \leq s_{\text{rscp}}(\boldsymbol{x}, y) + \frac{r}{\sigma}$, they shift the quantile $q_\alpha - \frac{r}{\sigma}$ to ensure robustness (details in § C.6). Instead of shifting the quantile we directly consider the upper bounds on the scores which is a slight generalization.

Proposition 5.3. *Let $\overline{s}(\boldsymbol{x}, y)$ and $\underline{s}(\boldsymbol{x}, y)$ be upper and lower bounds 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 $\overline{C}_\alpha(\boldsymbol{x}) = \{y : \overline{s}(\boldsymbol{x}, y) \geq q_\alpha\}$. 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*

$$\mathbb{P}\left[y_{n+1} \in \overline{C}_\alpha(\tilde{\boldsymbol{x}}_{n+1})\right] \geq 1 - \alpha \tag{44}$$

All omitted proofs are in subsubsection C.4.1. RSCP is a special case with $\overline{s}(\boldsymbol{x}, y) = s_{\text{rscp}}(\boldsymbol{x}, y) + \frac{r}{\sigma}$. We can equivalently rewrite $s_{\text{rscp}}(\boldsymbol{x}, y)$ to match the bound from Kumar et al. (2020) (see subsubsection C.6.1). In § 5.4 we improve the bound using the CDF. Tighter bounds result in smaller (more efficient) sets.

5.3.2 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 must 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 $\tilde{q}_\alpha = \mathbb{Q}\left(\alpha; \tilde{\mathcal{D}}_{\text{cal}}\right)$ as right as possible compared to q_α.[12] This shift increases the probability of rejecting true labels, resulting in a lower coverage. Namely, for $\beta = \text{Quant}^{-1}(\tilde{q}_\alpha; \mathcal{D}_{\text{cal}})$, the quantile inverse of the poisoned threshold $\tilde{q}$ w.r.t. the clean calibration set, the poisoned calibration set results in near $1-\beta$ coverage where by definition $1-\beta \leq 1-\alpha$. Given a potentially poisoned calibration set $\tilde{\mathcal{D}}_{\text{cal}}$ we certify the prediction sets via the following optimization problem:

$$\begin{aligned} q^* = \min_{z_i \in \mathcal{X}} \quad & q \\ \text{s.t.} \quad & q = \mathbb{Q}\left(\alpha; \{s(\boldsymbol{z}_i, y_i) : (\tilde{\boldsymbol{x}}_i, y_i) \in \tilde{\mathcal{D}}_{\text{cal}}\}\right) \\ & \forall i \leq n : \boldsymbol{z}_i \in \mathcal{B}(\tilde{\boldsymbol{x}}_i) \quad \text{and} \quad \mathbf{1}[\boldsymbol{z}_i \neq \tilde{\boldsymbol{x}}_i] \leq k \end{aligned} \tag{45}$$

The problem in Eq. 45 finds the most conservative quantile q^* and it holds that $q^* \leq q_\alpha$ 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. 45 certifies *at least* $1-\alpha$ coverage.

Proposition 5.4. *Let q^* to be the solution to the optimization problem in Eq. 45. The conservative prediction sets*

$$\overline{C}_\alpha(\boldsymbol{x}_{n+1}) = \{y_i : s(\boldsymbol{x}_{n+1}, y_i) \geq q^*\} \tag{46}$$

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

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. 45 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 § 5.4 how to obtain such bounds using randomized smoothing which we can use in both Proposition 5.3 and Proposition 5.4. Any other bounds are also applicable.

5.3.3 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:

$$\begin{aligned} q^* = \min_{z_i \in \mathcal{Y}} \quad & q \\ \text{s.t.} \quad & q = \mathbb{Q}\left(\alpha; \left\{s(\boldsymbol{x}_i, z_i) : (\boldsymbol{x}_i, \tilde{y}_i) \in \tilde{\mathcal{D}}_{\text{cal}}\right\}\right) \\ & \mathbf{1}[z_i \neq \tilde{y}_i] \leq k \end{aligned} \tag{47}$$

[12] 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 § C.1)

As before, since $q^* \leq q_\alpha$ we have that constructing the set as in Eq. 46 we maintain $\geq 1 - \alpha$ coverage even under worst-case label perturbation. We can solve both problems (Eq. 45 and Eq. 47) by writing them as mixed-interger linear programs. We provide technical details in § C.7.

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 § C.8.

5.4 Randomized Smoothing Bounds

To instantiate the conservative sets $\overline{C}_\alpha(\cdot)$ defined in § 5.3 we need bounds on the worst-case change in the 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.3 for more details. Our approach works with other smoothing schemes such as uniform noise for l_1 threat models (Levine and Feizi, 2021), but we focus on these two due to their popularity.

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. We have:

$$\max_{\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})} \mathbb{E}[s(\xi(\tilde{\boldsymbol{x}}, y)] \leq \max_{\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x}), h \in \mathcal{H}} \mathbb{E}[h(\xi(\tilde{\boldsymbol{x}}, y)] \tag{48}$$

Since, by definition $s(\cdot, \cdot) \in \mathcal{H}$, by maximizing over $\mathcal{H}$ we get an upper bound on $\hat{s}(\cdot, \cdot)$.

The solution to Eq. 48 is trivial unless we 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 point-wise bound for a given $(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ pair since it can be shown that the maximum in Eq. 48 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.3).

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

$$\begin{aligned} \overline{s}_{\text{mean}}(\tilde{\boldsymbol{x}}, y) = \max_{h \in \mathcal{H}} \quad & \mathbb{E}[h(\xi(\tilde{\boldsymbol{x}}), y)] \\ \text{s.t.} \quad & \mathbb{E}[h(\xi(\boldsymbol{x}), y)] = \hat{s}(\boldsymbol{x}, y) \end{aligned} \tag{49}$$

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 $a = b_1 \leq b_2 \leq \dots \leq b_m = b$ be m real numbers that partition the output space. Let $p_i = \mathbb{P}\left[s(\xi(x), y) \leq b_i\right]$. We define the problem:

$$\begin{aligned} \overline{s}_{\text{cdf}}(\tilde{\boldsymbol{x}}, y) = \max_{h \in \mathcal{H}} \quad & \mathbb{E}[h(\xi(\tilde{\boldsymbol{x}}), y)] \\ \text{s.t.} \quad \forall b_i : \quad & \mathbb{P}\left[h(\xi(\boldsymbol{x}), y) \leq b_i\right] = p_i \end{aligned} \tag{50}$$

The key insight for solving Eq. 50 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 $F_h(b_j) = \mathbb{P}\left[h(\boldsymbol{x}, y) \leq b_j\right]$, for any function h

$$\begin{aligned} \mathbb{E}[h(\boldsymbol{x})] &\leq \sum_{j=2}^{m} b_j \cdot [(F_h(b_j) - F_h(b_{j-1})] \\ &= b_m - \sum_{j=2}^{m-1} F_h(b_j) \cdot (b_{j+1} - b_j) \end{aligned} \tag{51}$$

Next, we show how to solve both problem for the two different smoothing schemes. For Gaussian smoothing both problems in Eq. 49 and Eq. 50 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. 49. We extend their approach to also solve Eq. 50 which is a novel contribution of potentially independent interest, e.g. to certify graph neural networks with regression tasks.

Bounds for Gaussian smoothing. For any perturbation $\tilde{\boldsymbol{x}}$ 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 $p = \hat{s}(\boldsymbol{x}, y) = \mathbb{E}[s(\xi(\boldsymbol{x}, y)]$. The CDF bound is

$$\overline{s}_{\text{cdf}}(\tilde{\boldsymbol{x}}, y) \leq \sum_{j=2}^{m} \Phi_\sigma\left(\Phi_\sigma^{-1}(p_i) + r\right)(b_i - b_{i-1}) \tag{52}$$

where $p_i = \mathbb{P}\left[s(\xi(\boldsymbol{x}), y) \leq b_i\right]$ and Φ_σ is the CDF of $\mathcal{N}(0, \sigma^2)$. For both cases the lower-bound is similarly computed by flipping the sign of r.

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 $\mathcal{R}_i = \{\boldsymbol{z} : \mathbb{P}[\xi(\boldsymbol{x}) = \boldsymbol{z}] / \mathbb{P}[\xi(\tilde{\boldsymbol{x}}) = \boldsymbol{z}] = c_i\}$. For the mean variant, we greedily distributing 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_i masses in each region and each bin $[b_i, b_{i+1}]$. Technical details, including the linear programming formulations are in § C.3. 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 miliseconds.

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.

5.5 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 pseudo-code in Algorithm 1 that works with any score function. We provide the code in the supplementary material.

Algorithm 1 CDF-Aware Sets (CAS, Evasion)

$q_\alpha = \mathbb{Q}(\alpha; \{\hat{s}(\boldsymbol{x}, y)\}_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{cal}}})$ ▷ Clean quantile
Compute $\overline{s}_{\text{cdf}}(\boldsymbol{x}, y)$, e.g. with Eq. 52 ▷ Upper bound
Return $\overline{C}_\alpha = \{y : \overline{s}_{\text{cdf}}(\boldsymbol{x}, y) \geq q_\alpha\}$ ▷ Conservative set

Poisoning. For poisoning attacks we simply use the conservative threshold q^* from Eq. 45 or Eq. 47 where we use the CDF-bounds in the constraints (see subsubsection 5.3.2). If the test examples are assumed clean we return $\overline{C}_\alpha = \{y : \hat{s}(\boldsymbol{x}, y) \geq q^*\}$. Since robustness to evasion and poisoning are independent, we can achieve simultaneous robustness to both evasion and poisoning via $\overline{C}_\alpha = \{y : \overline{s}_{\text{cdf}}(\boldsymbol{x}, y) \geq q^*\}$.

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. 45 and $|\mathcal{D}_{\text{cal}}| \times |\mathcal{Y}|$ binary variables for Eq. 47. See § C.7 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.

Calibration-time variant. For evasion we need to compute $\overline{s}(\boldsymbol{x}, y)$ via solving Eq. 50 (or Eq. 49) 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 5.5. *Let* $\overline{C}_\alpha(\tilde{\boldsymbol{x}}_{n+1}) = \left\{ y : \hat{s}(\tilde{\boldsymbol{x}}_{n+1}, y) \geq \underline{q}_\alpha \right\}$

$$\underline{q}_\alpha = \mathbb{Q}\left(\alpha; \left\{ \underline{s}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}} \right\}\right) \tag{53}$$

then $\Pr\left[y_{n+1} \in \overline{C}_\alpha(\tilde{\boldsymbol{x}}_{n+1}) \right] \geq 1 - \alpha$ *for* $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$ *and* $(\boldsymbol{x}_{n_1}, y_{n+1})$ *exchangeable with* $\mathcal{D}_{\text{cal}}$. *Moreover, vanilla CP covers the true label with probability* $\geq 1 - \beta$ *where*

$$\beta = \mathbb{Q}^{-1}\left(q_\alpha; \left\{ \underline{s}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}} \right\}\right) \tag{54}$$

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

With Proposition 5.5 we need only $|\mathcal{D}_{\text{cal}}|$ certified bounds as a pre-processing 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 C.1). With Eq. 54 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).

5.6 Finite Sample Correction

Solving Eq. 49, or Eq. 50 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 with adjustable $1 - \eta$ probability. To account for this we calibrate with $\alpha' = \alpha - \eta$ such that the final sets still have $1 - \alpha$ coverage (see § C.5). 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 § C.5 and we propose another approach built on Proposition 5.5. As we show in § 5.7 our correction results in smaller sets and still maintains valid $1 - \alpha$ coverage.

Proposition 5.6. *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* $\underline{q}_{\alpha+} = \mathbb{Q}\left(\alpha - \eta; \left\{ \underline{s}_{\text{cdf+}}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}} \right\}\right)$ *and* $\overline{C}_{\alpha+}(\boldsymbol{x}_{n+1}) = \{ y : \hat{s}_+(\boldsymbol{x}_{n+1}, y) \geq \underline{q}_{\alpha+} \}$. *Then*

$$\Pr\left[y_{n+1} \in \overline{C}_{\alpha+}(\tilde{\boldsymbol{x}}_{n+1}) \right] \geq 1 - \alpha \tag{55}$$

We compute $\underline{s}_{\text{cdf+}}(\boldsymbol{x}_i, y_i)$ by solving the minimization variant of Eq. 50 with CDF error correction through the Dvoretzky–Kiefer–Wolfowitz inequality Dvoretzky et al. (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.

The improvement in the bound from Eq. 49 to Eq. 50 depends on the distribution of randomized scores. The (true) calibration scores have a larger improvement since they are more spread out, compared to the test scores which can be concentrated around 0 for unlikely labels. Moreover, the error correction is relatively stronger for scores closer to 0. Due to both effects our calibration-time correction from Proposition 5.6 is more efficient for CAS compared to the test-time correction as suggested by Yan et al. (2024). For RSCP both corrections are equally good (details in § C.5).

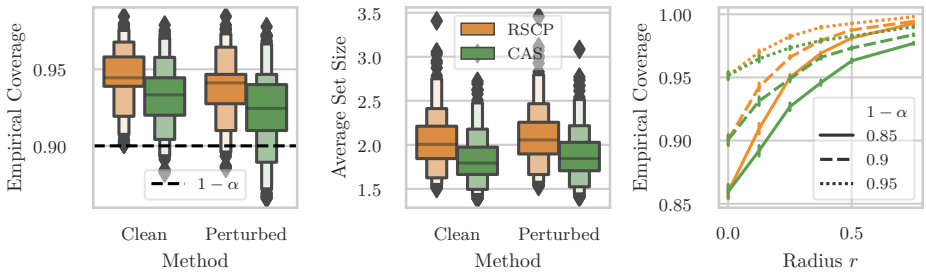

Figure 5.1: Empirical coverage (left) and average set size (middle) of RSCP and CAS for clean and perturbed data. All sets are certified robust up to radius $r = 0.125$. Empirical coverage for different certified radii (right, clean data). All results are for `CIFAR-10` with Gaussian smoothing ($\sigma = 0.25$). CAS is less conservative since it is closer to the nominal $1 - \alpha$, and has smaller sets.

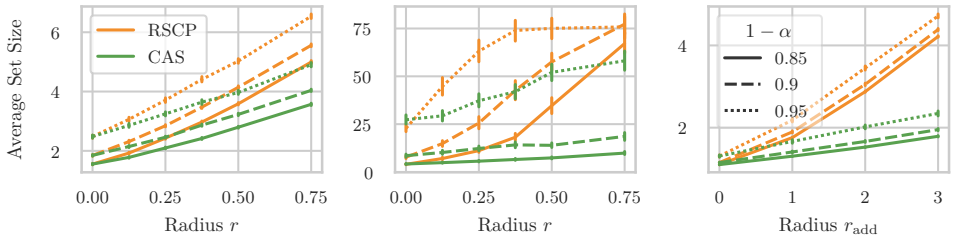

Figure 5.2: Average set size of CAS and RSCP under evasion for (from left to right) `CIFAR-10`, `ImageNet` (with TPS), and `Cora`.

5.7 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.3). 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 (also called efficiency) and the empirical coverage. In § C.10 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` McCallum et al. (2004). We used `ResNet-50` and `ResNet-110` pretrained on `CIFAR-10` and `ImageNet` with noisy data augmentation from Cohen et al. (2019). We trained a model `GCN` Kipf and Welling (2017) for node classification. All models are trained on data augmented with noise. For training the GNN we use 20 nodes per class with stratified sampling as the training set. The validation set is sampled similarly. The size of the calibration set is between 100 and 150 (sparsely labeled setting). We discuss the effect of the calibration set size in § C.1. We use APS as the main score function. We report results with other scores in § C.10. 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 10^4 Monte-Carlo samples. All results are an

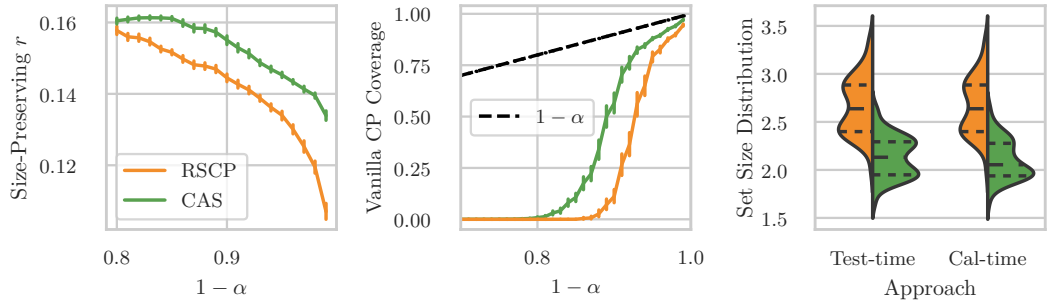

Figure 5.3: Maximum set size-preserving radius (left, mean and variance over test points). Lower bound $1-\beta$ on the robust coverage of vanilla CP (middle, Proposition 5.5). CAS certifies a larger lower bound. Distribution of prediction set sizes using the slower test-time vs. the faster calibration-time evasion certificate. Calibration-time shows slight improvement. All results are for `CIFAR-10`.

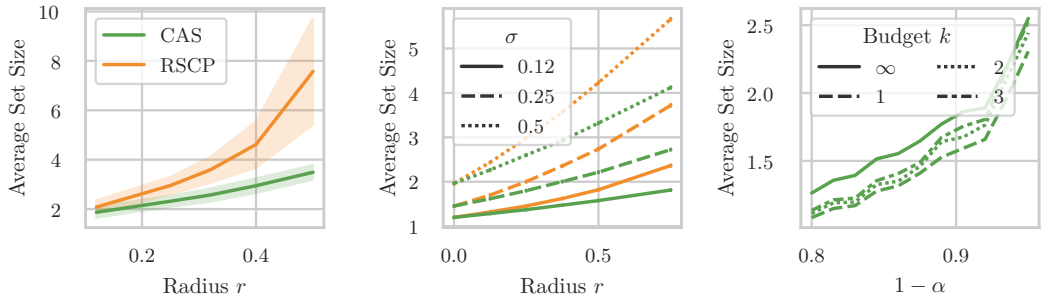

Figure 5.4: Comparison of error corrected RSCP and CAS (left). Effect of various smoothing σ on the average set size (middle). Smaller σ is better. Average set size for feature-poisoning for different perturbation budgets k (right). All results are on `CIFAR-10`.

average of 100 runs with exchangeable calibration sampling (details in § C.10).

Evasion Certificate. Robustness comes at a cost – the conservative sets are necessarily larger than non-robust sets. Consequently, on Fig. 5.1 (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. 5.1 (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. 5.1 (middle). The set size is slightly higher for perturbed inputs.

In Fig. 5.2 we see that CAS's results in smaller prediction sets, across all radii, and all nominal $1-\alpha$ values. The improvement is substantial and also grows with r – for larger radii it is doubled or even tripled, especially on `ImageNet` and `Cora`. Similarly, Fig. 5.3 (left) shows that with CAS we can consistently certify a larger maximum radius "for free".

Calibration-time evasion. Following Proposition 5.5 if we use vanilla CP, ignoring the adversary, we can certify a lower bound $1 - \beta$ on the worst-case robust coverage. In Fig. 5.3 (middle) we see that the certificate based on CAS leads to a better (higher) lower bound. At the same time, Proposition 5.5 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 via the lower bound on the calibration scores. Fig. 5.3 (right) 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.

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 Monte-Carlo samples. While the sets are still asymptotically valid without correction, as confirmed by Fig. 5.1 (left), correction is necessary for a valid certificate as argued by Yan et al. (2024). In Fig. 5.4 (left) we see that the size 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 § C.5).

Ablation study. In Fig. 5.4 (middle) 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, but smaller σ's lead to smaller sets on average.

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. 45). Here, we set $r = 0.12$ for `CIFAR-10` and show additional results in § C.10. In Fig. 5.4 we see that as expected a smaller budget k leads to less conservative sets which translates to smaller set sizes. Interestingly, even with an infinite budget where the attacker is allowed to perturb *all* calibration points the set size does not increase drastically. Making CP robust to poisoning comes at only a small cost. Note, $q^* = \underline{q}_\alpha$ for $k = \infty$ since setting each calibration score to its lower bound is one solution to Eq. 45, which equals calibration-time evasion.

Label poisoning. Next, we study label poisoning where now the attacker can perturb the ground-truth labels of the calibration points. In Table 5.1 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. This complements the results from Einbinder et al. (2022) which show that standard CP is already naturally robust to random (non-worst case) label noise. Finally, we can also combine feature with label poisoning and/or evasion. We discuss these variants in § C.7.

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 $\Pr[y \in C(\boldsymbol{x}) \mid \boldsymbol{x}]$ is impossible to achieve without strong assumptions (Barber et al., 2019a). Achieving near-conditional coverage is still and open problem. This means that CP can have over-coverage or under-coverage for different groups which

Table 5.1: Label poisoning for `CIFAR-10`.

k	Empirical Coverage	Average Set Size
0	0.897	1.41
1	0.916	1.58
2	0.923	1.62

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.

5.8 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 subsubsection C.6.2.

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. (2021) and propose to transform the smooth score (ranking + sigmoid scaling) using an additional holdout set. Anonymous (2024a) 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.

5.9 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 method CAS yields provably robust yet efficient (small) prediction sets.

6 Robust Conformal Prediction with a Single Binary Certificate

Abstract

Conformal prediction (CP) converts any model's output to prediction sets with a guarantee to cover the true label with (adjustable) high probability. Robust CP extends this guarantee to worst-case (adversarial) inputs. Existing baselines achieve robustness by bounding randomly smoothed conformity scores. In practice, they need expensive Monte-Carlo (MC) sampling (e.g. $\sim 10^4$ samples per point) to maintain an acceptable set size. We propose a robust conformal prediction that produces smaller sets even with significantly lower MC samples (e.g. 150 for CIFAR10). Our approach binarizes samples with an adjustable (or automatically adjusted) threshold selected to preserve the coverage guarantee. Remarkably, we prove that robustness can be achieved by computing *only one* binary certificate, unlike previous methods that certify each calibration (or test) point. Thus, our method is faster and returns smaller robust sets. We also eliminate a previous limitation that requires a bounded score function.

6.1 Introduction

Despite their extensive applications, modern neural networks lack reliability as their output probability estimates are uncalibrated (Guo et al., 2017). Many uncertainty quantification methods are computationally expensive, lack compatibility with black-box models, and offer no formal guarantees. Alternatively, conformal prediction (CP) is a statistical post-processing approach that returns prediction *sets* with a guarantee to cover the true label with high adjustable probability. CP only requires a held-out calibration set and offers a distribution-free model-agnostic coverage guarantee (Vovk et al., 2005; Angelopoulos and Bates, 2021). The model is used as a black box to compute conformity scores which capture the agreement between inputs $\boldsymbol{x}$ and labels y. These prediction sets are shown to improve human decision-making both in terms of response time and accuracy (Cresswell et al., 2024). CP assumes exchangeability between the calibration and the test set (a relaxation of the i.i.d. assumption), making it broadly applicable to images, language models, etc. CP also applies on graph node classification (Zargarbashi et al., 2023b; Huang et al., 2023b) where uncertainty quantification methods are limited. However, exchangeability, and therefore the conformal guarantee, easily breaks when the test data is noisy or subjected to adversarial perturbations.

Robust conformal prediction extends this guarantee to worst-case inputs $\tilde{\boldsymbol{x}}$ within a maximum radius around the clean point $\boldsymbol{x}$, e.g. $\forall \tilde{\boldsymbol{x}}$ s.t. $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq r$. In the evasion

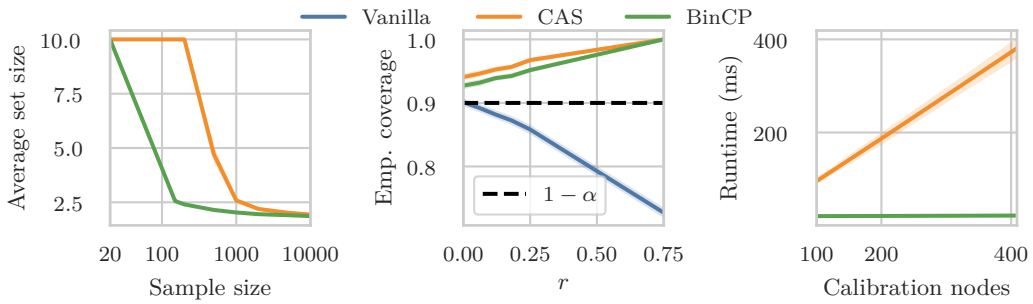

Figure 6.1: [Left] Average set size with different MC sample rates, [Middle] empirical coverage of vanilla and robust CPs under attack, and [Right] runtime of robust CP as a function of calibration datapoints (after computing the MC samples which is the number of lower bound computations).

setting, we assume that the calibration set is clean, and test datapoints can be perturbed. Building on the rich literature of robustness certificates (Kumar et al., 2020), recent robust CP baselines (Gendler et al., 2021; Zargarbashi et al., 2024; Jeary et al., 2024) use a conservative score at test time that is a *certified* bound on the conformity score of the clean unseen input. This maintains the guarantee even for the perturbed input since "if CP covers $\boldsymbol{x}$, then robust CP certifiably covers $\tilde{\boldsymbol{x}}$". However, the average set size increases, especially if the bounds are loose. The certified bounds can be derived through model-dependent verifiers (Jeary et al., 2024) or smoothing-based black-box certificates (Zargarbashi et al., 2024).

For the robustness of black-box models, an established approach is to certify the confidence score through randomized smoothing (Kumar et al., 2020), obtaining bounds on the expected smooth score. The tightness of these bounds depends on the information about the smooth score around the given input, e.g. the mean Yan et al. (2024), or the CDF Zargarbashi et al. (2024). Such methods: (i) assume the conformity score function has a bounded range, (ii) compute several certificates for each calibration (or test) point, and (iii) need a large number of Monte-Carlo samples to get tight confidence intervals. For the current SOTA method CAS (Zargarbashi et al., 2024), the accounting for sample correction inflates the prediction sets significantly for sample rates below 2000 (see Fig. 6.1-left). This inefficiency increases to trivially returning $\mathcal{Y}$ as the prediction set when we run with higher coverage rates or higher radii (see § 6.6). In contrast, we obtain robust and small prediction sets with only $\sim$ 150 MC samples. Additionally, these methods require computing certified bounds for (at least) each calibration point which we further show is a wasteful computation.

BinCP. We observe that smooth inference is inherently more robust. Even without certificates, randomized methods show a slower decrease in coverage under attack (see Fig. 6.6-right). Given any score function $s(\boldsymbol{x}, y)$ capturing conformity, Zargarbashi et al. (2024) and Gendler et al. (2021) define the smooth score as $\bar{s}(\boldsymbol{x}, y) = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma\boldsymbol{I})}[s(\boldsymbol{x}+\boldsymbol{\epsilon}, y)]$. Instead, we perform binarization via a threshold τ, i.e. $\bar{s}(\boldsymbol{x}, y) = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma\boldsymbol{I})}[\mathbb{I}[s(\boldsymbol{x}+\boldsymbol{\epsilon}, y) \geq \tau]] = \Pr_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma\boldsymbol{I})}[s(\boldsymbol{x}+\boldsymbol{\epsilon}, y) \geq \tau]$. Both are valid conformity scores, and both change slowly around any $\boldsymbol{x}$, however, our binarized CP (BinCP) method has several advantages. First, we define robust CP that only computes a single certificate. In comparison, CAS requires at least one certificate per calibration (or test) point. Second, our method can effortlessly use many existing binary certificates out of the box without any additional assumptions or modifications. A direct consequence is that we can use de-randomization techniques (Levine and Feizi, 2021) that completely nullify the need for sample correction under ℓ_1 norm. Third, when we do need sample correction, working with binary variables allows us to use tighter concentration inequalities (Clopper and Pearson, 1934) (see subsubsection D.4.2 for a detailed discussion). Thus, even with significantly lower MC samples, our method still produces small prediction sets (see Fig. 6.1-left). This improvement is even more pronounced for datasets with a large number of classes (e.g. ImageNet shown in Fig. 6.5). Finally, BinCP does not require the score function to be bounded which is a limitation in current methods. Our code is available on the BinCP Github repository.

6.2 Background

We assume a holdout set of labeled calibration datapoints $\mathcal{D}_{\text{cal}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ which is exchangeable with future test points $(\boldsymbol{x}_{n+1}, y_{n+1})$, both sampled from some distribution $\mathcal{D}$. We have black-box access to a model from which we compute an

arbitrary conformity[13] score $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, e.g. score $s(\boldsymbol{x}, y) = \pi_y(\boldsymbol{x})$ where $\pi_y(\boldsymbol{x})$ is the predicted probability for class y (other scores in § D.1).

Vanilla CP. For a user-specified nominal coverage $1 - \alpha$, let

$$q_\alpha = \mathbb{Q}\left(\alpha; \{s(\boldsymbol{x}_i, y_i)\}_{i=1}^n \cup \{\infty\}\right)$$

where $\mathbb{Q}(\cdot;\cdot)$ is the quantile function. The sets defined as $C(\boldsymbol{x}_{n+1}) = \{y : s(\boldsymbol{x}_{n+1}, y) \geq q_\alpha\}$ have $1 - \alpha$ guarantee to include the true label y_{n+1}. Formally, $\Pr[y_{n+1} \in C(\boldsymbol{x}_{n+1})] \geq 1 - \alpha$ (Vovk et al., 2005) where the probability is over $\mathcal{D}_{\text{cal}} \sim \mathcal{D}, \boldsymbol{x}_{n+1} \sim \mathcal{D}$. This guarantee, and later our robust sets, are independent of the mechanics of the model and the score function – the model's accuracy or the quality of the score function is irrelevant. A score function that better reflects input-label agreement leads to more efficient (i.e., smaller) prediction sets. For noisy or adversarial inputs, the exchangeability between the test and calibration set breaks, making the coverage guarantee invalid. Fig. 6.1-middle, and Fig. 6.6-right show that an adversary (or bounded worst-case noise) can decrease the empirical coverage drastically with imperceptible perturbations on each test point. As a defense, *robust* CP extends this guarantee to the worst-case bounded perturbations.

Threat model. The adversary's goal is to decrease the empirical coverage probability by perturbing the input. Let $\mathcal{B} : \mathcal{X} \to 2^{\mathcal{X}}$ be a ball that returns all admissible perturbed points around an input. For images a common threat model is defined by the ℓ_2 norm: $\mathcal{B}_r(\boldsymbol{x}) = \{\tilde{\boldsymbol{x}} : \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq r\}$ where the radius r controls the perturbation magnitude. Similarly, we can use the ℓ_1 norm. For binary data and graphs, Bojchevski et al. (2020) define $\mathcal{B}_{r_a, r_d}(\boldsymbol{x}) = \{\tilde{\boldsymbol{x}} : \sum_{i=1}^d \mathbb{I}[\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - 1] \leq r_d, \sum_{i=1}^d \mathbb{I}[\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + 1] \leq r_a\}$ where the adversary is allowed to toggle at most r_a zero bits, and r_d one bits.

Inverted ball $\mathcal{B}^{-1}$. At test time we are given a (potentially) perturbed $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$. However, to obtain robust sets, we need to reason about (the score) of the unseen clean $\boldsymbol{x}$. Naively, one might assume that $\boldsymbol{x} \in \mathcal{B}(\tilde{\boldsymbol{x}})$ – the clean point is in the ball around the perturbed point. However, this only holds in special cases such as the ball defined by the l_2 norm. For example, if a binary $\tilde{\boldsymbol{x}}$ was obtained by removing r_d bits and adding r_a bits, to able to reach the clean $\boldsymbol{x}$ from the perturbed $\tilde{\boldsymbol{x}}$ we need to add r_d bits and remove r_a bits instead since $\mathcal{B}_{r_a, r_d}$ unlike $\mathcal{B}_r$ is not symmetric. We define the inverted ball $\mathcal{B}^{-1}$ as the smallest ball centered at $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$ that includes the clean $\boldsymbol{x}$. Formally, $\mathcal{B}^{-1}$ should satisfy $\forall \tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x}) \Rightarrow \boldsymbol{x} \in \mathcal{B}^{-1}(\tilde{\boldsymbol{x}})$. For symmetric balls like ℓ_p-norms, $\mathcal{B}^{-1} = \mathcal{B}$. For the binary ball $\mathcal{B}^{-1}_{r_a, r_d} = \mathcal{B}_{r_d, r_a}$ we need to swap r_a and r_d to ensure this condition. Zargarbashi et al. (2024) also discuss this subtle but important aspect without formally defining $\mathcal{B}^{-1}$.

Robust CP. Given a threat model, robust CP defines a *conservative* prediction set $\bar{C}$ that maintains the conformal guarantee even for worst-case inputs. Formally,

$$\Pr_{\mathcal{D}_{\text{cal}} \cup \{\boldsymbol{x}_{n+1}\} \sim \mathcal{D}} [y_{n+1} \in \bar{C}(\tilde{\boldsymbol{x}}_{n+1}), \forall \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})] \geq 1 - \alpha \tag{56}$$

The intuition behind existing methods is as follows: (i) Vanilla CP covers $\boldsymbol{x}_{n+1}$ with $1 - \alpha$ probability (ii) if $y \in C(\boldsymbol{x}_{n+1})$ then $y \in \bar{C}(\tilde{\boldsymbol{x}}_{n+1})$. Thus, robust CP covers $\tilde{\boldsymbol{x}}_{n+1}$ with at least the same probability. Here, (ii) is guaranteed via certified lower bounds $\text{c}^{\downarrow}[s, \boldsymbol{x}, \mathcal{B}]$ or certified upper bounds $\text{c}^{\uparrow}[s, \boldsymbol{x}, \mathcal{B}^{-1}]$.

Theorem 6.1 (Robust CP from Zargarbashi et al. (2024)). *Define $s_y(\cdot) = s(\cdot, y)$. With $\text{c}^{\uparrow}[s_y, \tilde{\boldsymbol{x}}, \mathcal{B}^{-1}] \geq \max_{\boldsymbol{x}' \in \mathcal{B}^{-1}(\tilde{\boldsymbol{x}})} s(\boldsymbol{x}', y)$, let $\bar{C}_{\text{test}}(\tilde{\boldsymbol{x}}_{n+1}) = \left\{y : \text{c}^{\uparrow}[s_y, \tilde{\boldsymbol{x}}_{n+1}, \mathcal{B}^{-1}] \geq q\right\}$, then $\bar{C}_{\text{test}}$ satisfies Eq. 56 (test-time robustness). Alternatively, with $\text{c}^{\downarrow}[s_y, \boldsymbol{x}, \mathcal{B}] \leq \min_{\boldsymbol{x}' \in \mathcal{B}(\boldsymbol{x})} s(\boldsymbol{x}', y)$,*

[13] Conformity scores quantify agreement and are equivalent up to a sign flip to non-conformity scores.

define $q^{\downarrow} = \mathbb{Q}\left(\alpha; \left\{\mathrm{c}^{\downarrow}[s_{y_i}, \boldsymbol{x}_i, \mathcal{B}]\right\}_{i=1}^{n}\right)$. *Then* $\bar{C}_{\text{cal}}(\tilde{\boldsymbol{x}}_{n+1}) = \left\{y : s(\tilde{\boldsymbol{x}}_{n+1}, y) \geq q^{\downarrow}\right\}$ *also satisfies Eq. 56 (calibration-time robustness).*

In Theorem 6.1 test-time robustness uses $\mathcal{B}^{-1}$ since it queries the clean point from the perspective of the perturbed test input. Alternatively, calibration-time robustness uses $\mathcal{B}$ since the clean calibration point is given and we are finding the lower bound for the unseen test point in the test. The intuition is that the lower bound scores from the clean calibration points are exchangeable with the lower bound of the clean test input. The perturbed test input will surely have a higher score compared to this lower bound, hence it would be covered with higher probability.

We can obtain the $\mathrm{c}^{\downarrow}, \mathrm{c}^{\uparrow}$ bounds through neural network verifiers Jeary et al. (2024) or randomized smoothing (Cohen et al., 2019). We focus on the latter since we get model-agnostic certificates with black-box access. The coverage probability is theoretically proved in CP. Similarly, (adversarially) robust CP also comes with a theoretical guarantee. In both cases we can compute the empirical coverage as a sanity check. Another metric of interest in both cases is the average set size (the efficiency) of the conformal sets.

Randomized smoothing. A smoothing scheme $\xi : \mathcal{X} \to \mathcal{X}$ maps any point to a random nearby point. For continuous data Gaussian smoothing $\xi(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{\epsilon}$ adds an isotropic Gaussian noise to the input $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma \boldsymbol{I})$. For sparse binary data Bojchevski et al. (2020) define sparse smoothing as $\xi(\boldsymbol{x}) = \boldsymbol{x} \oplus \boldsymbol{\epsilon}$ where $\oplus$ is the binary XOR, and $\boldsymbol{\epsilon}[i] \sim \text{Bernoulli}(p = p_{\boldsymbol{x}[i]})$, where p_1, and p_0 are two smoothing parameters to account for sparsity. To simplify the notation we write $\boldsymbol{x} + \boldsymbol{\epsilon}$ instead of $\xi(\boldsymbol{x})$ in the rest of the paper for both Gaussian and sparse smoothing, but our method works for any smoothing scheme beyond additive noise. Regardless of how rapidly a score function $s(\boldsymbol{x}, y)$ changes, the smooth score $\bar{s}(\boldsymbol{x}, y) = \mathbb{E}_{\boldsymbol{\epsilon}}[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y)]$ changes slowly near $\boldsymbol{x}$. This enables us to compute tight $\mathrm{c}^{\downarrow}, \mathrm{c}^{\uparrow}$ bounds that depend on the smoothing strength. See § 6.4, § D.2, and subsubsection D.4.1 for details.

6.3 Binarized Conformal Prediction (BinCP)

We define conformal sets by binarizing randomized scores. We first show that this preserves the conformal guarantee for clean data. Then in § 6.4 we extends the guarantee to worst-case adversarial inputs. As we will see in § 6.6 our binarization approach has gains in terms of Monte-Carlo sampling budget, computational cost, and average set size.

Proposition 6.2. *For any two parameters* $p \in (0, 1), \tau \in \mathbb{R}$, *given a smoothing scheme* $\boldsymbol{x} + \boldsymbol{\epsilon}$,

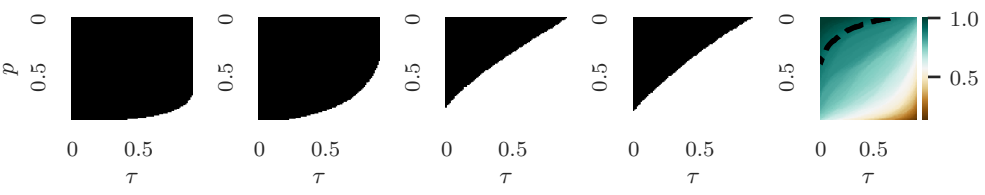

Figure 6.2: [Left] Function accept$(\boldsymbol{x}_i, y_i; p, \tau)$ for different (p, τ) pairs for four random CIFAR-10 instances. Black equals 1 and white equals 0. [Right] Empirical coverage for different (p, τ) pairs. Any (p, τ) pair on the dashed black line (the 0.9 contour) gives conformal sets with 90% coverage.

define the boolean function $\text{accept}[\cdot, \cdot; p, \tau]$ *and the prediction set* $C(\cdot; p, \tau)$ *as*

$$\text{accept}[\boldsymbol{x}, y; p, \tau] = \mathbb{I}[\Pr_{\boldsymbol{\epsilon}}[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y) \geq \tau] \geq p] \quad \textit{and} \quad C(\boldsymbol{x}; p, \tau) = \left\{y : \text{accept}(\boldsymbol{x}, y; p, \tau)\right\}$$

For any fixed p*, let*

$$\tau_\alpha(p) = \sup_\tau \left\{ \tau : \sum_{i=1}^{n} \text{accept}(\boldsymbol{x}_i, y_i; p, \tau) \geq (1-\alpha) \cdot (n+1) \right\} \tag{57}$$

then the set $C(\boldsymbol{x}_{n+1}; p, \tau_\alpha(p))$ *has* $1-\alpha$ *coverage guarantee. Alternatively, for any fixed* τ*, let*

$$p_\alpha(\tau) = \sup_p \left\{ p : \sum_{i=1}^{n} \text{accept}(\boldsymbol{x}_i, y_i; p, \tau) \geq (1-\alpha) \cdot (n+1) \right\} \tag{58}$$

again the prediction set $C(\boldsymbol{x}_{n+1}; p_\alpha(\tau), \tau)$ *has* $1-\alpha$ *coverage guarantee.*

The correctness of Proposition 6.2 can be directly seen by noticing that we implicitly define new scores.

Quantile view. Let $S_i = s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i)$ be the distribution of randomized scores for $\boldsymbol{x}_i$ and the true class y_i. Let $\tau_i(p) = \mathbb{Q}(p; S_i)$, we have that $\tau_\alpha(p) = \mathbb{Q}\left(\alpha; \{\tau_i(p)\}_{i=1}^n\right)$ is a quantile of quantiles. Similarly, define $p_i(\tau) = \mathbb{Q}^{-1}(\tau; S_i)$ then $p_\alpha(\tau) = \mathbb{Q}\left(\alpha; \{p_i(\tau)\}_{i=1}^n\right)$ is a quantile of inverse quantiles. Both $\tau_i(p)$ for a fixed p and $p_i(\tau)$ for a fixed τ are valid conformity scores for the instance $\boldsymbol{x}_i$, since exchangeability is trivially preserved. Therefore, $\tau_\alpha(p)$ and $p_\alpha(\tau)$ are just the standard quantile thresholds from CP on some new score functions. This directly gives the $1-\alpha$ coverage guarantee. This view via the implicit scores is helpful for intuition, but we keep the original formulation since it is more directly amenable to certification as we show in § 6.4. We provide an additional formal proof of Proposition 6.2 via conformal risk control (Angelopoulos et al., 2022) in § D.3.

Using either variant from Proposition 6.2 let (p_α, τ_α) equal $(p, \tau_\alpha(p))$ or $(p_\alpha(\tau), \tau)$ as the final pair of parameters. For test points $\boldsymbol{x}_{n+1}$ we accept labels whose smooth score distribution has at least p_α proportion above the threshold τ_α, i.e. $\text{accept}(\boldsymbol{x}_{n+1}, y; p_\alpha, \tau_\alpha) = 1$. The term "binarization" refers to mapping each score sample above τ to 1 and all others 0. For distributions with a strictly increasing and continuous CDF (e.g. isotropic Gaussian smoothing) both variants are equivalent.

Lemma 6.3. *Given distributions* $\{S_i\}_{i=1}^n$ *with strictly increasing and continuous CDFs, let* $\tau_\alpha(p)$ *be obtained from Eq. 57 with fixed* p *and* $p_\alpha(\cdot)$ *be as defined in Eq. 58. We have* $p_\alpha(\tau_\alpha(p)) = p$*.*

We defer all proofs to § D.3. For fixed p, Proposition 6.2 yields $(p, \tau_\alpha(p))$. Fixing $\tau = \tau_\alpha(p)$ we get sets with $(p_\alpha(\tau), \tau) = (p_\alpha(\tau_\alpha(p)), \tau_\alpha(p))$ which also equals $(p, \tau_\alpha(p))$ from Lemma 6.3. Fig. 6.2 shows the $\text{accept}(\boldsymbol{x}, y; p, \tau)$ function for several examples. This function is non-increasing in both parameters p and τ. In general, any arbitrary assignment of p, and τ, results in some expected coverage – $\text{accept}(\cdot, \cdot, p, \tau)$ equals to 1 for some number of $(\boldsymbol{x}_i, y_i)$s (Fig. 6.2-right). Pairs (p_α, τ_α) obtained from Proposition 6.2 are placed on the $1-\alpha$ contour of this expectation. The empirical coverage is close to this expectation due to exchangeability (Berti and Rigo, 1997).

Remarks. The scores $\tau_i(p)$ (and similarly $p_i(\tau)$ remain exchangeable whether the quantile over the smoothing distribution is computed exactly or estimated from any number of Monte-Carlo samples. That is, Proposition 6.2 holds regardless. However,

when need to be more careful when we consider the certified upper and lower bounds. In § 6.4 we first derive robust conservative sets that maintain worst-case coverage, assuming that we can compute probabilities and expectations exactly. Since this is not always possible, in § 6.5 we provide the appropriate sample correction that still preserves the robustness guarantee when using Monte-Carlo samples. We also discuss a de-randomized approach that does not need sample correction.

6.4 Robust BinCP

From Proposition 6.2 (either variant) we compute a pair (p_α, τ_α). Following Proposition 6.2, for clean $\boldsymbol{x}_{n+1}$, we have $\Pr[s(\boldsymbol{x}_{n+1} + \boldsymbol{\epsilon}, y_{n+1}) \geq \tau_\alpha] \geq p_\alpha$ with probability $1 - \alpha$. We will exploit this property. Define $f_y(\boldsymbol{x}) = \mathbb{I}[s(\boldsymbol{x}, y) \geq \tau_\alpha]$, we have $\bar{f}_y(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\epsilon}}[\mathbb{I}[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y) \geq \tau_\alpha]] = \Pr_{\boldsymbol{\epsilon}}[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y) \geq \tau_\alpha]$.

Conventional robust CP. One way to attain robust prediction sets is to apply the same recipe as Zargarbashi et al. (2024) (CAS) by finding upper or lower bounds on the new score function. CAS uses the smooth score $\bar{s}_y(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\epsilon}}[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y)]$. Instead, we can bound $\bar{f}_y(\boldsymbol{x})$ which is a smooth binary classifier. Note that as discussed in § 6.3 (quantile of inverse quantiles), $\bar{f}_y(\boldsymbol{x})$ is a conformity score function itself. Therefore, following Theorem 6.1, the test-time, and calibration-time robust prediction sets are

$$\bar{C}_{\text{test}}(\tilde{\boldsymbol{x}}_{n+1}) = \{y : \mathrm{c}^{\uparrow}[\bar{f}_y, \tilde{\boldsymbol{x}}_{n+1}, \mathcal{B}^{-1}] \geq p_\alpha\}, \quad \bar{C}_{\text{cal}}(\tilde{\boldsymbol{x}}_{n+1}) = \{y : \bar{f}_y(\tilde{\boldsymbol{x}}_{n+1}) \geq q^{\downarrow}\} \tag{59}$$

where $q^{\downarrow} = \mathbb{Q}\left(\alpha; \{\mathrm{c}^{\downarrow}[\bar{f}_{y_i}, \boldsymbol{x}_i, \mathcal{B}]\}_{i=1}^{n}\right)$. In short, we replace the clean $\bar{f}_{y_{n+1}}(\boldsymbol{x}_{n+1})$ with either its certified upper $\mathrm{c}^{\uparrow}$ or lower $\mathrm{c}^{\downarrow}$ bound. We elaborate on this approach before improving it.

Computing $\mathrm{c}^{\downarrow}$ and $\mathrm{c}^{\uparrow}$. Computing exact worst-case bounds on $\bar{f}$ ($\bar{f}_y$ for all y) is intractable and requires white-box access to the score function and therefore the model. Following established techniques in the randomized smoothing literature (Lee et al., 2019) we relax the problem. Formally,

$$\mathrm{c}^{\downarrow}[\bar{f}, \boldsymbol{x}, \mathcal{B}] = \min_{\substack{\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x}) \\ h \in \mathcal{H}}} \Pr_{\boldsymbol{\epsilon}}[h(\tilde{\boldsymbol{x}} + \boldsymbol{\epsilon})] \quad \text{s.t.} \quad \Pr_{\boldsymbol{\epsilon}}[h(\boldsymbol{x} + \boldsymbol{\epsilon})] = \Pr_{\boldsymbol{\epsilon}}[f(\boldsymbol{x} + \boldsymbol{\epsilon})] = \bar{f}(\boldsymbol{x}) \tag{60}$$

where $\mathcal{H}$ is the set of all measurable functions h. Since $f \in \mathcal{H}$ we have $\mathrm{c}^{\downarrow}[\bar{f}, \boldsymbol{x}, \mathcal{B}] \leq \bar{f}(\tilde{\boldsymbol{x}})$ for all $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$. The upper bound $\mathrm{c}^{\uparrow}[\bar{f}, \boldsymbol{x}, \mathcal{B}^{-1}]$ is the solution to a similar *maximization* problem.

Closed form. For ℓ_2 ball with Gaussian smoothing, Eq. 60 has a closed form solution $\Phi_\sigma(\Phi_\sigma^{-1}(\bar{f}_y(\boldsymbol{x})) - r)$ where Φ_σ is the CDF of the normal distribution $\mathcal{N}(\mathbf{0}, \sigma \boldsymbol{I})$(Cohen et al., 2019; Kumar et al., 2020). The upper bound is similarly computed by changing the sign of r. Yang et al. (2020) show the same closed-form applies solution for the ℓ_1 ball, and additionally, discuss other perturbation balls and smoothing schemes most of which are applicable. For sparse smoothing we can compute the bounds with a simple algorithm with $O(r_a + r_d)$ runtime (Bojchevski et al., 2020), which we discuss in § D.3. For ℓ_1 ball and uniform smoothing the lower bound equals $\bar{f}_y(\boldsymbol{x}) - 1/(2\lambda)$ where $\boldsymbol{\epsilon} \sim \mathcal{U}[0, 2\lambda]^d$ (Levine and Feizi, 2021). This bound can also be de-randomized (see § 6.5).

Single Binary Certificate. From the closed-form solutions we see that the bounds are independent of the definition of f, and the test point $\boldsymbol{x}$; i.e. their output is a function of the scalar $p := \bar{f}_y(\boldsymbol{x})$. We defer the discussion for why this holds to § D.2, and subsubsection D.4.1; in short the solution for any $\boldsymbol{x}$ can be obtained from alternative canonical points $\boldsymbol{u}$, and $\tilde{\boldsymbol{u}}$. Therefore, we write $\mathrm{c}^{\downarrow}[p, \mathcal{B}] = \mathrm{c}^{\downarrow}[\bar{f}_y, \boldsymbol{x}, \mathcal{B}]$ to show that $\mathrm{c}^{\downarrow}$

depends only on p and $\mathcal{B}$, and the same for $\mathrm{c}^{\uparrow}$. We also notice that in common smoothing schemes and perturbation balls, it holds that $\mathrm{c}^{\downarrow}[\mathrm{c}^{\uparrow}[p, \mathcal{B}^{-1}], \mathcal{B}] = p$ which allows us to reduce both calibration-time and test-time robustness to solving a single binary certificate. We formalize this in Lemma 6.4.

Lemma 6.4. *If* $\mathrm{c}^{\downarrow}[\mathrm{c}^{\uparrow}[p, \mathcal{B}^{-1}], \mathcal{B}] = p$ *for all* p, *then* $\bar{C}_{\text{test}}(\tilde{\boldsymbol{x}}_{n+1}) = \bar{C}_{\text{cal}}(\tilde{\boldsymbol{x}}_{n+1}) = \bar{C}_{\text{bin}}(\tilde{\boldsymbol{x}}_{n+1})$ *where* $\bar{C}_{\text{bin}}(\tilde{\boldsymbol{x}}_{n+1}) = \{y : \text{accept}(\tilde{\boldsymbol{x}}_{n+1}, y; \mathrm{c}^{\downarrow}[p_\alpha, \mathcal{B}], \tau_\alpha)\} = \{y : \Pr_{\boldsymbol{\epsilon}}[s(\boldsymbol{x}_{n+1} + \boldsymbol{\epsilon}, y_{n+1}) \geq \tau_\alpha] \geq \mathrm{c}^{\downarrow}[p_\alpha, \mathcal{B}]\}$.

To see why, let $\tilde{p}_{n+1} = \bar{f}_{y_{n+1}}(\tilde{\boldsymbol{x}}_{n+1})$. The test-time robust coverage requires $\mathrm{c}^{\uparrow}[\tilde{p}_{n+1}, \mathcal{B}^{-1}] \geq p_\alpha$. Since both $\mathrm{c}^{\downarrow}$, and $\mathrm{c}^{\uparrow}$ are non-decreasing w.r.t. p, we have $\mathrm{c}^{\downarrow}[\mathrm{c}^{\uparrow}[\tilde{p}_{n+1}, \mathcal{B}^{-1}], \mathcal{B}] \geq \mathrm{c}^{\downarrow}[p_\alpha, \mathcal{B}]$. We have the equivalent condition $\tilde{p}_{n+1} \geq \mathrm{c}^{\downarrow}[p_\alpha, \mathcal{B}]$. This implies that we only need to compute a single certificate $\mathrm{c}^{\downarrow}[p_\alpha, \mathcal{B}]$ once with the single p_α value given by Proposition 6.2. This also allows us to seamlessly integrate other existing binary certificates in a plug and play manner. In contrast, with Theorem 6.1 for $\bar{C}_{\text{test}}$ or $\bar{C}_{\text{cal}}$ we need at least one certificate per test (or calibration) point. Notably, these prediction set are identical to our cheaper $\bar{C}_{\text{bin}}$. For illustration, Fig. 6.3 shows the certified lower and upper bounds for various p_α values and various smoothing schemes.

Intuitively $\mathrm{c}^{\downarrow}[\mathrm{c}^{\uparrow}[p, \mathcal{B}^{-1}], \mathcal{B}] = p$ holds due to symmetry of the smoothing scheme w.r.t. $\mathcal{B}$, and $\mathcal{B}^{-1}$ and is satisfied by most smoothing schemes. In Lemma 6.5 we prove that Gaussian, uniform, and sparse smoothing all have this property.

Lemma 6.5. *For Gaussian, and uniform smoothing under* ℓ_1, *and* ℓ_2 *balls* $\mathcal{B}_r = \mathcal{B}_r^{-1}$. *For sparse smoothing and* $\mathcal{B}_{r_a, r_d}$ *we have* $\mathcal{B}_{r_a, r_d}^{-1} = \mathcal{B}_{r_d, r_a}$. *In all three cases we have* $\mathrm{c}^{\downarrow}[\mathrm{c}^{\uparrow}[p, \mathcal{B}^{-1}], \mathcal{B}] = p$.

To summarize, for robust BinCP, we first compute conformal thresholds (p_α, τ_α) from Proposition 6.2. Then for a perturbation ball $\mathcal{B}$ that satisfies $\mathrm{c}^{\downarrow}[\mathrm{c}^{\uparrow}[p, \mathcal{B}^{-1}], \mathcal{B}] = p$, we compute $\mathrm{c}^{\downarrow}[p_\alpha, \mathcal{B}]$ and compute the prediction sets with $(\mathrm{c}^{\downarrow}[p_\alpha, \mathcal{B}], \tau_\alpha)$ instead. The resulting sets have $1 - \alpha$ robust coverage.

Corollary 6.6. *With* (p_α, τ_α) *from Proposition 6.2 on a calibration set* $\mathcal{D}_{\text{cal}}$, *let* $\boldsymbol{x}_{n+1}$ *be exchangeable with* $\mathcal{D}_{\text{cal}}$ *and* $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$. *If for the smoothing scheme* ξ *and the threat model* $\mathcal{B}$ *and for all* p *we have* $\mathrm{c}^{\downarrow}[\mathrm{c}^{\uparrow}[p, \mathcal{B}^{-1}], \mathcal{B}] = p$, *then the set* $\bar{C}_{\text{bin}}(\tilde{\boldsymbol{x}}_{n+1}) = \{y : \Pr[s(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, y_{n+1}) \geq \tau_\alpha] \geq \mathrm{c}^{\downarrow}[p_\alpha, \mathcal{B}]\}$ *has* $1 - \alpha$ *coverage (the pseudocode is in § D.1).*

6.5 Robust BinCP with Finite Samples

The certificate in Corollary 6.6 relies on exact probabilities $\Pr[s(\boldsymbol{x}_{n+1} + \boldsymbol{\epsilon}, y_{n+1}) \geq \tau_\alpha]$ which is often intractable to compute. Instead, we can either apply de-randomization techniques or estimate high-confidence bounds of these probabilities. We first describe the latter approach. For each calibration point $(\boldsymbol{x}_i, y_i)$ we compute $q_i = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i) \geq \tau_\alpha]$ where m is the number of Monte-Carlo (MC) samples. For each label of the (potentially perturbed) test point we compute $\tilde{q}_{n+1,y} = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}[s(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, y) \geq \tau_\alpha]$. We use the Clopper-Pearson confidence interval (Clopper and Pearson, 1934) to bound the exact probabilities via the MC estimates. To ensure the sets are conservative we compute a lower bound for calibration points and an upper bound for test points. Collectively, all bounds are valid with adjustable $1 - \eta$ probability. To account for this, we set the nominal coverage level to $1 - \alpha + \eta$ such that we have $1 - \alpha$ coverage in total. Similar to Zargarbashi et al. (2024), we compute each bound with $1 - \eta/(|\mathcal{D}_{\text{cal}} + k|)$ probability where k is the number of classes. Let $p_i = \Pr[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i) \geq \tau_\alpha]$ for $i \in \{1, \ldots, n+1\}$ be the exact probabilities. The final sample-corrected robust predictions sets are given in Proposition 6.7.

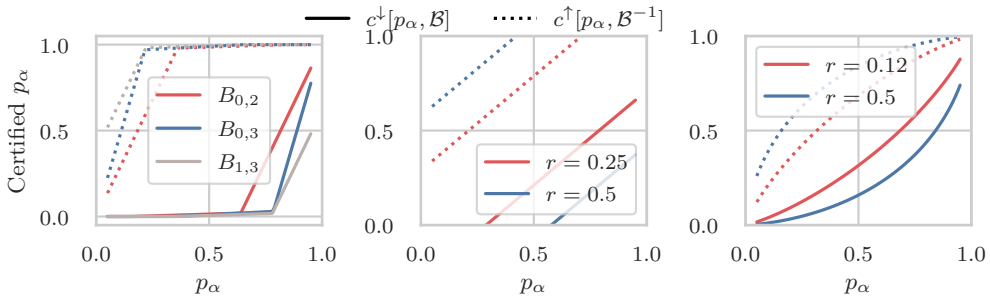

Figure 6.3: [From left to right] Certified bounds for sparse smoothing, ℓ_1 ball with de-randomized uniform smoothing (Levine and Feizi, 2021), and ℓ_2 (same as ℓ_1) ball with Gaussian smoothing.

Proposition 6.7. *Let $q_i^{\downarrow} \leq p_i$ hold with $1-\eta/(\mathcal{D}_{\text{cal}}+k)$ for each calibration point $i \in \{1, \ldots, n\}$ where k is the number of target classes. For a given test point $\tilde{\boldsymbol{x}}_{n+1}$ let $\tilde{q}^{\uparrow}_{n+1,y} \geq \tilde{p}_{n+1,y}$ with $1-\eta/(\mathcal{D}_{\text{cal}}+k)$ where $\tilde{p}_{n+1,y} = \Pr[s(\tilde{\boldsymbol{x}}_{n+1}+\boldsymbol{\epsilon}, y) \geq \tau_\alpha]$. With $p_\alpha^{\downarrow} = \mathbb{Q}\left(\alpha-\eta; \{q_i^{\downarrow}\}_{i=1}^n\right)$, we set the robust conformal threshold pair as $(\mathrm{c}^{\downarrow}[p_\alpha^{\downarrow}, \mathcal{B}], \tau_\alpha)$. Then the prediction set defined as $\bar{C}_+(\tilde{\boldsymbol{x}}_{n+1}; \mathrm{c}^{\downarrow}[p_\alpha^{\downarrow}, \mathcal{B}], \tau_\alpha) = \{y : \tilde{q}^{\uparrow}_{n+1,y} \geq \mathrm{c}^{\downarrow}[p_\alpha^{\downarrow}, \mathcal{B}]\}$ has $1-\alpha$ coverage probability.*

Such sample correction is a crucial step for smoothing-based robust CP, since the robustness certificate is probabilistic. The failure of the certificate depends to the failure of the confidence intervals. In contrast, for deterministic and de-randomized certificates such as DSSN (Levine and Feizi, 2021), we do not need sample correction since we can exactly compute p_i and $p_\alpha = \mathbb{Q}\left(\alpha; \{p_i\}_{i=1}^n\right)$. Note, vanilla (non-robust) BinCP does not need sample correction to maintain the guarantee (see § 6.3).

6.6 Experiments

We show that: (i) We can return guaranteed and small sets for both image classification and node classification, with a significantly lower number of Monte Carlo samples. (ii) Our sets are computationally efficient. (iii) There is an inherent robustness in randomized methods. (iv) We can also use de-randomized smoothing-based certificates that do not require finite sample correction.

Setup. We evaluate our method on two image datasets: CIFAR-10 (Krizhevsky, 2009) and ImageNet (Deng et al., 2009), and for node-classification (graph) dataset we use Cora-ML McCallum et al. (2004). For the CIFAR-10 dataset we use `ResNet-110` and for the ImageNet dataset we use ResNet-50 pretrained models with noisy data augmentation from Cohen et al. (2019). For the graph classification task we similarly train a GCN model Kipf and Welling (2017) on CoraML with noise augmentation. The GCN 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 250 (sparsely labeled setting) unless specified explicitly. Our reported results on conformal prediction performance are averaged over 100 runs with different calibration set samples. We calibrated BinCP with a $p = 0.6$ fixed value, however small changes in p do not influence the result. For the graph dataset we calibrated BinCP with $p = 0.9$. Intuitively as $\mathrm{c}^{\downarrow}[p, \mathcal{B}]$ has a sharp decay for sparse smoothing (see Fig. D.3-left), we set p to a number such that $\mathrm{c}^{\downarrow}[p, \mathcal{B}]$ still remains high.

We conducted our experiment using three different smoothing schemes. (i) Smooth-

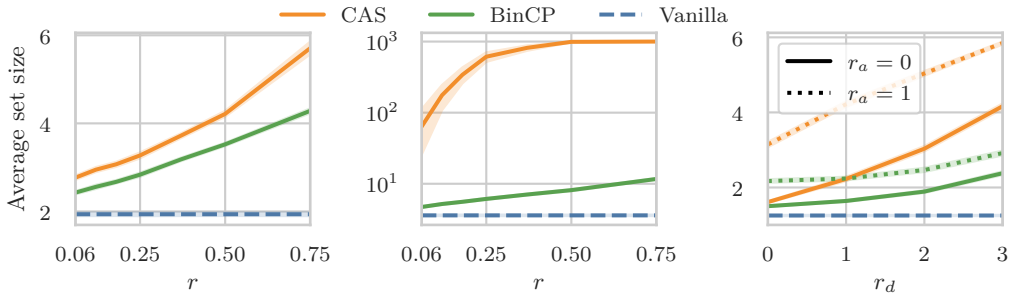

Figure 6.4: [Left to right] Average prediction set size of robust CP for CIFAR-10, and ImageNet with Gaussian smoothing ($\sigma = 0.5$), and CoraML with sparse smoothing. All results are for 2000 Monte-Carlo samples. We set $1 - \alpha = 0.85$ for ImageNet, and $1 - \alpha = 0.9$ for CIFAR-10 and CoraML.

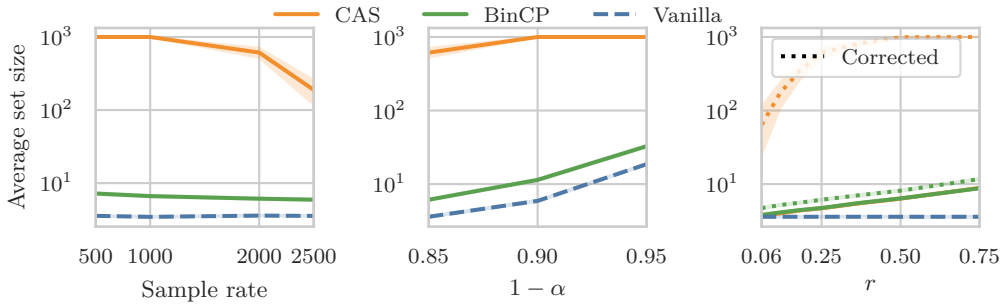

Figure 6.5: On ImageNet dataset, [left] average set size for $1 - \alpha = 0.85$ with various MC sampling budgets. [Middle] Set size across various levels of $1 - \alpha$ for 2×10^3 samples. [Right] Set size without sample correction (asymptotically valid assumption). The sample-corrected variants are shown with a dotted line. In all plots y-axis is log-scaled.

ing with isotropic Gaussian noise, $\sigma = 0.12, 0.25$, and 0.15. Our reported results for BinCP are valid for both ℓ_1, and ℓ_2 perturbation balls. (ii) De-randomized smoothing with splitting noise (DSSN) from Levine and Feizi (2021) from which we attain ℓ_1 robustness. We examine two smoothing levels $\lambda = 0.25/\sqrt{3}$, and $0.5/\sqrt{3}$. (iii) Sparse smoothing from Bojchevski et al. (2020) with $p_+ = 0.01$, and $p_- = 0.6$ on node attributes. We report robustness across $r_a \in \{0, 1\}$, and $r_d \in \{0, 1, 2, 3\}$. We compare our the result from BinCP to the SOTA method CAS (Zargarbashi et al., 2024). Previously it was shown that CAS significantly outperforms RSCP (Gendler et al., 2021) both with and without finite sample correction. In § 6.7 we discuss the other related works in detail. In the standard setup, we estimate the statistics (mean and CDF, or Bernoulli parameters) with 2×10^3 Monte-Carlo samples, and we set $1 - \alpha = 0.9$. This setup is picked in favor of the baseline since by increasing the nominal coverage or decreasing the sample size BinCP outperforms the baseline with an even higher margin. Throughout the paper we report different nominal coverages and MC sampling budgets.

Smaller set size. Fig. 6.4 shows that for all datasets, and both smoothing schemes (isotropic Gaussian and sparse smoothing), BinCP produces smaller prediction sets compared to CAS. Our prediction sets computed with Gaussian noise are robust to both ℓ_1, and ℓ_2 perturbation balls with the same radius (Yang et al., 2020). In Fig. 6.5, compare to CAS on ImageNet dataset over various sample rates, coverages, and radii. Since CAS with sample correction returns trivial sets for $1 - \alpha = 0.9$ (see Fig. 6.5-middle), in Fig. 6.5-left we compare the results for $1 - \alpha = 0.85$. By increasing the Monte Carlo

sample rate, the average set sizes of CAS and BinCP become closer – Fig. 6.1-left for CIFAR-10, Fig. 6.5-left for ImageNet, and Fig. 6.7-left for CoraML depict the impact of higher sampling budget. Intuitively as the number of classes increase (e.g. ImageNet dataset) the union bound over classes result in larger prediction sets (looser confidence intervals). In § D.5 (Fig. D.2) we show that BinCP is also consistently better for smaller radii (a setup similar to verifier based robust CP (Jeary et al., 2024)). For the same reported set size, we attain robust CP of significantly larger radii.

Note that Fig. 6.4-right, and Fig. 6.7-left show the performance on the *transductive* node classification setting where perfect robustness is achievable for free through memorization (see discussion by Gosch et al. (2024)). Nonetheless, comparing BinCP to CAS is still meaningful. Constructing robust conformal sets for GNNs in a realistic (inductive) setup is more challenging as discussed in subsubsection D.4.3.

Exact ℓ_1 robustness. Using the de-randomized DSSN certificate for ℓ_1 perturbation ball (Levine and Feizi, 2021) we derive the first smoothing-based de-randomized robust CP. As shown in Fig. 6.6-middle the de-randomized robust BinCP with uniform noise results in a significantly smaller set size across all radii compared to Gaussian noise (Fig. 6.4-left). Notably due to exactness of the computed statistics, for randomized DSSN-based certificate we bypass the finite samples correction.

Ignoring sample correction. While unrealistic in practice, Gendler et al. (2021) report results without applying finite sample correction. Zargarbashi et al. (2024) maintain small set sizes (with large MC sample rate) for CIFAR-10. However, for ImageNet and CoraML they only report the results without correction, and therefore with an "asymptomatically valid" coverage guarantee – valid when sample rate approaches infinity. In Fig. 6.5-left applying sample correction to CAS on datasets like ImageNet, inflates the prediction sets up to $\mathcal{Y}$, likely due to union bound on a large number of classes. Fig. 6.5-right shows that on ImageNet, both methods show similar prediction sets for asymptotically valid setup. Notably BinCP with sample correction is not far from the non-corrected setup, while CAS shows a large gap. Similarly Fig. 6.7-left shows that for sparse smoothing, increasing sample rate helps CAS considerably while its effect on BinCP is almost negligible. In subsubsection D.4.2 we explore the intuition behind how binarization can mitigate the impact of finite sample correction with fewer samples.

Number of samples. The upperbound in CAS is obtained through a two step process. First given the corrected CDF, we compute the worst case (adversarial) CDF. Then using

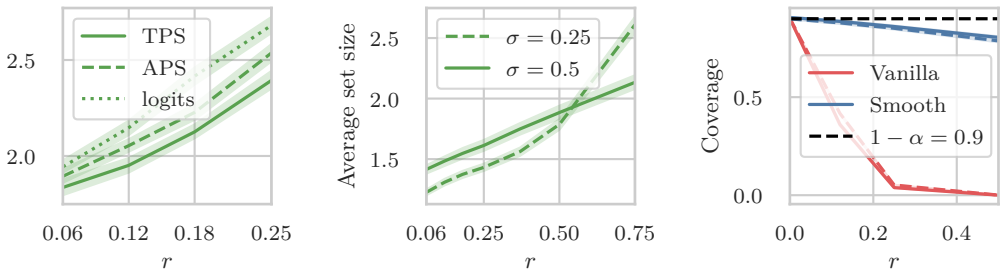

Figure 6.6: [Left] Performance of BinCP on different score functions under Gaussian smoothing with $\sigma = 0.25$. [Middle] Set size of BinCP with ℓ_1 robustness and derandomized DSSN smoothing ($\sigma = \lambda/\sqrt{3}$). [Right] Vanilla non-smooth and smooth ($\sigma = 0.25$) prediction (solid and dashed colored lines show TPS and APS score function) under attack. All results are on CIFAR-10.

upper bounded (or lower bounded) CDF, we apply the Anderson bound to obtain a bound on the mean from the CDF (Zargarbashi et al., 2024). Increasing the number of bins increases the computation slightly but produces tighter bounds. To observe this effect, without sample correction, we decrease the number of samples to a very low number ($\sim$ 10, however unrealistic) and in Fig. 6.7-middle we see that set size in CAS slightly increases even without accounting for finite sample estimation.

Effect of σ, and score function. The strength of Gaussian smoothing is controlled by σ in $\xi(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$. In Fig. 6.7-right we observe a trade-off in choosing σ for both methods. Higher smoothing intensity results in larger set sizes in the beginning, but by increasing the robustness radius the set size increases slowly. Still in all cases BinCP outperforms CAS. It is best practice to compute smooth prediction probabilities using a model trained with similar noise augmentation. We reported this result in § D.5 (Table D.2). Interestingly, when training and inference σ does not match, BinCP shows good results while CAS is more sensitive.

We mainly focus on TPS score function, in addition Fig. 6.6-left compares APS, and logit as score functions. Here across all radii TPS is more efficient. We further report results for APS in § D.5. Since logits are unbounded, CAS is not applicable to it.

Benefits of smoothing. The guarantee of robust CP breaks for adversarial (or noisy) inputs. In Fig. 6.6-right we compare vanilla prediction and smooth prediction sets under adversarial attack. Notably, smooth models even without a conservative certificate show an inherent robustness. As illustrated, the non-smooth model quickly breaks to near 0 coverage guarantee for very small r. Relatedly, recent verifier-based robust CP (Jeary et al., 2024) report comparably larger prediction sets even for one order of magnitude smaller radius (compared to the certified radii by BinCP). This intuitively suggests that for robust CP it seems that randomization is inherently beneficial.

Limitations. While BinCP reduce the required MC sample rate significantly (e.g. from 2000 to 150 on CIFAR-10), still this number of inferences is computationally intensive. The robustness in the input space $\mathcal{X}$ is not yet linked to robustness w.r.t. distribution shift. Although BinCP applies on sparse smoothing, a realistic threat model for graphs (inductive GNNs) is still not addressed.

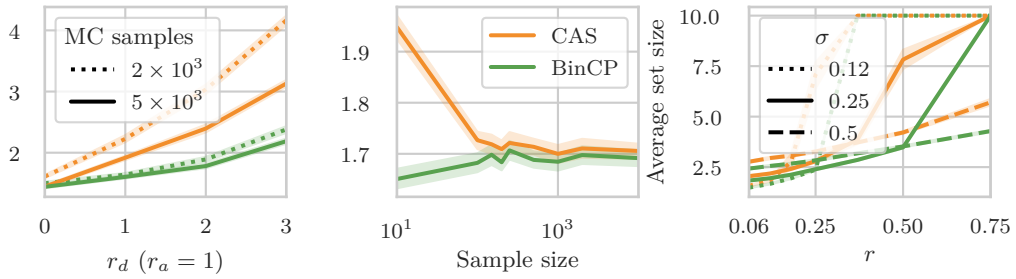

Figure 6.7: Comparison between BinCP and CAS for [Left] the effect of higher MC sample budget in CoraML dataset (sparse smoothing), [Middle] effect of low samples without finite samples correction for CIFAR-10 dataset ($\sigma = 0.25$), and [Right] various smoothing strengths σ.

6.7 Related Work

Robust CP via smoothing. Gendler et al. (2021) introduced the problem and defined a baseline robust CP method, RSCP (randomly smoothed conformal prediction), which applies Theorem 6.1 in combination with the mean-constrained upper bound for ℓ_2 perturbations and Gaussian smoothing. This upper bound has a closed form solution: $\overline{s}(\tilde{\boldsymbol{x}}, y) = \Phi(\Phi^{-1}(p) + r)$ where $p = \mathbb{E}[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y)]$. Originally, RSCP did not account for finite sample correction making its coverage guarantee only asymptomatically valid. Yan et al. (2024) show that correcting for finite samples in RSCP leads to trivial prediction sets $\bar{C}(\boldsymbol{x}) = \mathcal{Y}$. As a remedy, they define a new score function based on temperature scaling which in combination with conformal training (Stutz et al., 2021) improves the average set size. So far both methods use test-time robustness. In § D.5, we show that BinCP outperforms RSCP+.

In contrast Zargarbashi et al. (2024) utilizes the CDF structure of the score and instead apply the tighter CDF-based bound defining CDF aware sets (CAS). In combination with calibration-time robustness, they show that only $|\mathcal{D}_{\text{cal}}|$ certificate bounds should be computed to maintain a robust coverage guarantee as in Eq. 56. In addition to a gain in computational efficiency, they show that in the calibration-time robustness, the error correction budget can be used more efficiently. On CIFAR-10 they return a relatively small conformal set size. In all aforementioned methods, a large MC sampling budget (e.g. 10^4 samples) is assumed which is challenging for real-time applications. This issue is exacerbated for datasets like ImageNet where the large number of classes amplifies the effect of multiple testing corrections.

Robust CP via verifiers. Outside the scope of randomized smoothing Jeary et al. (2024) use neural network verification to compute upper (or lower) bounds. This requires white-box access to the model weights, while our proposed method works for any black-box model and randomized or exact smoothing-based certificate. Interestingly, in (Jeary et al., 2024) (Table 1) the empirical evaluation is for $r = 0.02$ which is smaller than the minimum radius we reported. For completeness, we evaluated BinCP on very small radii in § D.5 (Fig. D.2-left), and for the same r our sets are 2× smaller. As discussed in § 6.6 (Fig. 6.6) in general, smooth prediction, even without accounting for the adversary, shows to have an inherent robustness.

Other robustness results. Alternatively Ghosh et al. (2023) introduce probabilistic robust coverage which intuitively accounts for *average* adversarial inputs. This is in contrast with our core assumption of worst-case adversarial inputs. In other words, instead of $1 - \alpha$ coverage for any point within the perturbation ball around $\boldsymbol{x}_{n+1}$, "probabilistically robust coverage" guarantees that the probability to cover the true label remains above $1 - \alpha$ on average over all $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$, while we consider the worst-case $\tilde{\boldsymbol{x}}$. Their "quantile of quantiles" method looks superficially similar to BinCP as they also compute $n + k + 1$ quantiles. However there are two notable differences. Their first order of quantiles (on true calibration scores and the score for each class of the test point) is over random draws from the perturbation set. BinCP computes the first order quantiles ($\tau_i(p)$ in fixed τ setup) over the smooth score distribution. Their conservative quantile index is based on a user-specified hyperparameter that accounts for conservativeness while BinCP finds the certified probability $\mathrm{c}^{\downarrow}[p_\alpha, \mathcal{B}]$ for the worst case adversarial example. BinCP guarantees that any $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$ is covered if $\boldsymbol{x}$ is covered. Furthermore, there are other works addressing distribution or covariate shift in general beyond the score of worst-case noise robustness (Barber et al., 2022; Tibshirani et al., 2019b).

6.8 Conclusion

We introduce BinCP, a robust conformal prediction method based on randomized smoothing that produces small prediction sets with a few Monte-Carlo samples. The key insight is that we binarize the distribution of smooth scores, by a threshold (or thresholds) that maintains the coverage guarantee. We show that both calibration and test-time robustness approaches (discussed in § 6.2) are equivalent to computing a single binary certificate. This directly enables us to use any certificate that returns a certified lower-bound probability right out of the box; including de-randomized certificate for ℓ_1 norm. The binarization enables us to use tighter Clopper-Pearson confidence intervals. This leads directly to faster computation of prediction sets with a significantly lower Monte Carlo sample-rate (compared to the SOTA), and therefore less forward passes per input. Interestingly, we show that even without accounting for an adversarial setup, CP with smooth score shows more robustness to adversarial examples in comparison with the conventional vanilla CP.

7 One Sample is Enough to Make Conformal Prediction Robust

Abstract

For any black-box model, conformal prediction (CP) returns prediction *sets* guaranteed to include the true label with high adjustable probability. Robust CP (RCP) extends the guarantee to the worst case noise up to a pre-defined magnitude. For RCP, a well-established approach is to use randomized smoothing since it is applicable to any black-box model and provides smaller sets compared to deterministic methods. However, smoothing-based robustness requires many model forward passes per each input which is computationally expensive. We show that conformal prediction attains some robustness even with *a single forward pass on a randomly perturbed input*. Using any binary certificate we propose a single sample robust CP (RCP1). Our approach returns robust sets with smaller average set size compared to SOTA methods which use many (e.g. ~ 100) passes per input. Our key insight is to certify the conformal procedure itself rather than individual conformity scores. Our approach is agnostic to the task (classification and regression). We further extend our approach to smoothing-based robust conformal risk control.

7.1 Introduction

Modern neural networks return uncalibrated probability estimates (Guo et al., 2017), and other uncertainty quantification methods (like Bayesian and ensemble models, Monte-Carlo dropout) are computationally expensive. Additionally, these methods do not usually provide formal statistical guarantees. Instead, conformal prediction (CP) is a post-processing method returning prediction *sets* with a distribution-free and model-agnostic coverage guarantee, ensuring that the true answer is in the set with an adjustable high probability. To apply CP, we need a conformity[14] score function $s(\boldsymbol{x}, y)$ capturing the agreement between $\boldsymbol{x}$, and y (e.g. softmax). We compute a conformal threshold over a holdout set of calibration points, and for the test points, we form the set as all labels with scores exceeding that threshold. These sets are guaranteed to include (cover) the true label with $1 - \alpha$ probability (Angelopoulos et al., 2024).

As shown in Fig. 7.1-left (the red dashed line), this guarantee breaks by an unnoticeably small natural or adversarial noise to the test points – the empirical coverage drastically decreases by an imperceptible perturbation. Note that from this point forward, we call adversarial or natural noises as "perturbations", not to be confused with the noise we (as the defender) introduce on purpose. Robust CP (RCP) extends this guarantee to worst case bounded perturbations, ensuring that the perturbed input $\tilde{\boldsymbol{x}}$ is covered with the same or higher probability as the clean $\boldsymbol{x}$, if $\boldsymbol{x}$ is perturbed up to a known magnitude r (e.g. $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq r$). Previous RCP approaches find the highest/lowest possible conformity score within the perturbation ball, and replace the score with the worst-case bound(Gendler et al., 2021; Jeary et al., 2024; Zargarbashi et al., 2024; Zargarbashi and Bojchevski, 2025). These bounds can be computed either analytically (Lipschitz bound or verifiers) or through randomized smoothing. Here, the trade-off is between the computational cost and the guaranteed robust radius – analytical methods are robust to very smaller magnitudes of perturbation, while they only require a single forward pass per input. In contrast, while randomized smoothing needs many model forwards per single input, it provides robustness to significantly larger radii, and it applies over any black-box model.

Smoothing is to augment the input with random noise and inference from the distribution of the model's output over smooth inputs instead. For example, consider

[14]Many works define CP via a non-conformity (disagreement) score. The setups are equivalent with a change in the score's sign. Our robustness results are invariant to this definition.

adding isotropic Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ to the input $\boldsymbol{x}$, and defining the smooth score as the mean of the distribution $\mathbb{E}_{\boldsymbol{\epsilon}}[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y)]$. Regardless of the original model, the smooth model changes slowly near $\boldsymbol{x}$, as the two distribution $\boldsymbol{x} + \boldsymbol{\epsilon}$, and $\tilde{\boldsymbol{x}} + \boldsymbol{\epsilon}$ have a large overlap. This leads to model-agnostic upper bounds on the score, around $\boldsymbol{x}$. The upper bound is then used to decide whether a label is added to the prediction set. An important drawback here is the computational overhead. The score function (e.g. the expectation of the smooth scores) must be estimated via Monte-Carlo sampling. By reducing the number of samples, the confidence intervals for the mean widen, and the size of the prediction sets quickly increases.

We answer this question: *Can we design smoothing-based RCP without introducing computational overhead?* Interestingly, we show the vanilla CP combined with noise-augmented inference already has robust behavior (green dashed line in Fig. 7.1-left). With that, we define RCP1 (robust conformal prediction with one sample) that returns robust sets using a single inference per point. In practice, the resulting sets have the same guarantee, and similar size, to the previous SOTA which needs around 100 samples per input instead. By nullifying the need for sampling, we can use even larger models (like vision transformers) to return even smaller sets (see Fig. 7.1-middle). Importantly, we do not compete with the SOTA equipped with unlimited sampling (compute) budget. Instead, we propose a compute-friendly alternative that still produces small prediction sets in regimes infeasible for the other smoothing-based RCP methods (large models and limited computational power).

RCP1 is similar to vanilla CP only with two changes: (i) we use noise-augmented input (from the smoothing) to compute the scores, and (ii) we calibrate with a conservative $1 - \alpha'$ nominal coverage chosen such that our certified lower bound (in § 7.3) remains above $1 - \alpha$. RCP1 works with any binary certificate (see subsubsection 7.3.1), is agnostic to the model, the distribution of inputs, and score function, and interestingly, it is task independent – same binary certificate works for both classification, or regression. We use a similar process to define a smoothing-based conformal risk control (Fig. 7.1-right).

7.2 Background

CP requires a holdout set of labeled calibration points $\mathcal{D}_{\text{cal}} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$ that are exchangeable with the future test point $\boldsymbol{x}_{n+1}$. From the model's output, we define a score function $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ where it quantifies the agreement between $\boldsymbol{x}$, and y, e.g. softmax; see § E.1 for other scores. Vovk et al. (2005) show that under exchangeability (vanilla setup) the set $C_0(\boldsymbol{x}_{n+1}) = \{y : s(\boldsymbol{x}_{n+1}, y) \geq q\}$ for $q = \mathbb{Q}(\alpha; \{s(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}\})$ contains the true label y_{n+1} at least with $1 - \alpha$ probability.

$$\Pr_{\mathcal{D}_{n+1}}[y_{n+1} \in C_0(\boldsymbol{x}_{n+1})] = \Pr_{\mathcal{D}_{n+1}}[s(\boldsymbol{x}_{n+1}, y_{n+1}) \geq q] \geq 1 - \alpha \tag{61}$$

Here $\mathcal{D}_{n+1} = \mathcal{D}_{\text{cal}} \cup \{(\boldsymbol{x}_{n+1}, y_{n+1})\}$, and $\mathbb{Q}(\alpha; \mathcal{A})$ is the $\lfloor \alpha \cdot (1 - \frac{1}{n}) \rfloor$ quantile of the set $\mathcal{A}$. While the coverage guarantee is agnostic to the model (and the score), better model or score functions reflects in properties like the prediction set size (a.k.a efficiency). While methods like (Zargarbashi et al., 2024) require bounded score, our results are also agnostic to the choice of the score function (bounded or unbounded).

Threat model. We consider the worst case (or adversarial) perturbation, which yields a more powerful guarantee compared to probabilistic robustness e.g. from Ghosh et al. (2023). In our threat model, the adversary aims to decrease the empirical coverage below the guaranteed $1 - \alpha$ by adding an imperceptible noise to the test points (evasion).

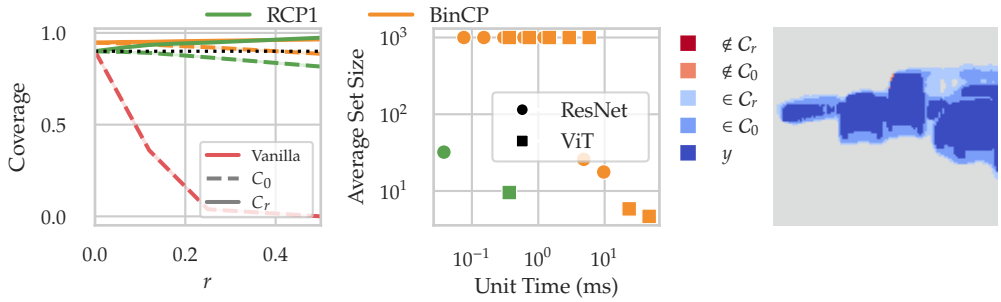

Figure 7.1: [Left] Empirical coverage under adversarial attack for vanilla CP, RCP1, and BinCP. Here C_r denotes prediction sets certified up to radius r, while C_0 uses the same randomized augmentation (guaranteed only for $r = 0$ – certified only for clean inputs). The dashed lines show the empirical coverage of C_0 under attack: vanilla CP degrades sharply, while smooth inference in BinCP and randomized augmented inference in RCP1 remain substantially more resilient. The solid lines show the empirical coverage of the robust sets C_r for BinCP and RCP1 on the same adversarial examples. For smooth methods we use `PGDSmooth` (Salman et al., 2019), and for vanilla CP we use standard `PGD`. Results are for CIFAR-10 with a `ResNet` and $\sigma = 0.5$. [Middle] The average time to compute, and the average set size for both RCP1 and BinCP, with ResNet and ViT models on the ImageNet dataset; both axis are log-scaled and pareto-optimal points are at the lower-left. RCP1 is more efficient. Both plots are with $\sigma = 0.5$. [Right] Smoothing-based robust conformal risk control. We show the coverage and miscoverage of the RCP1 mask for the class "car" in the segmentation task. Here risk is set to false negative rate.

The set of all possible perturbations is defined as a ball $\mathcal{B} : \mathcal{X} \to 2^{\mathcal{X}}$ around the clean input. We define an inverted ball $\mathcal{B}^{-1}$ as the smallest set that contains the original (clean) point from any possible perturbation; i.e. $\forall \tilde{x} \in \mathcal{B}(x) \Rightarrow x \in \mathcal{B}^{-1}(\tilde{x})$. For images, a common threat model is ℓ_2-norm: $\mathcal{B}_r(x) = \{\tilde{x} : \|\tilde{x} - x\|_2 \leq r\}$ where r is the radius of the perturbation. For symmetric balls like ℓ_2 we have $\mathcal{B} = \mathcal{B}^{-1}$, but this does not hold in general (e.g. (Bojchevski et al., 2020)).

Robust Conformal Prediction (RCP). Robust CP extends the guarantee in Eq. 61 to the worst case noise. For $\mathcal{B}$, prior works define a robust (conservative) prediction set $C_{\mathcal{B}}$ satisfying the following

$$\Pr_{\mathcal{D}_{n+1}} [y_{n+1} \in C_{\mathcal{B}}(\tilde{x}_{n+1}), \forall \tilde{x}_{n+1} \in \mathcal{B}(x_{n+1})] \geq 1 - \alpha \tag{62}$$

Eq. 62 is only meaningful for deterministic sets. We discuss this subtle point in § 7.3 and subsubsection E.2.1. Earlier smoothing-based RCP methods implicitly remove all inherent randomness, which makes the definition applicable to them. These methods can be summarized with the following two arguments: (i) for the exchangeable x_{n+1} CP covers the true label with $1 - \alpha$ probability, (ii) if the clean x_{n+1} was originally covered by (vanilla) CP, robust CP also covers $\tilde{x}_{n+1}$, because if a clean score is above q its upper bound (over any perturbed input) is also above q (Zargarbashi and Bojchevski, 2025). Thus, the vanilla set for x_{n+1} is a subset of the robust set for $\tilde{x}_{n+1}$. This results in robust coverage of *at least* $1 - \alpha$. To account for the inherent randomness in CP, in § 7.3, we redefine the threat model, and replace the argument (ii) by the following: "the perturbed $\tilde{x}_{n+1}$ has a higher probability to be in the robust prediction set compared to x_{n+1} being in the vanilla set". The new formulation still addresses the worst-case perturbation.

Certified bounds. For any function f and ball $\mathcal{B}(\boldsymbol{x})$ define the certified lower bound as $\mathrm{c}^{\downarrow}[f, \boldsymbol{x}, \mathcal{B}] \leq \inf\{f(\boldsymbol{z}) : \boldsymbol{z} \in \mathcal{B}(\boldsymbol{x})\}$, and similarly $\mathrm{c}^{\uparrow}[\cdot, \cdot, \cdot]$ as the upper bound (with sup). With this definition, for each $\boldsymbol{x}$ we have $\forall \tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x}), \mathrm{c}^{\downarrow}[s(\cdot, y), \boldsymbol{x}; \mathcal{B}] \leq s(\tilde{\boldsymbol{x}}, y) \leq \mathrm{c}^{\uparrow}[s(\cdot, y), \boldsymbol{x}, \mathcal{B}]$ where we plug in the score function for f. Zargarbashi et al. (2024) show that given these certified bounds within $\mathcal{B}$, the conservative sets defined either as $C_{\mathcal{B}}(\boldsymbol{x}_{n+1}) = \{y : \mathrm{c}^{\uparrow}[s(\cdot, y_{n+1}), \boldsymbol{x}_{n+1}; \mathcal{B}^{-1}] \geq q\}$ (test-time RCP), or similarly, $C_{\mathcal{B}}(\boldsymbol{x}_{n+1}) = \{y : s(\boldsymbol{x}_{n+1}, y) \geq \bar{q}\}$ for $\bar{q} = \mathbb{Q}\left(\alpha; \{\mathrm{c}^{\downarrow}[s(\cdot, y_i), \boldsymbol{x}_i; \mathcal{B}] : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}\}\right)$ (calibration-time RCP) attain $1 - \alpha$ robust coverage.

Randomized smoothing. One approach to compute these upper/lower bounds for any black-box model, or score is randomized smoothing. A smoothing scheme $\xi : \mathcal{X} \to \mathcal{X}$ adds a random noise to the input – maps it to a random point close to it. A common smoothing for continuous data (e.g. images) is the Gaussian smoothing $\xi(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon}$ is an isotropic Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$. While our method works for any smoothing, for easier notation we further use $\boldsymbol{x} + \boldsymbol{\epsilon}$ instead of $\xi(\boldsymbol{x})$.

For any score function s, the distribution of the smooth scores $s(\boldsymbol{x} + \boldsymbol{\epsilon}, y)$ changes slowly. This enables us to compute tight bounds on the smooth statistics (mean, quantile, etc.) within $\mathcal{B}$, or $\mathcal{B}^{-1}$. RSCP (Gendler et al., 2021; Yan et al., 2024), and CAS (Zargarbashi et al., 2024) set the score function directly to the mean of the distribution. BinCP (Zargarbashi and Bojchevski, 2025), uses the p-quantile instead. These statistics are often intractable to compute and therefore estimated using Monte-Carlo sampling, followed by a finite sample correction. RCP1 however nullifies the need to estimate these statistics. We discuss the related work further in § 7.5.

7.3 RCP1: Robust CP with One Sample

High level view. We prove that when the scores are computed on noise-augmented inputs, i.e. using $s(\boldsymbol{x} + \boldsymbol{\epsilon}, y)$ instead of $s(\boldsymbol{x}, y)$ for both calibration and prediction (see Algorithm 2), vanilla CP already yields robust prediction sets - and its coverage under perturbation can be bounded. We provide a sketch of our arguments. To prove it we first assume an abstract value β_{n+1} as the coverage probability of a specific clean point $\boldsymbol{x}_{n+1}$, which is over the random noise and the inherent randomness in the score. With $\tilde{\beta}_{n+1}$ as the coverage probability for $\tilde{\boldsymbol{x}}_{n+1}$ (again over augmented input) we show that $\tilde{\beta}_{n+1}$ can be lower bounded using the randomized smoothing certificates. By showing the convexity of the certificate w.r.t. β_{n+1}, we can directly lower bound the expected coverage over all inputs which is $1 - \alpha$. We never need to compute the abstract β_{n+1} or $\tilde{\beta}_{n+1}$. This sketch is illustrated in Fig. 7.2.

Limitation of Eq. 62. The universal quantifier in Eq. 62 implies that for $\geq 1 - \alpha$ fraction of test points, *all* $\tilde{\boldsymbol{x}}_{n+1}$ must be deterministically covered, including the clean $\boldsymbol{x}_{n+1}$. However, many CP methods (like APS (Angelopoulos et al., 2024)) incorporate internal stochasticity (e.g. to break ties), making the coverage event a random variable rather than a binary indicator. This is true even before we add our noise on top. Hence, Eq. 62 does not reduce to Eq. 61 for $r = 0$. For random sets, adversary can reduce the coverage *probability* for each $\boldsymbol{x}_{n+1}$. With u encoding all the (inherent) randomness in the sets, C_0 as the vanilla set, and C_r as the robust set for a ball $\mathcal{B}_r$, we rewrite the guarantee as:

$$\mathbb{E}_{\mathcal{D}_{n+1}}[\min_{\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}_r(\boldsymbol{x}_{n+1})} \Pr_u[y_{n+1} \in C_r(\tilde{\boldsymbol{x}}_{n+1}; u)]] \geq \Pr_{\mathcal{D}_{n+1}, u}[y_{n+1} \in C_0(\boldsymbol{x}_{n+1}; u)] \geq 1 - \alpha \quad (63)$$

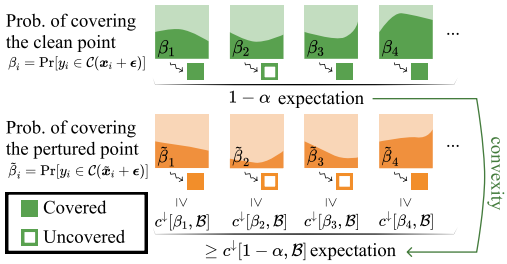

Figure 7.2: Illustration of the theory behind RCP1. The probabilities β_i never need to be computed, since due to the convexity of $c^{\downarrow}$ we can directly work with $1-\alpha$. Further description in § 7.3.

Algorithm 2. RCP1; the colored part shows the difference with vanilla CP.

Input: Calibration set $\mathcal{D}_{\text{cal}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$; nominal coverage $1-\alpha \in (0,1)$; score $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$; potentially perturbed test point $\tilde{\boldsymbol{x}}_{n+1}$

1: Compute $s_i \leftarrow s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}$.
2: Set $1-\alpha' \leftarrow c^{\uparrow}[1-\alpha, \mathcal{B}^{-1}]$ ▷ e.g., Gaussian smoothing with $\mathcal{B}_r$: $\Phi_\sigma(\Phi_\sigma^{-1}(1-\alpha)+r)$.
3: Set $\bar{q}_\alpha = \mathbb{Q}\left(\alpha', \{s_i\}_{i=1}^n\right)$.
4: For input $\tilde{\boldsymbol{x}}_{n+1}$ **return**

$$C_r(\tilde{\boldsymbol{x}}_{n+1}) = \{y : s(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, y) \geq \bar{q}_\alpha\}$$

General worst-case guarantee. We prove the lower bound coverage guarantee in RCP1, as implemented in Algorithm 2. We state the result in terms of random variables, which have a specific realization in practice. Let $Z_i : i \in [n+1]$ be $n+1$ exchangeable random variables, where $Z_i = (X_i, Y_i)$ for $X_i \in \mathcal{X}$ (e.g. $\mathcal{X} = \mathbb{R}^d$), and $Y_i \in \mathcal{Y}$ (e.g. $\mathcal{Y} = [K]$). Let $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be any measurable score function. Let $E_i : i \in [n+1]$ be i.i.d. random variables from a distribution supported on $\mathcal{X}$ (e.g. $E_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$). We define $\hat{S}_i = s(X_i, Y_i)$, and $S_i = s(X_i + E_i, Y_i)$. Let $\delta \in \mathcal{B}_r$ be any arbitrary perturbation up to radius r, define $\tilde{X}_{n+1} = X_{n+1} + \delta$, and $\tilde{S}_{n+1} = s(\tilde{X}_{n+1} + E_{n+1}, Y_{n+1})$ accordingly.

Proposition 7.1. *Let $q = \mathbb{Q}(\alpha; s(X_i + E_i, Y_i) : (X_i, Y_i) \in \mathcal{D}_{\text{cal}})$. Given a certified lower bound $c^{\downarrow}[\cdot, \mathcal{B}]$ as later defined in Eq. 64, and $\mathcal{E}_{n+1} = \{E_i\}_{i=1}^{n+1}$, for any perturbation $\delta \in \mathcal{B}_r$ we have*

$$\Pr_{\mathcal{D}_{n+1}, \mathcal{E}_{n+1}}[s(\tilde{X}_{n+1} + E_{n+1}, Y_{n+1}) \geq q] \geq c^{\downarrow}[1-\alpha, \mathcal{B}]$$

Proof. Adding i.i.d. noise E_i is permutation equivariant, thus using S_i's doesn't break exchangeability Angelopoulos et al. (2024). From Vovk et al. (2005) we have $\Pr[S_{n+1} \geq q] \geq 1-\alpha$ for $q = \mathbb{Q}\left(\alpha; \{S_i\}_{i=1}^n\right)$, where the probability is over $\mathcal{D}_{n+1}$, and $\mathcal{E}_{n+1}$. We can rewrite this as

$$\Pr_{\mathcal{D}_{n+1}, \mathcal{E}_{n+1}}[S_{n+1} \geq q] = \mathbb{E}_{\mathcal{D}_{n+1}, \mathcal{E}_n}\left[\Pr_{E_{n+1}}[S_{n+1} \geq q] \mid \mathcal{D}_{n+1}, \mathcal{E}_n\right] = \mathbb{E}_{\mathcal{D}_{n+1}, \mathcal{E}_n}[\beta_{n+1}] \geq 1-\alpha$$

where we define $\beta_{n+1} := \Pr_{E_{n+1}}[S_{n+1} \geq q \mid \mathcal{D}_{n+1}, \mathcal{E}_n]$. Here, β_{n+1} is only a probability over the last noise E_{n+1} (and any other internal randomness in the score) for a fixed $\mathcal{D}_{n+1}$ and $\mathcal{E}_n$. We call β_{n+1} the clean instance-wise coverage. Similarly for $\tilde{X}_{n+1}$ we define $\tilde{\beta}_{n+1}$.

We can bound any smooth binary function with an existing certified lower bound $c^{\downarrow}[\cdot, \mathcal{B}]$. Formally,

$$\Pr_{E_{n+1}}[S_{n+1} \geq q \mid \mathcal{D}_{n+1}, \mathcal{E}_n] = \beta_{n+1} \quad \Rightarrow \quad \Pr_{E_{n+1}}[\tilde{S}_{n+1} \geq q \mid \mathcal{D}_{n+1}, \mathcal{E}_n] \geq c^{\downarrow}[\beta_{n+1}, \mathcal{B}]$$

Note that here both β_{n+1} and $\tilde{\beta}_{n+1}$ share the same $\mathcal{D}_{n+1}$, and $\mathcal{E}_n$ and both are over the random variable E_{n+1} and the inherent randomness of the score. Later in Lemma 7.2, we show that the function $c^{\downarrow}[\beta, \mathcal{B}]$ is convex and increasing in β. This helps us to bound

the adversarial coverage guarantee as

$$\begin{aligned}\Pr\left[Y_{n+1} \in C(\tilde{X}_{n+1})\right] = \Pr\left[\tilde{S}_{n+1} \geq q\right] &= \mathbb{E}_{\mathcal{D}_{n+1},\mathcal{E}_n}[\Pr_{E_{n+1}}[\tilde{\beta}_{n+1}]] \\ \text{(from certificate)} &\geq \mathbb{E}_{\mathcal{D}_{n+1},\mathcal{E}_n}[\mathrm{c}^{\downarrow}[\beta_{n+1}, \mathcal{B}]] \\ \text{(from convexity (Lemma 7.2), } \mathbb{E}[\mathrm{c}^{\downarrow}[\beta,\cdot]] \geq \mathrm{c}^{\downarrow}[\mathbb{E}[\beta],\cdot]]) &\geq \mathrm{c}^{\downarrow}[\mathbb{E}_{\mathcal{D}_{n+1},\mathcal{E}_n}[\beta_{n+1}], \mathcal{B}] \\ &\geq \mathrm{c}^{\downarrow}[1-\alpha, \mathcal{B}]\end{aligned}$$

where the last inequity holds due to vanilla CP and monotonicity. □

Just like with any other CP with any kind of randomness in the score (e.g. APS), the guarantee only holds marginally over $\mathcal{E}_{n+1}$ and the internal randomness. In other words, the coverage probability is higher than $\mathrm{c}^{\downarrow}[\beta_{n+1}, \mathcal{B}]$ for specific $\tilde{X}_{n+1}$, and $\mathrm{c}^{\downarrow}[1-\alpha, \mathcal{B}]$ on average, if we draw a random $\mathcal{E}_{n+1}$. If we instead fixed $\mathcal{E}_{n+1}$ and the adversary knows the noise, the guarantee can easily break. Note that β_{n+1} is an abstract quantity, the probability that $X_{n+1} + E_{n+1}$ is covered. In principle we can not estimate β_{n+1}, since the label is not known. Nonetheless, due to the convexity, we can lower bound the coverage guarantee directly without that information.

Instance-wise worst case coverage. Proposition 7.1 relies on lower bounding the worst-case (adversarial) $\tilde{\beta}_{n+1} = \Pr_{\epsilon_{n+1}}[s(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}_{n+1}, y_{n+1}) \geq q]$, for the perturbed $\tilde{\boldsymbol{x}}_{n+1} = \boldsymbol{x}_{n+1} + \boldsymbol{\delta} \in \mathcal{B}(\boldsymbol{x}_{n+1})$ given $\beta_{n+1} := \Pr_{\epsilon_{n+1}}[s(\boldsymbol{x}_{n+1} + \boldsymbol{\epsilon}_{n+1}, y_{n+1}) \geq q]$. Here, $(\boldsymbol{x}_{n+1}, y_{n+1})$ and $\boldsymbol{\epsilon}_{n+1}$ are realizations of (X_{n+1}, Y_{n+1}) and E_{n+1}. This is conditional to q, and hence to $\mathcal{D}_{\text{cal}}$ and $\mathcal{E}_n$. Formally, we define a binary classifier, $f(\boldsymbol{z}) = \mathbb{I}[s(\boldsymbol{z}, y_{n+1}) \geq q]$ and $g(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{\epsilon}_{n+1}}[f(\boldsymbol{z} + \boldsymbol{\epsilon}_{n+1})]$ for which we have $\beta_{n+1} := g(\boldsymbol{x}_{n+1})$. Note that $\mathcal{X}$ is a convex subset of $\mathbb{R}^d$ and the score $s(\cdot, y)$ is continuous everywhere, therefore our classifier is measurable (Massena et al., 2025). We can lower bound $\tilde{\beta}_{n+1} = \min_{\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})} g(\tilde{\boldsymbol{x}}_{n+1})$ and therefore $g(\tilde{\boldsymbol{x}}_{n+1})$ for the given $\boldsymbol{x}_{n+1}$ using the existing (binary) classification certificates, e.g. Cohen et al. (2019).

Certified lower bound $\tilde{\beta}$. A smoothing binary certificate computes the bound $\mathrm{c}^{\downarrow}[g(\cdot), \boldsymbol{x}_{n+1}, \mathcal{B}]$ regardless of the original definition of f – mechanics of the score function or model – and only as a function of the value $\beta = g(\boldsymbol{x}_{n+1})$. We use the $\mathrm{c}^{\downarrow}[\beta, \mathcal{B}]$ notation, following Zargarbashi and Bojchevski (2025). For the known $\boldsymbol{x}_{n+1}$, a (pointwise) certified lower bound on $g(\tilde{\boldsymbol{x}}_{n+1}) : \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$ is obtained by searching for the worst measurable binary function $h : \mathcal{X} \to \{0, 1\}$ in $\mathcal{H}$ (set of all measurable functions) such that h has the same smooth output as f at $\boldsymbol{x}_{n+1}$. Formally:

$$\mathrm{c}^{\downarrow}[\beta, \mathcal{B}_r] = \min_{h \in \mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[h(\tilde{\boldsymbol{t}} + \boldsymbol{\epsilon}) = 1] \quad \text{s.t.} \quad \Pr_{\boldsymbol{\epsilon}}[h(\boldsymbol{t} + \boldsymbol{\epsilon}) = 1] = \mathbb{E}_{\boldsymbol{\epsilon}}[g(\boldsymbol{x}_{n+1})] = \beta \tag{64}$$

The pair $(\boldsymbol{t}, \tilde{\boldsymbol{t}})$ are called canonical points. Cohen et al. (2019); Yang et al. (2020) discuss this in detail. Intuitively, the optimization in Eq. 64 is translation (and in some cases rotation) invariant, and with the symmetries in the ball and the smoothing scheme, for any $\boldsymbol{x}$, and $\tilde{\boldsymbol{x}}$ we can use a fixed set of canonical points. For ℓ_p balls, and symmetric additive smoothing (including isotropic Gaussian noise) these points are one at the center, and the other at the edge (or vertex) of the ball; i.e. $\boldsymbol{t} = [0, 0, \ldots, 0]$, and $\tilde{\boldsymbol{t}} = [r, 0, \ldots, 0]$. For a detailed discussion also see section D.1 from Zargarbashi and Bojchevski (2025). Since the function f itself is a feasible solution to Eq. 64, it is a valid lower bound for $g(\boldsymbol{x}_{n+1})$.

The mean-constrained binary certificate in Eq. 64 is a common problem in the randomized smoothing literature. It is efficiently solvable and in many cases has a

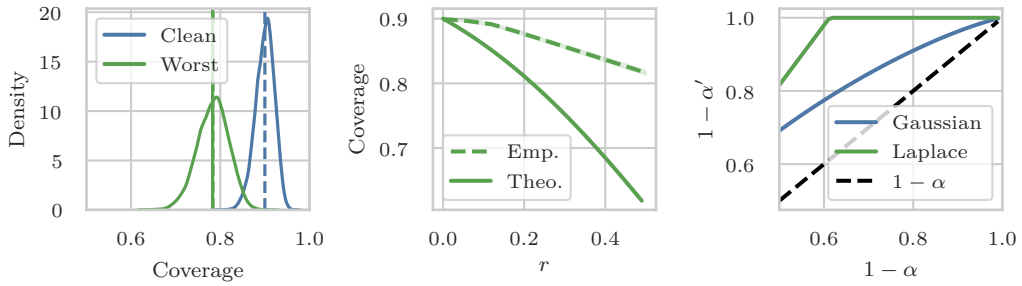

Figure 7.3: [Left] Samples from the Beta distribution of clean coverage, and worst-case coverage. The dashed line is the empirical average, and the overlapping solid line is $\mathrm{c}^{\downarrow}[1-\alpha, \mathcal{B}]$. [Middle] The empirical, and theoretical worst-case coverage. We show the empirical coverage value under the PGDSmooth attack with $\sigma = 0.5$ over various radii. We use the same sigma to show the theoretical lower bound coverage.[Right] The robust $1-\alpha'$ for two smoothing schemes. We report the $\mathrm{c}^{\downarrow}$ function for Gaussian and Laplace smoothing both with $\sigma = 0.5$. The plots are not empirical.

closed form solution. For the isotopic Gaussian smoothing with ℓ_2 (and ℓ_1) ball the lower bound is $\tilde{\beta} = \Phi_\sigma(\Phi_\sigma^{-1}(\beta) - r)$ where Φ_σ is the CDF of the Gaussian distribution $\mathcal{N}(0, \sigma)$ (Kumar et al., 2020). Using the recipe from Yang et al. (2020), in subsubsection 7.3.1, we discuss how to compute $\mathrm{c}^{\downarrow}[p, \mathcal{B}]$ for other smoothing schemes.

Lemma 7.2. $\mathrm{c}^{\downarrow}[\beta, \mathcal{B}]$ *as the solution to Eq. 64 is convex and monotonically increasing w.r.t.* β.

We defer the proofs to subsubsection E.2.2. There we rigorously prove Lemma 7.2 directly from the definition of Eq. 64 via duality. Here we provide a sketch of an alternative proof that is insightful. Lee et al. (2019) show that to solve Eq. 64, the space $\mathcal{X}$ can be divided to (finite or infinite) regions of constant likelihood ratio $\mathcal{R}_t = \{\boldsymbol{z} : \frac{\Pr[\boldsymbol{z}=\tilde{\boldsymbol{t}}+\boldsymbol{\epsilon}]}{\Pr[\boldsymbol{z}=\boldsymbol{t}+\boldsymbol{\epsilon}]} = c_t\}$. If we can sort these regions in descending order w.r.t c_t, the problem reduces to the following linear program which is a fractional knapsack problem:

$$\mathrm{c}^{\downarrow}[\beta, \mathcal{B}] = \min_{\boldsymbol{h} \in [0,1]^T} \boldsymbol{h}^\top \cdot \boldsymbol{q} \quad \text{s.t.} \quad \boldsymbol{h}^\top \cdot \boldsymbol{p} = \beta$$

where T is number or regions, h_t is the average value of $h(\boldsymbol{z})$ inside the region $\mathcal{R}_t$, $p_t = \Pr[\boldsymbol{t} + \boldsymbol{\epsilon} \in \mathcal{R}_t], q_t = \Pr\left[\tilde{\boldsymbol{t}} + \boldsymbol{\epsilon} \in \mathcal{R}_t\right]$, and the vectors $\boldsymbol{h}, \boldsymbol{p}, \boldsymbol{q}$ gather all h_t, p_t, q_t's (see Lee et al. (2019) for the derivation). W.l.o.g assume that the $\boldsymbol{p}$, and $\boldsymbol{q}$ are sorted decreasingly w.r.t. c_t. The optimal solution is $\boldsymbol{h}^\star = [1, 1, \ldots, m, 0, \ldots, 0]$, for some $m \in (0, 1)$ which is to fill regions in order up to when $\boldsymbol{h} \cdot \boldsymbol{p}$ reaches β. Each index of $\boldsymbol{h}^\star$ is one region being filled and by setting $h_t^* = 1$ the $\boldsymbol{h} \cdot \boldsymbol{q}$, and $\boldsymbol{h} \cdot \boldsymbol{p}$ increase by q_t, and p_t. Therefore $\mathrm{c}^{\downarrow}[\beta, \mathcal{B}] = \boldsymbol{h}^\star \cdot \boldsymbol{q}$ is a continuous piecewise linear function with slope of q_t/p_t which is increasing across regions. A piecewise linear function with an increasing slope in each piece is convex. This convexity directly helps us to bound $\mathbb{E}[\mathrm{c}^{\downarrow}[\beta, \mathcal{B}]] \geq \mathrm{c}^{\downarrow}[\mathbb{E}[\beta], \mathcal{B}]$.

Note that in Proposition 7.1 the guarantee is over the coverage probability and independent of the setup; therefore, without any change, one can use it to make conformal regression robust. Regardless of the downstream task the certificate is always for binary classification. Furthermore, the result is not restricted to a specific scheme and can be used for any smoothing and perturbation ball (see subsubsection 7.3.1).

Coverage distribution. The expected coverage probability is itself a random variable with expectation higher than $1-\alpha$. Under mild assumptions, $\Pr[S_{n+1} \geq q \mid \mathcal{D}_{\text{cal}}] \sim \text{Beta}((1-\alpha)\cdot(n+1), \alpha(n+1))$ (Angelopoulos et al., 2024). That is, for any given fixed calibration set, the coverage fluctuates around $1-\alpha$, with variance inversely proportional to the size of the calibration set. Since in practice we only have one calibration set, understanding this distribution, and its variance, is important. While we do not know the distribution of the robust coverage, we can compute a conservative estimate by convolving the CDF of Beta and the function $\mathrm{c}^{\downarrow}[\cdot, \mathcal{B}]$ (see Fig. 7.3-left). Similarly, convexity helps to bound the mean of this new distribution as shown in Proposition 7.1. Note, our method does not take into account the distribution of the scores or inputs (unlike BinCP Zargarbashi and Bojchevski (2025)) and as a result it is very conservative. In Fig. 7.3-middle we show comparison of our guaranteed lower-bound coverage and the empirical coverage under adversarial attack, highlighting that our guarantee accounts for significantly more damage.

Maintaining $1-\alpha$ coverage. Proposition 7.1 says that under perturbation, the coverage guarantee of CP calibrated with $1-\alpha$ over augmented inference decreases at most by $\mathrm{c}^{\downarrow}[1-\alpha, \mathcal{B}]$. A simple solution to attain $1-\alpha$ robust coverage is to set the nominal coverage to a value $1-\alpha'$ such that $\mathrm{c}^{\downarrow}[1-\alpha', \mathcal{B}] \geq 1-\alpha$. In general, we can find $1-\alpha'$ using binary search, however, from (Zargarbashi and Bojchevski, 2025) we know that in smoothing schemes like Gaussian, we have $\mathrm{c}^{\uparrow}[\mathrm{c}^{\downarrow}[p, \mathcal{B}], \mathcal{B}^{-1}] = p$ (see subsubsection E.2.3, Lemma E.1). Therefore, to attain $1-\alpha$ robust coverage, we only need to set the threshold as the $\mathrm{c}^{\uparrow}[1-\alpha, \mathcal{B}^{-1}]$ quantile of the calibration scores (see Fig. 7.3-right).

7.3.1 Robust Conformal Sets with Randomized Smoothing of All Shapes and Sizes

Both RCP1, and BinCP work with any smoothing and ball $\mathcal{B}$. However, some binary certificates are given as a robust radius r^* – the radius up to which the prediction remains the same, i.e. $\mathrm{c}^{\downarrow}[p, \mathcal{B}_{r^*}] = 0.5$. Yang et al. (2020) provide a recipe to compute the r^* for general ℓ_p certificate under additive randomized smoothing. We tweak their "differential" method to derive probability bounds. We phrase Proposition 7.3 in a notation close to Yang et al. (2020) and far from our own, however the takeaway is simple: in short we define $\Omega(p)$ such that $1/\Omega(p)$ encodes the minimum perturbation to make an infinitesimal increase in the worst case classifier with expected value β. We use line integral to find the $\mathrm{c}^{\uparrow}$ for the worst case classifier at radius r; formally $\sup\{\overline{\beta} : \int_{\beta}^{\overline{\beta}} \frac{1}{\Omega(p)} \mathrm{d}p \leq r\}$. We can find this supremum either analytically (see subsubsection E.2.4) or via binary search using the existing closed forms provided by Yang et al. (2020).

Proposition 7.3. *For a binary classifier $f(\boldsymbol{x})$, and an additive smoothing function ξ, Let $\boldsymbol{\mathcal{U}} = \{\boldsymbol{x} : f(\boldsymbol{x}) = 1\}$ be the decision boundary and $\boldsymbol{\mathcal{U}} - \boldsymbol{z}$ as the same set translated by $-\boldsymbol{z}$, such that $\boldsymbol{z}$ goes to the origin. Let $\xi(\boldsymbol{\mathcal{U}}) = \Pr_{\boldsymbol{\epsilon}\sim\xi}[\boldsymbol{\epsilon} \in \boldsymbol{\mathcal{U}}]$ be the expectation of the decision boundary under smoothing, and $\beta = \xi(\boldsymbol{\mathcal{U}} - \boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x}+\boldsymbol{\epsilon})]$. Define:*

$$\Omega(p) := \sup_{\boldsymbol{\delta}:\|\boldsymbol{\delta}\|=1} \sup_{\boldsymbol{\mathcal{U}}\in\mathbb{R}^d:\xi(\boldsymbol{\mathcal{U}})=p} \lim_{r\to 0^+} \frac{\xi(\boldsymbol{\mathcal{U}} - r\boldsymbol{\delta}) - p}{r}$$

Assuming $\Omega(p)$ is strictly positive for $p \in [\mathrm{c}^{\downarrow}[\beta, \mathcal{B}_r], \mathrm{c}^{\uparrow}[\beta, \mathcal{B}_r]]$, and defining $F(\gamma) =$

$\int_{\gamma}^{1/2} \frac{1}{\Omega(p)} \mathrm{d}p$:

$$\mathrm{c}^{\uparrow}[\beta, \mathcal{B}_r] = \sup\{\beta' : F(\beta') \geq F(\beta) - r\} \tag{65}$$

Similarly, we have $\mathrm{c}^{\downarrow}[\beta, \mathcal{B}_r] = \inf\{\beta' : F(\beta') \leq F(\beta) + r\}$.

7.3.2 Extension to Conformal Risk Control

We use robustness certificates to define smoothing-based robust risk control for the first time. Let $C_\lambda(\cdot)$ be a conformal set, where $\lambda \leq \lambda_{\max}$ controls the set size. For a risk function $\mathcal{L}(\boldsymbol{x}_i, y_i; \lambda) \in [a, b]$ that is right-continuous and non-increasing w.r.t. λ, if $\mathcal{L}(\boldsymbol{x}_i, y_i; \lambda_{\max}) \leq \alpha$, Angelopoulos et al. (2022) show:

$$\mathbb{E}_{\mathcal{D}_{n+1}}[\mathcal{L}(\boldsymbol{x}_{n+1}, y_{n+1}; \lambda^*)] \leq \alpha \quad \text{for} \quad \lambda^* = \inf\{\lambda : \frac{\sum_{i=1}^{n} \mathcal{L}(\boldsymbol{x}_i, y_i; \lambda) + b}{n+1} \leq \alpha\}$$

Here $\alpha \in [a, b]$ is any user adjusted risk level. Similar to conformal prediction, we can also define a randomly augmented risk function $\mathcal{L}(\boldsymbol{x}_i + \boldsymbol{\epsilon}_i, y_i; \lambda)$. The noise does not break the exchangeability and therefore $\mathbb{E}[\mathcal{L}(\boldsymbol{x}_{n+1} + \boldsymbol{\epsilon}_{n+1}, y_{n+1}; \lambda^*)] \leq \alpha$ for the λ^* computed on the randomly augmented calibration set. Due to the continuous nature of the risk function, we now use confidence certificates:

$$\mathrm{c}_c^{\uparrow}[\beta, \mathcal{B}] = \max_{h \in \mathcal{H}} \mathbb{E}[h(\tilde{\boldsymbol{x}}_{n+1})] \quad \text{s.t.} \quad \mathbb{E}[h(\boldsymbol{x}_{n+1})] \leq \beta \tag{66}$$

Here $h : \mathcal{X} \to [0, 1]$. Similarly, Eq. 66 can be efficiently solved, and for the Gaussian distribution it has a closed form solution of $b \cdot \Phi_\sigma(\Phi_\sigma^{-1}(\frac{\beta-a}{b-a}) + r) - a(1 - \Phi_\sigma(\Phi_\sigma^{-1}(\frac{\beta-a}{b-a}) + r))$. With $[a, b] = [0, 1]$ (e.g. for the false negative rate risk) the closed form is identical to the classification certificate (Kumar et al., 2020).

7.4 Experiments

Metrics and Baseline. We evaluate average set size (lower is better), and empirical coverage (exceeding $1 - \alpha$ on average). Note that in RCPs the empirical coverage conservatively exceeds $1 - \alpha$ by increasing r. Under perturbation this decreases at worst to $1 - \alpha$. As BinCP (Zargarbashi and Bojchevski, 2025) outperforms other robust CP approaches (Zargarbashi et al., 2024; Yan et al., 2024), we set it as our main comparison baseline. All recent smoothing-based RCPs return non-informative sets ($C(\boldsymbol{x}) = \mathcal{Y}$) for low number of samples (e.g. ≤ 32). Note that our main contribution is to return efficient sets with *one inference per input*; therefore we do not expect RCP1 to outperform BinCP for a large sample-rate. Our reported results are over 100 iterations with calibration set randomly sampled from the data. Further details are in § E.3, and the code is in our GitHub.

Since we certify the coverage guarantee (instead of scores), we can use the same *binary* certificate for both classification and regression tasks. We discuss the classification here, and defer the regression task to § E.3. The algorithm remains the same, only for the regression we use the absolute distance from the ground truth as the score. To the best of our knowledge, this is the first conformal regression certificate based on randomized smoothing.

Classification. We compare methods for the CIFAR10, and ImageNet datasets. We have two inference pipelines The original pipeline from BinCP, and CAS (computationally cheap setup): we use the ResNet models trained with noise augmentation from Cohen

et al. (2019). Because of the model size, large sample-rates, although inefficient, are not unrealistic. We also evaluate on an alternative more expensive pipeline outlined by Carlini et al. (2022): the input is first denoised by a diffusion model and then classified by a vision transformer. For CIFAR-10 we combine a 50M-parameter diffusion model from Dhariwal and Nichol (2021), with a `ViT-B/16` from Dosovitskiy et al. (2020), pretrained on ImageNet at 224×224 resolution and finetuned on CIFAR10 with 97.9% accuracy for the HuggingFace implementation. For ImageNet we use a 552M-parameter class-unconditional diffusion model followed by BEiT-L model (305M parameters) from Bao et al. (2021) achieving 88.6% top-1 validation accuracy. We use the implementation provided by the timm library (Wightman, 2019).

Smaller set sizes. Increasing the sample-rate (number of model forwards) in BinCP decreases the set size. Fig. 7.1-middle compares the set size and computation time for BinCP and RCP1 on the ImageNet dataset. Here, RCP1 shows similar set size to BinCP with 64 to 128 inferences per point. We also compare set size per radii for CIFAR-10 in Fig. 7.7, and for ImageNet in Fig. 7.5. A single inference over the larger pipeline (diffusion and ViT) for RCP1 takes significantly less time compared to the cheaper pipeline with enough samples for BinCP. Therefore we can easily achieve a considerably better set size with an unnoticeably more computation only by using a better model. In Fig. 7.4-middle, and right we compare BinCP and RCP1 in set size per sample rate (for BinCP) and radius for the CIFAR-10 and ImageNet datasets. Our complete comparison on this experiment is in § E.3. Note that it is significantly inefficient to run ≥ 100 forwards passes per image on the ViT models. Additionally, we use the results in subsubsection 7.3.1 to show that the method works similarly for any smoothing scheme and threat model. For that we show the performance of BinCP (under two sample rates) and RCP1 for the ℓ_1 ball under uniform smoothing distribution in Fig. 7.6-right.

For a dataset like ImageNet (with 1000 classes), the average set size alone is not a measure of usability. Consider a CP returning 50% singleton sets and $|\mathcal{Y}|$ for the rest, compared to a CP returning sets of size 100 for all inputs. Surely, the latter option is not usable even though it has smaller average set size. Hence, we also report the proportion of the prediction sets with less than 5 elements in Fig. 7.5-right (also see § E.3). This metric is only trustworthy if the we don't sacrifice the coverage in smaller sets. In Fig. 7.4-left we show that these sets have coverage larger than $1 - \alpha$.

Small radii. Jeary et al. (2024), and Massena et al. (2025) propose RCP using verifiers and Lipschitz constant of the network. Although their result is for one order of magnitude smaller radii (e.g. 0.02 instead of 0.25), their methods are efficient by using one forward per input. With RCP1 being the same in that metric, we compare with them in Fig. 7.6-left and middle reporting performance on smaller radii. Aside better performance, RCP1 has a black-box access and works for any model. Intuitively, as shown in Fig. 7.1-left, smooth (or augmented) inference is significantly more robust to

Table 7.1: Estimated runtimes (in `HH:MM:SS`) for 1000 inputs using an H-100 GPU. Results are scaled from a full experimental run assuming a linear cost in both the number of inputs and samples.

Pipeline	Dataset	1 Sample	64 Samples	128 Samples	256 Samples
ViT	CIFAR-10	0:00:01	0:01:09	0:02:19	0:04:39
	ImageNet	0:00:33	0:35:30	1:11:00	2:22:01

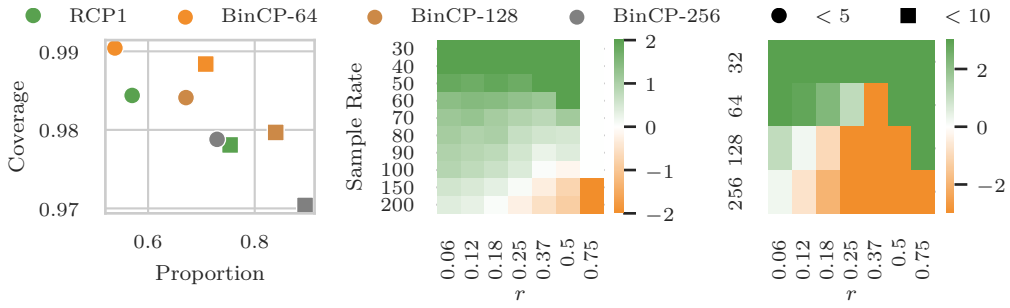

Figure 7.4: [Left] Proportion, and coverage of the prediction sets with ≤ 5, and ≤ 10 elements for the ViT model. [Middle] $|C_{r,\text{BinCP}}| - |C_{r,\text{RCP1}}|$ for the CIFAR-10 dataset with a ResNet model. [Right] ImageNet dataset and ViT models ($r = 0.25$). In all plots $\sigma = 0.5$ and RCP1 uses a single sample.

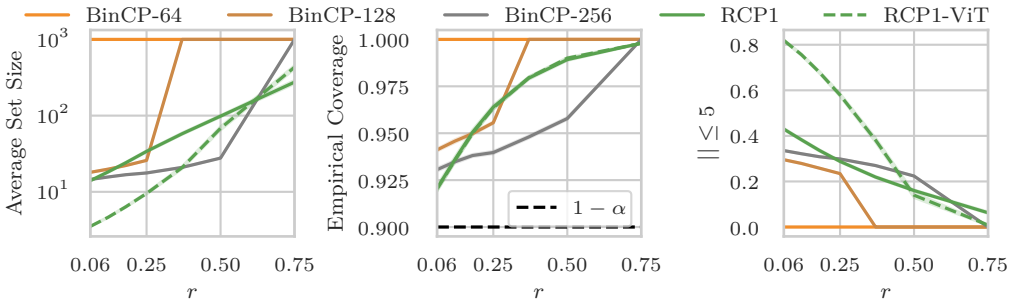

Figure 7.5: On ImageNet, cheap setup with $\sigma = 0.5$: [Left] Compares the average set size of BinCP with various sample rates to RCP1. [Middle] the empirical coverage. [Right] proportion of prediction sets with size less than 5 elements. Dashed line is the result of the ViT model with the same σ.

perturbations.

Time-comparison. With t_{cert}, and t_f as the time for computing bounds, and for the model's inference time, other smoothing based RCPs at best require $O(n_{\text{mc}} \times \mathcal{D}_{\text{cal}} \times t_f + t_{\text{cert}})$ for calibration where n_{mc} is the number of MC samples. For each test point they also require $O(n_{\text{mc}} \times t_f)$ time. RCP1 takes the same time as the normal model's inference plus an additional $O(t_{\text{cert}})$ for calibration. Similarly, RCP1 takes n_{mc} less memory compared to other smoothing RCPs. We show the runtime of the ViT pipeline for the used datasets in Table 7.1. Note that this is only the time to compute logits as the other processes (including certificates) are negligiable compared to it. The runtime of BinCP with a sample rate comparable to RCP1 is significantly high for large models like ViT; for instance, RCP1 and a comparable BinCP (with 128 samples on ImageNet) need ~2′,46", and ~5h 55′ to process 5000 images.

Table 7.2: Risk and mask size for the Cityscapes dataset. Risk level is 0.15, with 100 calibration points. The variance is not over calibration sampling but over the images and $r = 0.06$.

Class	Risk	Robust Risk	(True) Class Prop.	Mask Prop.	Robust Mask Prop.
Pedestrian	0.1474 ± 0.2797	0.1111 ± 0.2588	0.0160 ± 0.0279	0.1891 ± 0.1257	0.2522 ± 0.1312
Car	0.1466 ± 0.2582	0.0833 ± 0.2032	0.0539 ± 0.0545	0.0832 ± 0.0733	0.1101 ± 0.0807

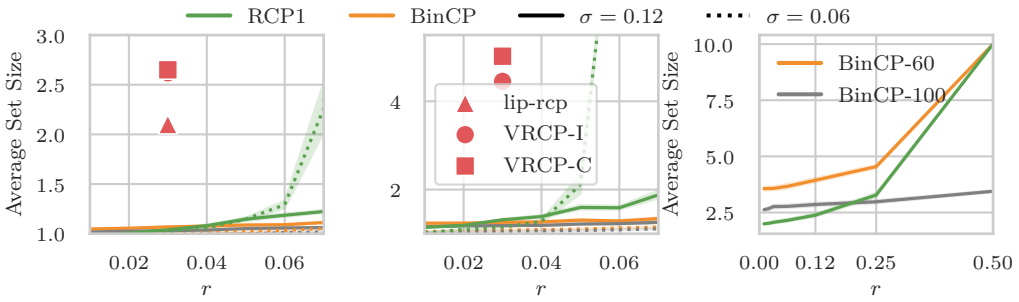

Figure 7.6: Performance on smaller radii in comparison with non-smoothing RCPs (Jeary et al., 2024; Massena et al., 2025) for CIFAR-10 and a ViT model. [Left] $1 - \alpha = 0.9$, and [Middle] $1 - \alpha = 0.95$. Smoothing is better. [Right] Performance of the methods for a Uniform-ℓ_1 certificate for $\epsilon \sim \text{Uniform}[-1/\sqrt{3}, 1/\sqrt{3}]$. Here we use the ℓ_1 certificate from Yang et al. (2020). Here the smoothing scheme is $\epsilon = \text{Uniform}[-\sigma/\sqrt{3}, \sigma/\sqrt{3}]$.

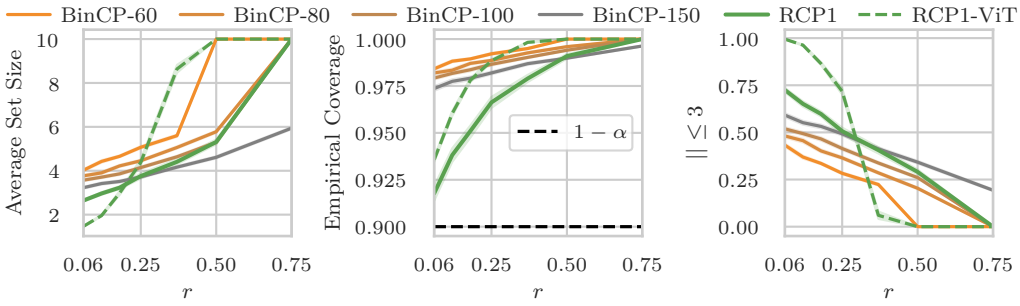

Figure 7.7: On CIFAR-10, ResNet model with $\sigma = 0.5$: [Left] The average set size compared to BinCP (with various sample rates), RCP1 [Middle] the empirical coverage, and [Right] proportion of sets with less than 3 labels. Dashed line shows the ViT model (with $\sigma = 0.25$).

Robust conformal risk control. We use the model from Fischer et al. (2021) on the CityScapes dataset (Cordts et al., 2016) which is a scene segmentation task. We mask the regions where a target class (e.g. car) might be present. The error function is the false negative ratio (FNR) – the portion of the pixels from the target class that is not masked. We take the $\exp(f(\boldsymbol{x}))_y$ as the score of the class y, and we set the mask as $\mathbb{I}[\exp(f(\boldsymbol{x}))_y \geq 1 - \lambda]$. Note that here the classes could possibly overlap. We calibrate by finding a λ that results in a FNR loss less than the user adjusted tolerable risk. So far, this is the first result for smoothing-based robust conformal risk control. Similar to RCP1, we first smooth the image data (one sample), then we compute the λ that results in $\mathrm{c}_c^{\downarrow}[\alpha, \mathcal{B}]$ risk. We report the results in Table 7.2, and show an example in Fig. 7.8.

Limitation: Increased variance. In Fig. 7.9, RCP1 shows considerably more randomness in the prediction sets compared to BinCP. This is essentially due to the random definition of the prediction set and the score function – the prediction set in RCP1 is a function of the random variable $\boldsymbol{\epsilon}$. This randomness does not affect the final robust / vanilla coverage.

7.5 Related Work

Gendler et al. (2021) initially proposed robust CP resilient to adversarial examples (worst-case noise) without accounting for finite samples (asymptotically valid setup). Yan et al. (2024) added finite sample correction and proposed a new score to return (set size) efficiency. Both mentioned works were using randomized smoothing and the mean of the smooth score to bound the worst case perturbations. Zargarbashi et al. (2024) (CAS) proposed to use the CDF information of the smooth score – a more restrictive constraint and therefore returned smaller prediction sets. All of these methods required unrealistically expensive setup with 10^4 MC samples to be able to return acceptably small sets. Zargarbashi and Bojchevski (2025) (BinCP) defined a quantile-based score over the distribution of the smooth scores, that allowed same set size as CAS with orders of magnitude less samples (e.g. 200 would be sufficient). In contrast with all of the methods RCP1 works with a single augmented sample and without involving finite sample correction, which lies at the computation efficiency side of this trade-off.

Orthogonal to randomized smoothing, Jeary et al. (2024), and Massena et al. (2025) use verifiers and Lipschitz continuity of the networks to bound the score function. Their robust radii are one order of magnitude smaller than smoothing RCP, but instead they do not require many forward passes per input. In Fig. 7.6 we show that our approach (with the same computational efficiency) provides smaller prediction sets outperforming their results; plus, our approaches works for any black-box model.

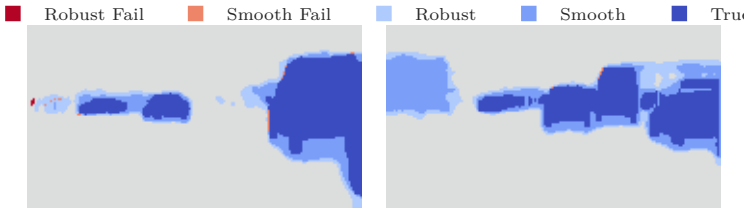

Figure 7.8: Performance of vanilla and robust risk control. From lighter to darker the colors are robust region, vanilla (non-robust) region, and the ground truth region. Here $\sigma = 0.25$, and $r = 0.06$.

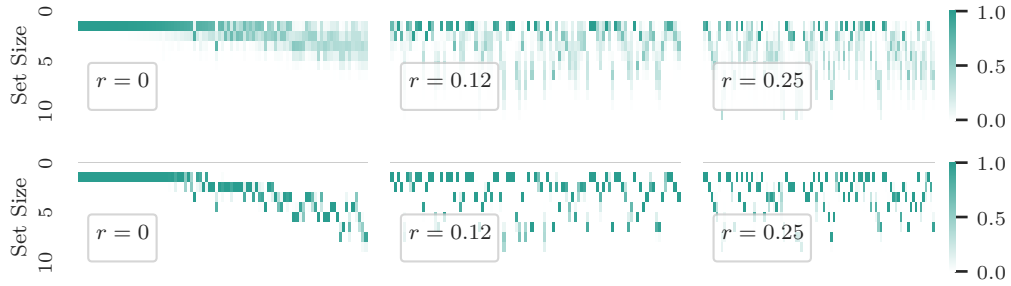

Figure 7.9: Randomness in set sizes for a CIFAR-10 dataset and ResNet model. The x-axis sorts datapoints in a fixed order (same across all plots), and the y axis shows the set size. The intensity of pixels shows the probability of a specific set size for that point over ϵ. [Top] shows RCP1, and [Bottom] BinCP (150 samples). RCP1 has larger variance compared to BinCP.

Notably all mentioned works provide robustness to the worst-case noise which is orthogonal to probabilistically robust CP Ghosh et al. (2023).

7.6 Conclusion

While offering small sets for larger radii, smoothing-based RCP methods need many forward passes per input. Instead, we show that noise-augmented inference combined with CP is inherently robust, and with that, we propose RCP1 which needs only one forward pass per input. Our approach returns sets with size similar to state of the art while nullifying the need of many MC samples. Prior smoothing RCPs provide their guarantee by lower bounding the scores, hence they need to estimate (some statistic about) the distribution of score for each individual input. Alternatively we only apply the lower bound on the coverage guarantee which is known in prior to be $1 - \alpha$.

8 Front Loaded Conformal Prediction: Heavy Calibration, light weight inference

Abstract

Robust conformal prediction (RCP) addresses confidence miscalibration in machine learning models by producing prediction sets with guaranteed coverage — these sets are guaranteed to include the true label with a user-specified high probability, even under worst-case noise. Recent works use randomized smoothing, as it provides robustness for black-box models at larger radii. Currently, there exist two setups for smoothing-based RCP: one requires extensive Monte Carlo sampling at calibration and test time but results in smaller prediction sets; the other setup produces larger prediction sets but uses a single sample at both stages. In deployment, calibration—as a one-time preprocessing step—can accommodate substantially higher computational overhead than inference. Inspired by that, we offer procedures in between: we increase the sample rate at calibration time while keeping it either one or very low during test time. This calibration-time sampling opens the possibility of reducing the size of the prediction sets. With a large enough test set (which is often the case in production), our Front-Loaded RCPs have the same computational complexity as the state of the art, while producing considerably smaller sets at larger radii.

8.1 Introduction

While calibration of uncertainty estimates is crucial in safety-critical domains (Guo et al., 2017), existing approaches typically incur substantial computational overhead and they also lack strong statistical guarantees. These computational costs arise both during training and at inference time, often requiring retraining and multiple forward passes. Instead, conformal prediction (CP) wraps around any black-box model and returns prediction *sets* that are guaranteed to cover the true answer with $1-\alpha$ adjustable probability. This guarantee requires a holdout calibration set and a score function that captures the agreement between the data and any potential answer (e.g. between each image and possible target labels in image classification). During the calibration (pre-processing) step, CP computes an acceptance threshold from the calibration set; during inference, it returns the prediction set consisting of all answers whose scores exceed this threshold. Assuming exchangeability between the test example and the calibration points, the resulting prediction set covers the true answer (or label) with

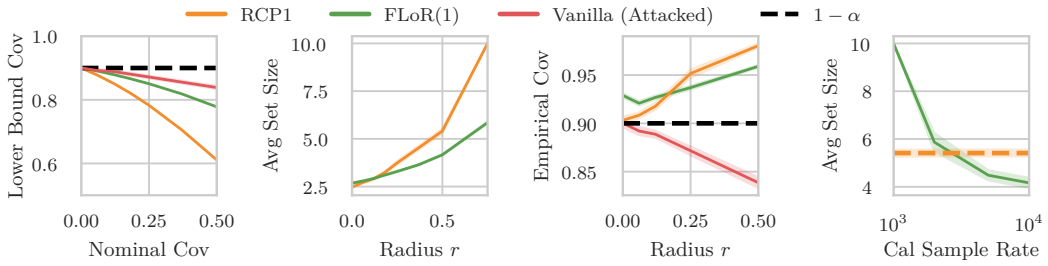

Figure 8.1: [Left] Worst-case empirical coverage (under adversarial attack) compared with the guaranteed lower bound coverage from RCP1 and Flor$^{(1)}$ (without accounting for worst-case noise, see § 8.5). [Middle left] Average set size of BinCP, RCP1, and Flor$^{(1)}$ across radii. [Middle right] Empirical coverage of methods under adversarial attacks. [Right] Average set size across various MC sampling budgets. Note that here RCP1 works with a single sample in both calibration and test (see § 8.5 for details).

$1 - \alpha$ probability.

In real-life scenarios, test and calibration instances can differ as in production the inputs can be noisy or adversarially perturbed. In that case, the exchangeability and coverage guarantee is shown to break severely under even a very small magnitude of perturbation in the input space. Robust CP (RCP) adapts results from the literature of adversarial robustness into the conformal prediction framework and extends the coverage guarantee to account for the worst case perturbations. In both vanilla and robust CP the coverage probability is guaranteed, therefore the race is over other features of the sets; e.g. average set size or "usability" (which we use in place of the commonly used "efficiency" not to be mistaken with the computational efficiency).

Usually, this robustness is achieved by bounding the changes in the score function – either the given score function, or redefining a new score that changes slowly. A promising approach for it is to use randomized smoothing as it works with any model through black-box access, it can be defined for variety of perturbation schemes (Yang et al., 2020), and its prediction sets remains usable for large radii. A downside is that it often requires an extensive Monte-Carlo sampling since it relies on estimating statistics (e.g. mean) of the score around the inputs (Yan et al., 2024). This substantially limits the practicality of smoothing RCP in real-world deployments, primarily because test-time inference is computationally expensive – the calibration is similarly inefficient however we only run it once as a pre-process.

Interestingly, Zargarbashi et al. (2025) show that smoothing can be integrated into CP while still requiring only a single forward pass per input (in both calibration and test). The key idea is to apply certified bounds not to the score itself, but directly to the coverage probabilities. By showing the convexity of the smoothing-based certified lower-bound function, they circumvent the need to estimate any intermediate statistics and instead directly certify the known target probability $1 - \alpha$. While this test-time speedup makes the method applicable to real-time settings, the average prediction set size cannot be further reduced. Consequently, when the computational budget is increased, RCP1 falls behind state-of-the-art sampling-heavy robust CP methods in terms of usability. Given this cost to set size trade-off, the open problem is to ask if there is way to keep the *test-time computational efficiency* while decreasing the set size.

We propose Flor$^{(1)}$ that bridges between the two endpoints of the mentioned trade-off following this argument; while the test-time inference speed is important, one can tolerate calibration-time computational overhead, as it is a one-time pre-process (after processing a number of test points, this computational overhead becomes negligible in total cost as shown in Fig. 8.2). Intuitively while RCP1 uses the convexity of the certified lower bound to directly apply it over the known value $1 - \alpha$ (Jensen's inequality over the coverage probability future test points, see § 8.3), we estimate them from calibration points through Monte Carlo (MC) sampling. Proposing front loaded RCP (Flor$^{(1)}$), we show that the coverage probability in calibration points can represent the test points; and with that, we use conformal risk control to estimate the worst-case expected coverage by averaging over worst-case coverage probability of calibration points. Our method outperforms RCP1 intuitively since Jensen's inequality make the setup unnecessarily conservative (see Fig. 8.1-right). Same as RCP1, our framework works with any binary classification certificate, and without any modification it can apply to regression as well.

As noted by Zargarbashi et al. (2025), a further limitation of single-sample smoothing-based RCPs is that they return inherently stochastic prediction sets. Meaning that for some test points the method may be unlucky and produce excessively large prediction

sets. In the single sample setup, this effect is unavoidable since additional resampling of the set will invalidate the theoretical guarantees. To remedy that we propose Flor$^{(k)}$ which directly accounts for test time majority voting over prediction set, and returns more deterministic sets under a very low test-time sample rate. At the setup (high calibration and very low test-time sample rate), Flor$^{(k)}$ also outperforms state of the art BinCP.

Compared to related works (see § F.2) our Flor$^{(1)}$ (same as RCP1) has the same computational complexity as verification or Lipschitz-based RCPs, while providing usable prediction sets up that are robust to one order of magnitude larger radii. Flor$^{(1)}$, and Flor$^{(k)}$ return considerably smaller prediction sets (see Fig. 8.3) compared to the state of the art RCP1, and BinCP under the same sampling setup.

8.2 Background

With $\mathcal{D}_{n+1} = \{x_i, y_i\}_{i=1}^{n+1}$ as an exchangeable dataset sampled from the data distribution $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$, and $\mathcal{D}_{\text{cal}}$ as the calibration set taking the first n data points, for clean x_{n+1} CP returns a prediction set with the following guarantee:

$$\Pr_{\mathcal{D}_{n+1}}[y_{n+1} \in C_0(x_{n+1})] \geq 1 - \alpha \tag{67}$$

Here $C_0(x_{n+1}) = \{y : s(x_{n+1}, y) \geq 1 - \lambda\}$ for any score function $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ capturing the agreement between x, and y[15], and the threshold $\lambda = 1 - \mathbb{Q}(\alpha; \{s(x_i, y_i) : (x_i, y_i) \in \mathcal{D}_{\text{cal}}\})$, where $\mathbb{Q}(\beta; \mathcal{A})$ denotes the $\lfloor \alpha - \frac{1}{n} \rfloor$ quantile of the set $\mathcal{A}$. The guarantee holds regardless of the choice of score function. The choice of score function reflects in secondary properties such as smaller average set size (usability). For classification, s can be the softmax outputs of the model.

Conformal risk control. We prove the robust guarantee in Flor$^{(1)}$, and Flor$^{(k)}$ through conformal risk control (CRC) framework (Angelopoulos et al., 2022).

Theorem 8.1 (Conformal Risk Control - rephrased)**.** *Let λ be a parameter (larger λ yields more conservative output), and $L_i : \Lambda \to (-\infty, B]$ for $i = 1, \ldots, n+1$ be exchangeable random*

[15]CP can equivalently be defined using non-conformity scores. Setups are equivalent by flipping the sign. Following prior works, we adopt the conformity formulation, as it aligns more naturally with intuition for classification.

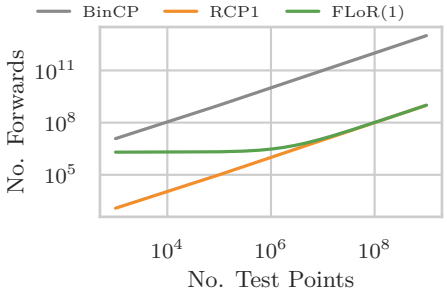

Figure 8.2: Total number of forward passes required by each method while increasing the number of expected test points. Note that both axis are log scaled which means that parallel lines differ by exponential factor. Flor$^{(1)}$ in calibration, and BinCP in both phases use 10^3 samples per point, and 500 calibration points.

functions. If (i) L_is are non-increasing right-continuous w.r.t. λ, (ii) for $\lambda_{\max} = \sup \Lambda$ we have $L_i(\lambda_{\max}) \leq \alpha$, and (iii) $\sup_\lambda L_i \leq B < \infty$, then we have:

$$\mathbb{E}_{\mathcal{D}_{n+1}}[L_{n+1}(\hat{\lambda})] \leq \alpha \tag{68}$$
$$\textit{for} \quad \hat{\lambda} = \inf\left\{\lambda : \frac{\sum_{i=1}^{n} L_i(\lambda)}{n+1} + \frac{B}{n+1} \leq \alpha\right\}$$

Conformal prediction itself is a special case of conformal risk control. In case that $B = 1$, by simplifying Eq. 68, we have $\hat{\lambda} = \inf\left\{\lambda : \sum_{i=1}^{n} L_i(\lambda) \leq \alpha(n+1) - 1\right\}$.

CP robust to a threat model. The guarantee in Eq. 67 relies on the exchangeability between the calibration $\mathcal{D}_{\text{cal}}$ and test point $\boldsymbol{x}_{n+1}$. An adversary (or natural noise) can break this exchangeability by adding imperceptible noise to the test point (evasion). Even a small perturbation can drastically decrease the coverage guarantee. Here we model the worst-case bounded noise (which we call adversarial perturbation) as the point with most miscoverage probability to be within a ball $\mathcal{B} : \mathcal{X} \to 2^{\mathcal{X}}$ around the clean input. While our approach is independent of the choice of $\mathcal{B}$, a common choice for images is the ℓ_2 norm ball: $\mathcal{B}_r(\boldsymbol{x}) = \{\tilde{\boldsymbol{x}} : \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq r\}$ where r is the radius of the perturbation. Robust conformal prediction (RCP) aims to extend the guarantee in Eq. 67 to any possible perturbation within $\mathcal{B}$:

$$\mathbb{E}_{\mathcal{D}_{n+1}}[\min_{\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}_r(\boldsymbol{x}_{n+1})} \Pr_u[y_{n+1} \in C_r(\tilde{\boldsymbol{x}}_{n+1}; u)]] \tag{69}$$
$$\geq \Pr_{\mathcal{D}_{n+1}, u}[y_{n+1} \in C_0(\boldsymbol{x}_{n+1}; u)] \geq 1 - \alpha$$

Here u encodes any randomness in defining the prediction set, i.e., randomness in the score function. For deterministic sets (i.e., no randomness in u), the inner probability is either 0 or 1; therefore, the guarantee in Eq. 69 reduces to:

$$\Pr_{\mathcal{D}_{n+1}}[y_{n+1} \in C_{\mathcal{B}}(\tilde{\boldsymbol{x}}_{n+1}), \forall \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})] \geq 1 - \alpha$$

We use C_0 and $C_{\mathcal{B}}$ (or C_r for ℓ_2 balls) to denote the vanilla and robust prediction sets, respectively. One way to attain robustness is to bound the changes in the score function within $\mathcal{B}$ — either by finding bounds for a given score function (through verification) or by designing a score function that changes slowly around the input. Through randomized smoothing, Gendler et al. (2021) (RSCP) propose using the smooth score $\hat{s}(\boldsymbol{x}, y) = \mathbb{E}_\epsilon[s(\boldsymbol{x} + \epsilon, y)]$, where ϵ is random noise drawn from a standard distribution (e.g., for images, one choice is isotropic Gaussian noise). Furthermore, Zargarbashi and Bojchevski (2025) (BinCP) use a quantile-based score $\hat{s}_q(\boldsymbol{x}, y) = \mathbb{Q}(\tau; s(\boldsymbol{x} + \epsilon, y))$ for some arbitrary $\tau \in (0, 1)$ to reduce the required sample rate: methods that define the score as a statistic over the smoothed original score require extensive Monte-Carlo sampling to estimate that statistic (mean, quantile, etc.), followed by a finite-sample correction; in the case of BinCP, the required sample rate is an order of magnitude lower by bounding over a Bernoulli variable instead of a continuous variable.

We define a robust CP based on certified bounds: the lower bound $\mathrm{c}^{\downarrow}(f, \mathcal{B}(\boldsymbol{x})) \leq \inf\{f(\boldsymbol{z}) : \boldsymbol{z} \in \mathcal{B}(\boldsymbol{x})\}$, and similarly $\mathrm{c}^{\uparrow}(\cdot, \cdot)$ as the upper bound (with sup). Therefore, for all $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$, we have $\mathrm{c}^{\downarrow}(s(\cdot, y), \mathcal{B}(\boldsymbol{x})) \leq s(\tilde{\boldsymbol{x}}, y) \leq \mathrm{c}^{\uparrow}(s(\cdot, y), \mathcal{B}(\boldsymbol{x}))$. For example, one way to define robust sets is test-time upper bounds over the score function (Zargarbashi et al., 2024): $C_{\mathcal{B}}(\boldsymbol{x}_{n+1}) = \{y : \mathrm{c}^{\uparrow}(s(\cdot, y_{n+1}), \mathcal{B}^{-1}(\boldsymbol{x}_{n+1})) \geq 1 - \lambda\}$. Here $\mathcal{B}^{-1}$ is the smallest set that contains the clean $\boldsymbol{x}$ for any perturbed $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$. For symmetric balls like ℓ_p we have $\mathcal{B}^{-1} = \mathcal{B}$. Another example is to set the score over randomly augmented

input $\hat{s}(\boldsymbol{x}, y) = s(\boldsymbol{x} + \boldsymbol{\epsilon}, y)$ and apply the certified lower bound directly on the coverage probability instead of the score function; i.e. $\beta := \Pr[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y) \geq q_\alpha]$ (Zargarbashi et al. (2025)-RCP1).

Bounds from randomized smoothing. Smoothing allows us to compute certified bounds for any black-box (model or score) function. A smoothing scheme $\psi : \mathcal{X} \to \mathcal{X}$ adds random noise (from a prespecified distribution) to the input — mapping it to a random nearby point. Intuitively, in the smooth classifier (or in our case, the randomized classifier), there is a large overlap between the push forward distributions for any two inputs within $\mathcal{B}$. This overlap enables to compute closed-form upper, or lower bounds regardless of the structure of the model, and the point $\boldsymbol{x}$. In almost all cases, internally the bound is found through optimization for any classifier that shares same statistics (e.g. mean) as the black-box function f, and for any pair of points that are r apart. As the bounds are independent of the value $\boldsymbol{x}$, and the function f, and only they are a function of the probability $p := \Pr_{\boldsymbol{\epsilon}}[f(\boldsymbol{x} + \boldsymbol{\epsilon}) = 1]$ we use the alternative notation $\mathrm{c}^{\downarrow}(p, \mathcal{B})$ instead of $\mathrm{c}^{\downarrow}(f, \mathcal{B}(\boldsymbol{x}))$.

$$\mathrm{c}^{\downarrow}(\beta, \mathcal{B}_r) \leq \min_{\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})} \Pr_{\boldsymbol{\epsilon}}[s(\tilde{\boldsymbol{x}} + \boldsymbol{\epsilon}) = 1] \quad \text{given} \quad \Pr_{\boldsymbol{\epsilon}}[s(\boldsymbol{x} + \boldsymbol{\epsilon}) = 1] = \beta \tag{70}$$

One common smoothing setup is isotropic Gaussian smoothing and ℓ_2 ball. For example, in case of image classification, we add $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma \boldsymbol{I})$ to each input $\boldsymbol{x}$. and the lower bound has a closed form solution of $\mathrm{c}^{\downarrow}(p, \mathcal{B}) = \Phi_\sigma(\Phi_\sigma^{-1}(p) - r)$ where Φ_σ is the CDF of the Gaussian distribution $\mathcal{N}(0, \sigma)$. Similarly, the upper bound $\mathrm{c}^{\uparrow}(\beta, \mathcal{B}_r)$ can be defined by replacing the min with max in Eq. 70 and in this specific setup the close form is the same up to changing $-r$ with $+r$. Furthermore, for any smoothing function and threat model, Zargarbashi et al. (2025) re-derives the original results from (Lee et al., 2019) in terms of lower bound over probabilities.

8.3 Calibration-Intensive Robust CP

We first recall RCP1 as it shares a similar logic.

Recall: RCP1 algorithm. RCP1 (Zargarbashi et al., 2025) follows the standard conformal prediction pipeline with a single modification: during both calibration and

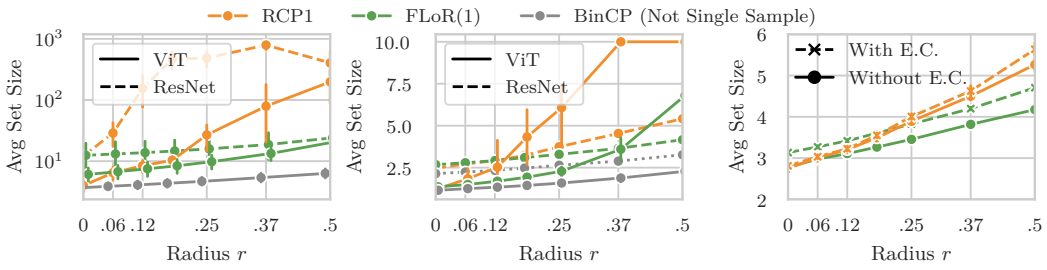

Figure 8.3: Comparison of set size across various radii for the [Left] ImageNet dataset, and [middle] CIFAR-10 dataset both with $\sigma = 0.5$, over both ViT, and ResNet models. Note that the BinCP is drawn as an ideal baseline, while it is not comparable to other methods since in the test time, it sill requires MC sampling. Results for adaptive prediction set (APS) with and without finite sample correction (ResNet).

inference, the input to the score function is augmented with random noise $\boldsymbol{\epsilon}$; i.e. $\hat{s}(\boldsymbol{x}, y) = s(\boldsymbol{x} + \boldsymbol{\epsilon}, y)$. The i.i.d. added noise does not break the exchangeability, and therefore the coverage guarantee is $1 - \alpha$ for clean inputs (in expectation). This probability reduces to $\mathrm{c}^{\downarrow}(1 - \alpha, \mathcal{B})$ under worst-case perturbations within $\mathcal{B}$. To achieve a coverage above $1 - \alpha$ in the worst case, calibration is performed at level $\tilde{\alpha}$ such that $\mathrm{c}^{\downarrow}(1 - \tilde{\alpha}, \mathcal{B}) = 1 - \alpha$. In many cases, we can equivalently set $1 - \tilde{\alpha} = \mathrm{c}^{\uparrow}(1 - \alpha, \mathcal{B}^{-1})$.

Under the hood, RCP1 aims to certify the lower bound for $\tilde{\beta}_{n+1} = \Pr_{\boldsymbol{\epsilon}}[y_{n+1} \in C_{\mathcal{B}}(\tilde{\boldsymbol{x}}_{n+1})] \geq \mathrm{c}^{\downarrow}(\beta_{n+1}, \mathcal{B})$. However, since β_{n+1} is unknown and the method aims to avoid Monte-Carlo estimation, it leverages the convexity of the certified lower bound function to directly apply the bound to $1 - \alpha$, which corresponds to the expected value of β_{n+1}.

RCP1 is unnecessarily conservative. While the convexity argument (i.e., $\mathrm{c}^{\downarrow}(1-\alpha, \mathcal{B}) \leq \mathbb{E}_{\mathcal{D}_{n+1}}[\mathrm{c}^{\downarrow}(\beta_{n+1}, \mathcal{B})]$) reduces the computational cost at both calibration and test time, it makes the guarantee very conservative – there is a considerable difference between $\mathbb{E}_{\mathcal{D}_{n+1}}[\mathrm{c}^{\downarrow}(\beta_{n+1}, \mathcal{B})]$, and $\mathrm{c}^{\downarrow}(\mathbb{E}_{\mathcal{D}_{n+1}}[\beta_{n+1}], \mathcal{B})$ which is also empirically shown in Fig. 8.1-left. On the other hand, computational cost is considerably less critical during calibration, since it is a one-time preprocessing step, whereas the dominant cost arises at inference time due to the potentially unbounded number of test examples. We propose to directly estimate $\mathbb{E}_{\mathcal{D}_{n+1}}[\mathrm{c}^{\downarrow}(\beta_{n+1}, \mathcal{B})]$ via Monte Carlo sampling at calibration time. Importantly, our method still requires only a single sample at inference time, thereby keeping the dominant computational cost low. In Fig. 8.2, we compare the cost RCP1 and Flor$^{(1)}$ in number of requrired forward passes. It clearly shows that the computation costs of the two methods converge by incraesing the number of test points – the bias in the computational cost becomes negligible over time. It means that RCP1 yields coverage guarantee that is unnecessarily more conservative, and in return brings a computational efficiency that becomes inconsiderable over time.

Flor$^{(1)}$. First we provide a high level view of the proof. Following Zargarbashi et al. (2025), the noise augmentation over inputs $\hat{s}(\boldsymbol{x}, y) = s(\boldsymbol{x} + \boldsymbol{\epsilon}, y)$ does not break the exchangeability and therefore, running vanilla CP over the new scores $\hat{s}$ results in $1 - \alpha$ coverage. As we are introducing randomness adding $\boldsymbol{\epsilon}$, the coverage of each point becomes a Bernoulli random variable (dependent to $\boldsymbol{\epsilon}$) with success probability $\beta_{n+1} = \Pr_{\boldsymbol{\epsilon}}[s(\boldsymbol{x}_{n+1} + \boldsymbol{\epsilon}, y_{n+1}) \geq 1 - \lambda]$, rather than a deterministic event (note that in some score functions like APS, this variable is already non-deterministic). Moving from $\boldsymbol{x}_{n+1}$ to $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$, the worst-case coverage for $\tilde{\boldsymbol{x}}_{n+1}$ bounded by $\tilde{\beta}_{n+1} = \mathrm{c}^{\downarrow}(\beta_{n+1}, \mathcal{B})$.

As we avoid estimating β_{n+1} (since we aim for single sample inference), we apply conformal risk control. Let

$$L_i(\lambda) = \max_{\tilde{\boldsymbol{x}}_i \in \mathcal{B}(\boldsymbol{x}_i)} \Pr_{\boldsymbol{\epsilon}}[s(\boldsymbol{x}_i, y_i) \leq 1 - \lambda] \tag{71}$$

Here $L_i(\lambda)$ is the risk function capturing the probability of miscoverage for the worst $\tilde{\boldsymbol{x}}_i$ with the threshold $1 - \lambda$. From the certified lower bound function we have $1 - \mathrm{c}^{\downarrow}(\beta_i(\lambda), \mathcal{B}) \geq L_i(\lambda)$. Note that β_i is implicitly a function of λ. Replacing the risk $L_i(\lambda)$ with $1 - \mathrm{c}^{\downarrow}(\beta_i(\lambda), \mathcal{B})$ only increases the value λ making the setup more conservative. Through conformal risk control (as we show further) we find a $\lambda^{\star}$ that guarantees $\mathbb{E}_{\mathcal{D}}[\tilde{\beta}_{n+1}] \geq 1 - \alpha$. Sampling one instance from each realization of $\tilde{\beta}_{n+1}$ (over the test points) results in expected worst-case coverage of $1 - \alpha$.

To lower bound $\tilde{\beta}_i$, we first estimate β_i via Monte Carlo sampling. Using m samples per input $\boldsymbol{x}_i$, we compute $\hat{\beta}_i = \frac{1}{m} \sum_{j=1}^{m} \mathbb{I}[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}_j, y_i) \geq 1 - \lambda]$. Via Clopper-Pearson

inequality (Clopper and Pearson, 1934), we obtain a lower confidence bound $\underline{\beta}_i$ such that $\Pr\left[\beta_i \geq \underline{\beta}_i\right] \geq 1 - \delta/|\mathcal{D}_{\text{cal}}|$, which implies a union bound for success probability to be $1-\delta$ over all calibration points. Under worst-case perturbations within $\mathcal{B}$, the coverage probability at $\tilde{\boldsymbol{x}}_i$ is lower bounded by $\mathrm{c}^{\downarrow}(\underline{\beta}_i, \mathcal{B}) \leq \tilde{\beta}_i$. By replacing $1 - \mathrm{c}^{\downarrow}(\underline{\beta}_i, \mathcal{B})$ with $L_i(\lambda)$ and tuning for $1 - \alpha + \delta$ coverage, we can secure the worst case $1-\alpha$ coverage at test time.

All the above arguments lead to the following result:

Proposition 8.2. *Given clean calibration points $Z_1, \ldots, Z_n$ ($Z_i = (X_i, Y_i)$), and a potentially perturbed $\tilde{Z}_{n+1} = (\tilde{X}_{n+1}, Y_{n+1})$, for $\tilde{X}_{n+1} \in \mathcal{B}(X_{n+1})$, let $E_i : i \in [n+1] \sim \psi$ from a predefined smoothing scheme ψ, we have*

$$\Pr_{E_{n+1}, \mathcal{D}_{n+1}}[Y_{n+1} \in C(\tilde{X}_{n+1})] \geq 1 - \alpha$$

For $C(\tilde{X}_{n+1}) = \{y : s(\tilde{X}_{n+1} + E_{n+1}, y) \geq 1 - \lambda\}$ where $1-\lambda$ is defined as following: Let $\beta_i(\lambda) = \Pr_{E_i}[s(X_i + E_i, Y_i) \geq -\lambda]$ and $\underline{\beta}_i \leq \beta_i$ with $1 - \delta/|\mathcal{D}_{\text{cal}}|$ probability. Then

$$\lambda = 1 - \sup\{\lambda' : \frac{1}{n+1} \sum_{i=1}^{n} \mathrm{c}^{\downarrow}(\underline{\beta}_i(\lambda'), \mathcal{B}) \geq 1 - \alpha + \delta\}$$

Finding λ in practice. In general, for conformal risk control, the variable λ either admits a closed-form solution (like the special case of conformal prediction) or can be computed via binary search. In our setup, where $\beta_i(\lambda)$ must be estimated from MC-samples, a naive approach would be to resample all calibration points at each of the $O(\log n)$ steps of the binary search – we cannot reuse the same samples across iterations, as this corresponds to multiple hypothesis testing over $\beta_i(\lambda)$ for different values of λ. To circumvent this computational cost, we split our sampling budget into a certification and a considerably smaller tuning budget; i.e. $m = m_{\text{cert}} + m_{\text{tune}}$. We first compute λ by running binary search using m_{tune} samples, and then validate the resulting candidate using the m_{cert} budget. During the tuning step, we ensure that the selected λ ultimately results in a risk below α when validated over the larger sample rate. In other words, during tuning, we account for finite sample correction, and the certified lower bounds.

During binary search, for any trial threshold λ, we estimate the coverage probability per each calibration point $\hat{\beta}_i = \frac{1}{m_{\text{tune}}} \sum_{j=1}^{m_{\text{tune}}} \mathbb{I}[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}_j, y_i) \geq 1 - \lambda]$, and compute the confidence interval lower bound $\underline{\beta}_i$ through Clopper-Pearson intervals with confidence $1 - \delta/|\mathcal{D}_{\text{cal}}|$. Notably, as we later validate the λ, computing the confidence interval is not necessary. It is only to simulate how the interval length can effect the validation step, therefore we imitate the actual certification step, where the confidence lower bound is computed over m_{cert} samples per calibration point – i.e. $\hat{\beta}_i \cdot m_{\text{cert}}$ successes over m_{cert} samples. In order to ensure that the resulting λ passes the validation phase, we also add an extra bias to the binary search by setting the coverage probability to $1 - \alpha + \delta + \delta_0$ where δ_0 is a very small value accounting for potential mismatch between the simulated, and the actual confidence intervals. Finally, the decision rule for binary search is

$$1/(n+1) \sum_{i=1}^{n} \mathrm{c}^{\downarrow}(\underline{\beta}_i, \mathcal{B}) \geq 1 - \alpha + \delta + \kappa$$

Finding the best value for λ, we validate it by estimating $\hat{\beta}_i$ over m_{cert} samples, and computing the lower bound $\underline{\beta}_i$, this time using the actual number of successes. The final

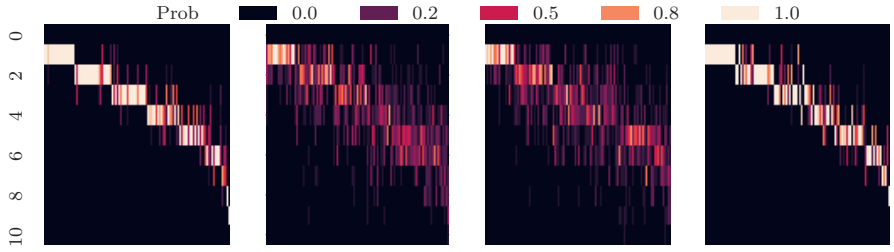

Figure 8.4: Distribution of set sizes across inputs. The x-axis shows various inputs, and the y-axis shows different set sizes. [From left to right] plots are depicting BinCP, RCP1, Flor$^{(1)}$, and Flor$^{(k)}$ all over the same evaluation subset of test points of CIFAR-10 dataset. Here $r = 0.25$, $\sigma = 0.5$, and we used ResNet model. In all plots, inputs are sorted with the same reference. Here BinCP and Flor$^{(k)}$ are both calibrated with 10^4 MC samples, and tested with 101 samples.

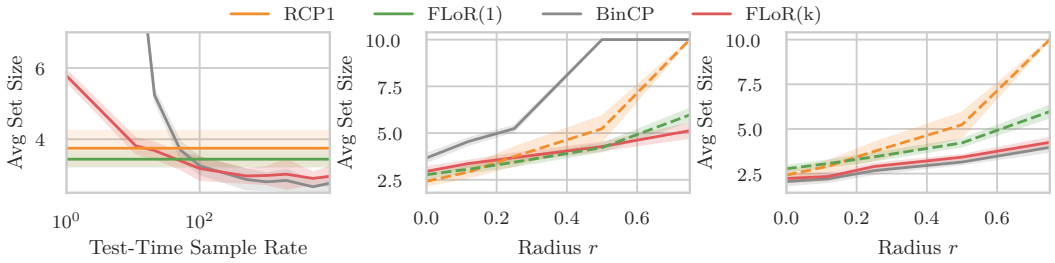

Figure 8.5: [Left] Proportion of sets with size ≤ 1, 3, and 5, for ImageNet dataset, and ResNet model with $\sigma = 0.5$. [Middle] On the same setup, the average set size for both methods across differnt model and smoothing σ's. [Right] Average set size by increasing sample rate (here $r = 0.5$), and the results are over CIFAR-10 dataset and ResNet model.

guaranteed coverage is computed higher than $1/(n+1)\sum_{i=1}^{n} \mathrm{c}^{\downarrow}[\underline{\beta}_i, \mathcal{B}] - \delta$. We expect this value to be above $1-\alpha$. In case the validation fails to ensure that, we have to repeat the process once again (maybe with a higher δ_0).

Why Flor$^{(1)}$ yields better sets. As shown in Fig. 8.3, out Flor$^{(1)}$ produces smaller sets while maintaining the same robust guarantee as RCP1. Both RCP1 and Flor$^{(1)}$ share the same definition of the prediction set, and the same objective for calibration. The key difference arises on how to aim for that objective. RCP1 first proves the $E_{\mathcal{D}}[\beta_{n+1}] \geq 1-\alpha$, and uses the Jensen's inequlity to directly lower bound the objective with $\mathrm{c}^{\downarrow}(1-\alpha, \mathcal{B})$. Instead Flor$^{(1)}$ finds this value by approximating worst case $\tilde{\beta}_{n+1}$ from calibration points. Therefore their difference boils down to the distance of the Jensen's lower bound with the real expectation. As Jensen's inequality always results in a lower value compared to the actual push-forward expectation (i.e. $f(E[X]) \leq E[f(X)]$ for convex f), asymptotically, Flor$^{(1)}$ always results in smaller prediction sets compared to RCP1. There is only an exception in radii close to 0 which is due to conservativeness of δ_0 and the confidence lower bounds.

8.4 Reducing Stochasticity

Both RCP1 and Flor$^{(1)}$ produce inherently stochastic prediction sets – the indicator $\mathbb{I}[s(\boldsymbol{x}_{n+1} + \boldsymbol{\epsilon}, y) \geq 1-\lambda]$ remains random even after fixing $\boldsymbol{x}_{n+1}$, y, and λ. Methods such

as BinCP exhibit comparatively lower variability by definition of their score function. Fig. 8.4 clearly shows this stochasticity reflected in the variance of prediction set sizes across inputs. As in the single sample regime this stochasticity is inevitable, we need to sample more from $\boldsymbol{x}_{n+1} + \boldsymbol{\epsilon}$ to reduce the variance. Our goal is to achieve the smallest set size using a very low test-time sample rate. One approach is to use BinCP with different sample rates during calibration (higher) and test (lower). Alternatively, we can aggregate prediction sets from repeated sampling through different operators: *union* preserves coverage monotonicity, but it rapidly inflates the prediction set. *Majority voting* stabilizes set size but provably reduces guaranteed coverage to $1 - 2\alpha$ (Angelopoulos et al., 2024). We instead propose Flor$^{(k)}$ through the same framework as Flor$^{(1)}$ which is aware of test time resampling, and majority vote aggregation which results in robust coverage guarantees, and returns set sizes that are smaller than BinCP for the very-low sampling regime (e.g. 21 to 101 samples).

Flor$^{(k)}$. While majority vote reduces the guarantee to $1 - 2\alpha$, we overcome this issue by calibrating with an objective aware of a majority-vote test-time aggregator. Let $\Phi_{\text{bin}}(t, k, p)$ denote the binomial CDF. Our method identifies positive values (k, α_0, β_0) that satisfy the following:

$$(1 - \alpha + \alpha_0)\Big(1 - \Phi_{\text{bin}}\Big(\tfrac{k-1}{2}, k, \tfrac{1}{2} + \beta_0\Big)\Big) \overset{?}{\geq} 1 - \alpha \tag{72}$$

Note that fixing two, the third variable always has a solution. Following proposition holds.

Proposition 8.3. *With an odd integer $k \geq 1$, positive offsets $(\alpha_0, \beta_0) \geq 0$ satisfying Eq. 72, and a smoothing distribution ψ. Let $(Z_i)_{i=1}^{n+1}$ with $Z_i = (X_i, Y_i)$ be exchangeable, and let $\mathcal{B}_r(x)$ be the perturbation ball. For each $i \in [n]$ and threshold λ, define the (robust) inclusion probability*

$$\tilde{\beta}_i(\lambda) \;:=\; \min_{\tilde{X}_i \in \mathcal{B}_r(X_i)} \Pr_{E_i \sim \psi}\big[s(\tilde{X}_i + E_i,\, Y_i) \geq 1 - \lambda\big].$$

estimated in same way as Proposition 8.2, resulting in lower confidence bound $\underline{\beta}_i(\lambda)$ satisfying

$$\Pr\big[\underline{\beta}_i(\lambda) \leq \beta_i(\lambda)\big] \;\geq\; 1 - \delta/|\mathcal{D}_{\text{cal}}|.$$

consider the risk function

$$L_i(\lambda) \;:=\; \mathbb{I}\Big[\mathrm{c}^{\downarrow}(\underline{\beta}_i(\lambda), \mathcal{B}) \leq \tfrac{1}{2} + \beta_0\Big], \qquad i \in [n], \tag{73}$$

if λ satisfies the following ineuqality

$$\frac{1}{n+1}\left(\sum_{i=1}^{n} L_i(\hat{\lambda}) + 1\right) \;\leq\; \alpha - \alpha_0 - \delta.$$

The majority vote prediction set $C^{(k)}$ defined as following

$$C^{(k)}(\boldsymbol{x}_{n+1}) = \{y : \frac{1}{k}\sum_{t=1}^{k} \mathbb{I}\big[s(\boldsymbol{x}_{n+1} + \boldsymbol{\epsilon}_t,\, y) \geq 1 - \lambda\big] > \frac{1}{2}\}$$

has the robust $1 - \alpha$ guarantee.

Proof. Consider the following risk function

$$L'_i(\lambda) = \mathbb{I}[\min_{\tilde{\boldsymbol{x}}_i \in \mathcal{B}(\boldsymbol{x}_i)} \Pr_{\boldsymbol{\epsilon}}[s(\tilde{\boldsymbol{x}}_i + \boldsymbol{\epsilon}, y_i) \geq 1 - \lambda] \leq \frac{1}{2} + \beta_0]$$

Same as Proposition 8.2 this risk monotonically decreasing w.r.t. λ, right continuous, and upper bounded by 1. By the definition of the certified lower bound function we have $L_i(\lambda) \geq L'_i(\lambda)$ for L_i defined in Eq. 73. Therefore calibrating with L_i through risk control, yields a λ that satisfies $E_{\mathcal{D}_{n+1}}[L_{n+1}(\lambda)] \leq \alpha - \alpha_0$ which implies the following

$$\Pr_{\mathcal{D}}\left[\beta_{n+1}(\lambda) \geq \tfrac{1}{2} + \beta_0\right] \;\geq\; 1 - \alpha + \alpha_0.$$

Note that

$$\Pr_{\substack{\mathcal{D}_{n+1} \\ \boldsymbol{\epsilon}_{n+1,1:k}}}\left[y_{n+1} \in C^{(k)}(\boldsymbol{x}_{n+1})\right] \geq \Pr_{\mathcal{D}}\left[\beta_{n+1}(\lambda) \geq \tfrac{1}{2} + \beta_0\right] \times$$

$$\Pr_{\boldsymbol{\epsilon}_{n+1,1:k}}\left[y_{n+1} \in C^{(k)}(\boldsymbol{x}_{n+1}) \mid \beta_{n+1}(\lambda) \geq \frac{1}{2} + \beta_0\right]$$

where the second term captures the probability of majority vote over k samples including the true label, while the probability of including it is higher than $\frac{1}{2} + \beta_0$. We have

$$\begin{aligned}
&\Pr_{\boldsymbol{\epsilon}_{n+1,1:k}}\left[y_{n+1} \in C^{(k)}(\boldsymbol{x}_{n+1}) \mid \beta_{n+1}(\lambda) \geq \frac{1}{2} + \beta_0\right] \\
&\qquad = \Pr_{\boldsymbol{\epsilon}_{n+1,1:k}}\left[\mathrm{Bin}\left(k, \tfrac{1}{2} + \beta_0\right) \geq \tfrac{k+1}{2}\right] \\
&\qquad \geq \left(1 - \Phi_{\mathrm{bin}}\left(\tfrac{k-1}{2}, k, \tfrac{1}{2} + \beta_0\right)\right)
\end{aligned}$$

Therefore combining the two terms with (k, α_0, β_0) satisfying Eq. 72 we have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}_{n+1}}\left[\Pr_{\boldsymbol{\epsilon}_{1:k}}\left(y_{n+1} \in C^{(k)}(\boldsymbol{x}_{n+1})\right)\right] \\
&\geq (1 - \alpha + \alpha_0)\left(1 - \Phi_{\mathrm{bin}}\left(\tfrac{k-1}{2}, k, \tfrac{1}{2} + \beta_0\right)\right) \geq 1 - \alpha.
\end{aligned}$$

□

8.5 Experiments

Via empirical evaluation we show that (i) Flor$^{(1)}$ provides robust guarantee: while theoretically guaranteed, we also empirically show that the coverage of Flor$^{(1)}$ remains

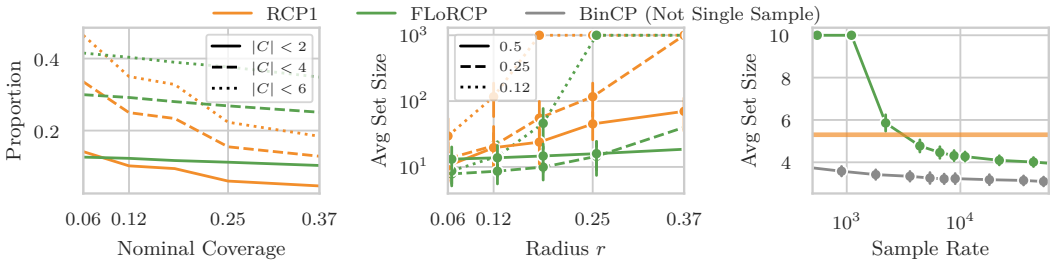

Figure 8.6: [Left] Proportion of sets with size ≤ 1, 3, and 5, for ImageNet dataset, and ResNet model with $\sigma = 0.5$. [Middle] On the same setup, the average set size for both methods across differnt model and smoothing σ's. [Right] Average set size by increasing sample rate (here $r = 0.5$), and the results are over CIFAR-10 dataset and ResNet model.

above the nominal $1-\alpha$ in presence of adversarial attack. (ii) Flor$^{(1)}$ returns smaller sets compared to prior state of the art RCP1: We show that as we increase the calibration sample rate, the set size in Flor$^{(1)}$ decreases. This is while due to its nature, RCP1 can not get more efficient by increasing the computational resources. (iii) Flor$^{(k)}$ reduces the stochasticity of the prediction set to a level equal with BinCP while under a very low sample rate, it outperforms BinCP in set size.

Our comparison baselines are RCP1 (Zargarbashi et al., 2025), and BinCP Zargarbashi and Bojchevski (2025). There is a tradeoff between the two methods: RCP1 works with a single noise augmented sample per input (which works with roughly the same computation time as normal model forward) while it returns relatively larger sets. On the oposite side of the trade-off, BinCP returns smaller sets while it requires extensive Monte-Carlo sampling (both at the calibration and test time) – to outperform RCP1 BinCP needs roughly between 70 - 150 MC samples per each input. Flor$^{(1)}$ stays in the middle of this trade-off, but close to RCP1. It works with single sample at the test time (where the actual load is), while it returns smaller set size at the cost of Monte Carlo sampling only during calibration time. Note that while we report the results for BinCP in some plots, we only bring those results as a comparison to best possible setup, while with the extensive test time sampling of BinCP, *the two methods are not comparable*.

Following Zargarbashi et al. (2025), we employ two distinct classification pipelines. (i) The first is a computationally lightweight configuration: we rely on ResNet models trained with noise augmentation from Cohen et al. (2019). With the relatively small model size, operating at large sampling rates—while computationally inefficient—remains feasible in practice. (ii) We also study a second, more computationally demanding pipeline introduced by Carlini et al. (2022). Here, each input is first denoised using a diffusion model and subsequently classified by a vision transformer. In this setup it is unrealistically expensive to apply Monte-Carlo sampling during test-time, which is a shortcoming of BinCP – this setup exclusively fits RCP1, and Flor$^{(1)}$. The calibration-time MC sampling in Flor$^{(1)}$ runs only once prior to deployment which is a tolerable cost.

Valid coverage guarantees. Same as previous robust CP methods, we show that Flor$^{(1)}$ satisfies the guarantee in Eq. 69, both theoretically (see § 8.3) and empirically (see Fig. 8.1-third from right). Both methods remain above $1-\alpha$. Note that the adversarial attack is aware of randomized smoothing which makes it even stronger than the conventional PGD.

Conservativeness gap. As long as the lower bound coverage is provided, we prefer the empirical coverage to be closer to $1-\alpha$, since increasing the coverage naturally results in inflating the prediction sets. Since RCP1 shortcuts the estimation of β_{n+1} values (see § 8.3) using the convexity of certified lower-bound, it becomes unnecessary more conservative. We remedy that by directly estimating the worst case coverage from the calibration points. In Fig. 8.1-right, show that Flor$^{(1)}$ is significantly less conservative and closer to the actual empirical value under adversarial attack. There we do not account for the perturbation and directly query on the worst case guaranteed coverage.

Average Set Size. In Fig. 8.3 we show that overall, Flor$^{(1)}$ results in smaller prediction sets for both ImageNet and CIFAR-10 datasets. As we increase the radius r, Flor$^{(1)}$ outperforms RCP1 by higher margins. The effect is consistent across the efficient (ResNet) and computationally expensive (diffusion + ViT) setups. Notably ViT shows a significantly better performance, for the ImageNet dataset. Intuitively we do not expect

Flor$^{(1)}$ to outperform BinCP as it also benefits from expensive MC sampling during the test time. We also report the proportion of small sets ($|C(x_{n+1})| \leq 1$, 3, and 5) for the ImageNet dataset in Fig. 8.6-left. For test-time sampling, we compare our Flor$^{(k)}$ with BinCP, both calibrated with 10^4 samples and allowed for various sample-rates from 21 to 9001. From the motivation of the work, clearly we favor the setup with very small test-time sample rate, and in that setup Flor$^{(k)}$ outperforms BinCP. We also compare the Flor$^{(1)}$ in set size using the adaptive prediction sets (APS) score function from Romano et al. (2020). We discuss the reasons why RCP1 performs better at smaller radii in the following.

Discussion for smaller radii r. Setting aside the finite sample correction and accounting for small discrepancies between tuning and validation phases (via δ_0), by definition, Flor$^{(1)}$ performs better than RCP1 for any radius. In practice these two factors slightly affect the set size resulting in the performance of RCP1 being better for very small radii. By increasing the radius r, this effect becomes negligible. To highlight this, we also compare the set size of APS score with and without these two factors (see Fig. 8.3-right). Notably by disabling finite sample correction (considering asymptotically valid setup), we see that Flor$^{(1)}$ outperforms RCP1 across all radii.

Higher sample-rates. In Fig. 8.6-right we show how increasing the calibration-time sample rate can help the set size efficiency. While reporeted, note that again BinCP is not comparable with Flor$^{(1)}$, as by using the same computation for BinCP we can run Flor$^{(1)}$ with orders of magnitude higher sample rate; e.g. $|\mathcal{D}_{\text{cal}}| = 200$, only after inference over 50,000 test point, the computational cost of Flor$^{(1)}$ with the highest experimented sample rate (60,000) becomes lower than BinCP showing the same set size at a significantly low sample rate (250). Therefore, here, the x axis does not represent the same thing.

Effect of σ. Consistent with all previous smoothing-based RCP approaches, by increasing the smoothing σ, the set size slightly increases for all radii, while the slope of the inflation for larger r's remains steady.

8.6 Conclusion

We introduced Flor$^{(1)}$, a front-loaded robust conformal prediction framework: keeping the prediction time as fast as the model inference (like SOTA), by using more computational power during pre-processing (calibration) we reduce the average set size and maintain the same certified robustness guarantee. Our approach has the advantages of the both ends in the trade-off between computational efficiency (single sample RCP with relatively larger set sizes) and set-size efficiency (through expensive MC sampling). While in Flor$^{(1)}$, and other smoothing-based single sample (RCP1) the prediction set is random, we proposed Flor$^{(k)}$ which under a very low test-time sampling budget returns more deterministic sets still with set size below the state of the art with same sampling budget.

9 Conclusion, Limitations, and Future Directions

This dissertation studied how to make conformal prediction (CP) more reliable in two settings: graph-structured data with dependency, and adversarially perturbed data or data with worst-case noise. For the first direction, the study is focused on both understanding the theoretical conditions for CP validity, and how a similarity network can improve the uncertainty scores, and align them with the ground truth label distribution. For the second direction, the objective was not only to preserve validity, but also to make the framework practical in terms of computational cost.

9.1 Main Limitations, Follow-ups, and Lessons Learned

Limitation 1: Marginal coverage remains the dominant guarantee. All proposed methods in this thesis, as in mainstream CP, target marginal validity. This leaves open the possibility that specific subgroups or covariate regions are under-covered even when marginal coverage is maintained. In practice, this issue can be critical whenever failures concentrate on hard modes of the data distribution. In some parts of the thesis we report the conditional coverage; empirically, DAPS, and all robust CP methods show improved conditional coverage, but in the theoretical analysis, we do not provide any conditional coverage guarantee.

Through the lens of adversarial attacks, a potential future work is to solve the problem of searching for features or subgroups with low coverage; we can define adversarial search for worst-slice coverage. Such search can help us to train a model that results in a better coverage in that specific subgroup. Doing this iteratively can help us to improve the overall conditional coverage, and not only the marginal one.

Limitation 2: DAPS can be also applied on i.i.d. data, but the choice of similarity is not straightforward. DAPS shows that diffusion can improve set-size efficiency under homophily, given the graph structure. For us, it remains an open question whether we can build a similarity graph for i.i.d. data that captures uncertainty-relevant structure? We tried some naive approaches, such as building a kNN graph from penultimate-layer embeddings in image models, but it did not provide the expected improvements. This suggests that the error in similarities from this approach might be very correlated with the model's confidence. An interesting future direction is to design similarity metrics that are more complementary to the model's confidence, and can capture local calibration errors. Such kernel can both be used to improve the set size efficiency of CP, and to improve the approximate conditional coverage of the method.

Limitation 3: Exchangeability assumptions are restrictive in evolving graphs. The inductive framework addresses node/edge exchangeable sequences, but real graph streams are often governed by temporally structured, non-exchangeable mechanisms. Moreover, inductive GNN settings exhibit a distinctive difficulty that is not present in i.i.d. data like time-series: gradual shifts in the data distribution, when applied on graphs can shift both calibration and test scores jointly.

Consider this intuitive example: an adversary takes an exchangeable node-inducive sequence, and by only permuting the nodes, it can make the empirical coverage of the method to drop significantly. This however does not happen for i.i.d. data – if the original permutation has marginal coverage of $1 - \alpha$, then any permutation of the data will also have the same marginal coverage mainly because the calibration scores remain the same. This highlights that the beyond-exchangeability setting is

more challenging for graphs, and also opens an interesting line of research to design adversarial permutation search for worst-case in dependency structures.

Limitation 4: Threat models remain idealized. Same as other adversarial robustness works, the robust CP line in this thesis largely follows norm-bounded perturbation sets. These are mathematically convenient for certified analysis, but they can be mismatched to practical perturbation mechanisms. For images, realistic corruptions and semantic changes are not fully captured by a fixed ℓ_p ball. Not all imperceptible perturbations are equally likely to be within a small ℓ_p radius. While our methods are designed to be agnostic to the specific perturbation ball, and smoothing scheme, still the problem of defining a realistic threat model remains open and very important.

Limitation 5: Robustness-efficiency trade-offs persist under repeated updates. Our proposed Flor$^{(1)}$ (see § 8) can achieve near-inference-time robust prediction, and heavy using single time calibration, it can reduce the set size. However, in case that the model is updated frequently, the cost of recalibration can dominate the overall cost of robust CP. This means that in these settings, the robustness-efficiency trade-off remains between RCP1 (§ 7), and BinCP (§ 6), which means computationally cheaper with unimprovable set size, or more expensive with smaller set size. The question of how to adapt recalibrations efficiently without the need of expensive MC sampling remains open for robust CP approaches.

Lesson from the full trajectory. A key lesson which is learned from BinCP, and specially RCP1, is that a combination of two qualities, like in our case distribution-free coverage guarantee, and robustness can be achieved with a careful redesign of the full pipeline, and not only by adding one component to the existing one. CAS only uses existing robustness certificates, to achieve robustness in CP without major redesign of the CP calibration pipeline, but turns out that the probabilistic nature of CP guarantee can easily allow us to circumvent the need of extensive MC sampling which is present in all existing randomized smoothing based robustness certificates.

9.2 Future Directions

1. Generalizing beyond classification. This thesis already touches robust risk control and regression extensions, but The framework can be extended to other tasks as well. For instance, a specific direction is to use the insight from RCP1, and Flor$^{(1)}$ to design computational cheaper robustness certificates for the in-domain input data.

2. Free-form language generation and LLMs as a major gap. An explicit shortcoming of this dissertation is that it does not cover free-form language outputs. This is a significant boundary of scope, and a primary future direction. For LLM systems, uncertainty is needed at semantic level – a single statement can have multiple phrasings. Key open problems include: defining conformity scores for generative outputs, handling semantic equivalence classes of acceptable answers, and designing robustness notions that accounts for rephrasing, and semantic similarity.

Importantly, while adapting conformal prediction in LLMs, an important consideration is conditional coverage. In classification settings we are assuming every user of the model is inferencing on the similar distribution. This assumption can be wildly violated in LLMs, where users can have very different distributions of inputs. One uses LLM

to summarize normal text, and the other uses it to summarize scientific papers, and calibrating over both of these distributions jointly can lead to an uneven coverage over the two distributions, and therefore different users can have very different reliability from the same framework.

9.3 Retrospective Insights

Looking back, several insights became clearer only after the full sequence of projects.

First, robustness and uncertainty quantification are closely related but not identical goals. Improving one does not automatically improve the other unless the design explicitly couples them. In other words, we can design adversarial examples that are misclassified with high confidence under existing uncertainty quantification approaches. This is in contrast with the core principle of uncertainty quantification, as it should decrease confidence when inputs are not coming from the same distribution as the training data.

Second, while there are mathematically elegant approaches providing guarantees over the models behavior (e.g. randomized smoothing), their application can be computationally prohibitive. It is not an unrealistic assumption, that in many regimes we expect the model to perform well *with a high probability*. Such probabilistic view, can allow us to redesign the existing approaches in a computationally efficient way, while still providing the required guarantees.

9.4 Final Remarks

This dissertation advances distribution-free uncertainty quantification in two difficult regimes: structured dependence and adversarial perturbation. On graphs, it clarifies conditions for validity and introduces structure-aware methods that improve efficiency. Under worst-case perturbations, it develops robust CP methods that progressively improve the robustness-efficiency frontier, culminating in near-inference-speed robust prediction.

At the same time, the thesis leaves clear open fronts: stronger conditional reliability, dynamic non-exchangeable settings, realistic perturbation models, recalibration-efficient robustness, and free-form language outputs. These are not peripheral extensions; they are central to making uncertainty quantification truly trustworthy in modern machine learning systems.

The overall direction is therefore straightforward: preserve the finite-sample rigor of conformal prediction, while broadening its assumptions and reducing its operational cost until robust uncertainty quantification can be routinely deployed wherever model decisions matter.

A Supplementary Material: Conformal Prediction for Graph Neural Networks

A.1 Conformal Prediction Algorithm

Assuming that a black box model f has been chosen, as well as a score function $s(\cdot, \cdot)$, the construction of a conformal prediction set is straightforward. On a hold-out calibration set, we compute conformal scores for each pair of input and the corresponding true class. Then we sort them and save the α quantile as a calibration quantile variable. During the evaluation procedure, for each datapoint, we evaluate the conformal score for each of the classes, and we take those with scores higher than the quantile as elements of the prediction set. See Fig. A.1 for a better intuition about the position of the quantile with respect to true classes and false classes. Algorithm 3 outlines the steps to obtain a prediction set with a coverage guarantee equal to a user-selected $1 - \alpha$.

Algorithm 3 Conformal prediction pseudo-code

Input: Model f, score function s, held-out[a] labeled calibration data $\mathcal{D}_{\text{cal}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, unseen new input $(\boldsymbol{x}_{n+1}, ?)$, coverage guarantee $1 - \alpha$

1: $\forall (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}$ compute $s(\boldsymbol{x}_i, y_i) = s_i$
2: Sort all scores $\mathcal{S} = \{s_i\}_{i=1}^{n}$
3: Set $\hat{q} := \text{Quantile}\left(\frac{\lfloor (n-1)(\alpha) \rfloor}{n}; \mathcal{S}\right)$
4: Compute $s(\boldsymbol{x}_{n+1}, y_j)$, for all $y_j \in \mathcal{Y}$

Return: $C_\alpha(\boldsymbol{x}_{n+1}) = \{y_j : \; s(\boldsymbol{x}_{n+1}, y_j) \geq \hat{q}\}$

[a] Recall that in the transductive setting, the feature and graph structure of the calibration (and the test nodes) are available to the model during training, but their labels are not. Thus, held-out here refers to the labels.

In addition to the mentioned algorithm, the Python implementation including the code to reproduce reported results is accessible at `https://github.com/soroushzargar/DAPS`.

A.1.1 Computational Complexity of DAPS

Alongside the time complexity of conformal prediction, to apply DAPS or its generalizations we have to consider additional computation. Simple diffusion takes $O(E)$, and its k-hop generalization takes $O(k \cdot E)$ additional runtime. This complexity is added to the whole procedure and we need to run it only once (for transductive setting). The complexity of the SP generalization is dominated by the complexity to compute the propagated scores. In practice, we do not compute the inverse matrix but rather use only a few (e.g. 10) steps of power iteration which is enough to get a good approximation. There is a rich literature on scalable approximations to personalized PageRank that is also applicable here. We highlight that we applied DAPS to `OGBN Products` a graph with more than 2.4 million nodes (with a wall-clock time of 631 ms).

Experimental setting. We based our implementation on PyTorch Geometric Fey and Lenssen (2019). Given the computation efficiency of DAPS, we run all our experiments

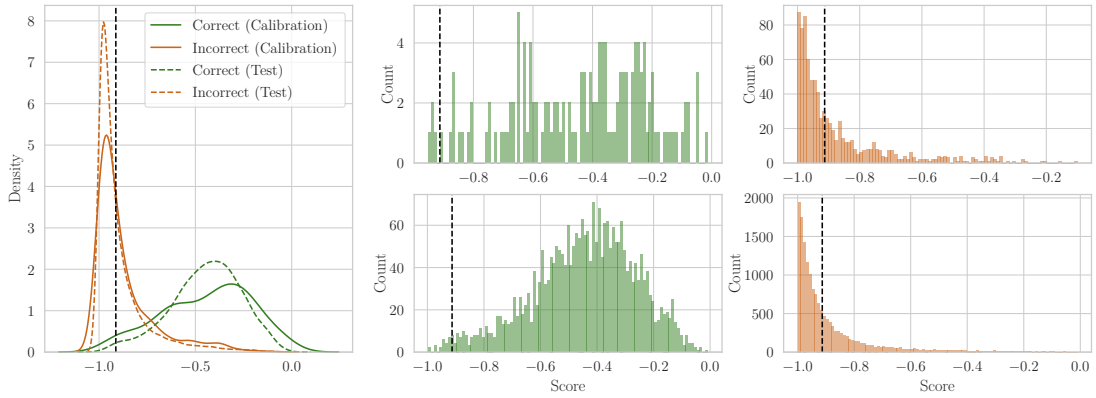

Figure A.1: Scores with respect to the selected quantile. A density plot of the scores (left) where solid lines show the calibration set and dashed lines show the test set. The 4×4 boxes (middle and right) show a histogram of calibration scores (upper row) and test scores (lower row) for both true classes (green) and false classes (orange). On all plots, the dashed black line shows the place of the α quantile.

both on CPU (Intel(R) Xeon(R) Platinum 8368 CPU @ 2.40GHz) and, even if not necessary, on GPU (NVIDIA A100-SXM4-40GB).

A.2 Tuning Calibration Parameters

Split conformal prediction relies on held-out labeled data for calibration. It is not always realistic to assume that a large proportion of data is accessible for this purpose. This restriction becomes more critical in sparsely-labeled semi-supervised node classification tasks since the goal is to predict all node labels in the graph based on a small proportion of training nodes. Methods like DAPS and RAPS require additional labeled data for tuning calibration parameters. Naively, one might require two other labeled sets, analogous to the calibration/evaluation sets used in the final algorithm, in order to estimate the effect of different hyperparameters (e.g. different values of λ). We show that tuning can be conducted using only one set, which we call the tuning set. Our tuning procedure is the same as the procedure in RAPS, which uses a tuning set to select λ and k. Here, we only provide a principled justification of why this is a good idea. Specifically, we show that the expected set size on the tuning set is almost surely the same as the expected set size on the test set.

The expected set size is determined by: (1) performing a calibration over tuning scores, (2) defining prediction sets for elements in the same tuning set, and finally (3) computing the effective set size for nodes in the tuning set. The calculated number is an approximation for the effective set size over the rest of the unlabeled nodes. Formally, the expected set size for the conformal prediction with $1-\alpha$ coverage and the holdout tuning set $\mathcal{I}_\tau$ is derived as

$$\mathbb{E}[\text{ESS}_\alpha(\mathcal{D})] = \frac{\sum_{\boldsymbol{x}_i \in \mathcal{I}_\tau} \sum_{j \in \{1,\dots,C\}} \mathbb{I}\left(s(\boldsymbol{x}_i, y^{(j)}) > \text{Quantile}\left(\frac{\lfloor (n-1)(\alpha) \rfloor}{n}; \{s(\boldsymbol{x}_i, y_i)\}_{i=1}^n\right)\right)}{|S_\tau|} \tag{74}$$

where $s(\boldsymbol{x}_i, y^{(j)})$ is the score value for class j and node i in the tuning set. This could be rewritten as $\mathbb{E}[\text{ESS}_\alpha(\mathcal{D})] = \frac{1}{|\mathcal{D}|} \sum_{\boldsymbol{x} \in \mathcal{D}} \mathbb{E}[\text{ESS}_\alpha(\boldsymbol{x})]$.

For simplicity, we consider binary classification, but the extension to multiple classes

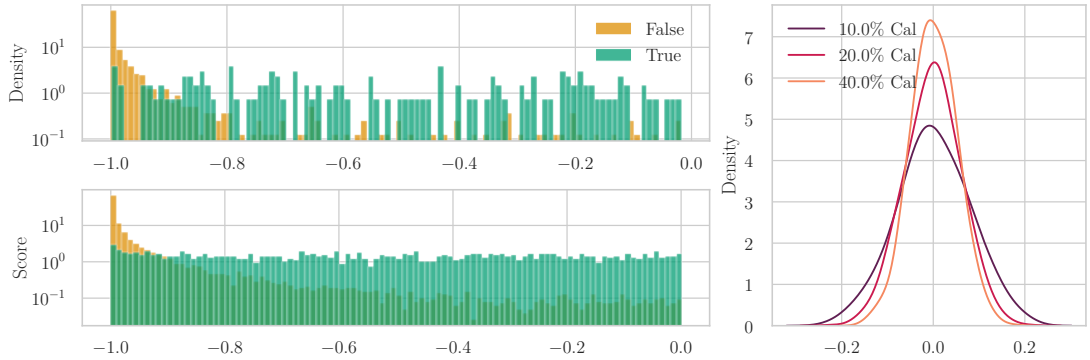

Figure A.2: The density histogram of the scores in the calibration set (upper left) and the evaluation set (lower left) alongside the marginal difference between the estimated set size and the actual effective set size (right). As shown in the figure, the estimation error for the average set size is centered around zero and the concentration of this error is correlated with the size of the calibration set. Here we use the APS score, but similar results hold for all other scores.

is trivial. Assume $S = \{s(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \sim \mathcal{D}$ is sampled exchangeably from the test dataset. Note that in this notation y_i is the true class for $\boldsymbol{x}_i$; we also call the false class as y_i'. For $1-\alpha$ coverage, we take $\hat{q} := \text{Quantile}\left(\frac{\lfloor (n-1)(\alpha) \rfloor}{n}; \{s(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}\right)$. There exists a α^r for the derived value $\hat{q}$ for which we have $\hat{q} = \text{Quantile}\left(\frac{\lfloor (n-1)(\alpha^r) \rfloor}{n}; \{s(\boldsymbol{x}_i, y_i')\}_{i=1}^{n}\right)$. For each individual node $\boldsymbol{x}_i$ we assume that the probability of class $y^{(1)}$ is η_i, hence the probability of $y^{(2)}$ is equal to $1-\eta_i$. The expected set size for the node x_i is based on two independent random events; $y^{(1)} \in C(x_i)$, and $y^{(2)} \in C(\boldsymbol{x}_i)$. If each of these classes is selected in the prediction set, the expected set size increases by one unit; hence the expected prediction set size for the set $C(\boldsymbol{x}_i)$ is

$$\mathbb{E}[|C(\boldsymbol{x}_i)|] = 1 \times \mathbb{P}[y^{(1)} \in C(\boldsymbol{x}_i)] + 1 \times \mathbb{P}[y^{(2)} \in C(\boldsymbol{x}_i)] \tag{75}$$

We can expand $\mathbb{P}[y^{(j)} \in C(\boldsymbol{x}_i)]$ to two conditions based on whether $y^{(j)}$ is true, and a supplementary term that determines whether the set contains $y^{(j)}$ given that it is true or false:

$$\mathbb{E}[|C(\boldsymbol{x}_i)|] = [\eta_i(1-\alpha) + (1-\eta_i)(1-\alpha^r)] + [(1-\eta_i)(1-\alpha) + (\eta_i)(1-\alpha^r)] = (1-\alpha) + (1-\alpha^r) \tag{76}$$

Based on definition of α^r we have

$$\mathbb{E}[|C(\boldsymbol{x}_i)|] = 1 - \alpha + \sum_{j: y^{(j)} \neq y_i} \mathbb{I}\left(s(\boldsymbol{x}_i, y^{(j)}) > \text{Quantile}\left(\frac{\lfloor (n-1)(\alpha) \rfloor}{n}; \{s(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}\right)\right) \tag{77}$$

This yields an expected prediction set size for any node inside the tuning set. Similarly we have $\mathbb{E}[\text{ESS}_\alpha(\mathcal{I})] = \frac{1}{|\mathcal{I}|}\sum_{x_i \in \mathcal{I}} \mathbb{E}[\text{ESS}_\alpha(\boldsymbol{x}_i)]$. As the tuning set $\mathcal{I}$ is exchangeably sampled from $\mathcal{D}$, the plug-in estimator is unbiased (see Berti and Rigo (1997)), hence $\mathbb{E}[\text{ESS}_\alpha(\mathcal{D})] = \mathbb{E}[\text{ESS}_\alpha(\mathcal{I})]$. Note, for the special case of i.i.d., one can also use the Dvoretzky–Kiefer–Wolfowitz inequality Dvoretzky et al. (1956) to characterize the approximation error, however, we omit this discussion here since we focus on the more general exchangeable setting.

In our experiments, we use a tuning set with the same size as the calibration set (and the training/validation sets) of just 20 nodes per class on average. Thus, the total

sum of labeled nodes used is still relatively small. This is in contrast to e.g. applications of CP in computer vision, where a large number of labels are available. Both RAPS and DAPS share the same random indices for the splits. We exclude the tuning set from any further evaluation or calibration steps. In other words, to make sure that the coverage guarantee is not violated we do not reuse the tuning set during the final calibration which uses "fresh" data. As an empirical verification of the above statement, we empirically compare our estimate of the expected average set size (i.e. the efficiency) with the average set size on the test set. In Fig. A.2 we see that the distribution of errors has a mean around zero and the variance scales with the number of calibration samples.

A.3 Proofs

In this section, we provided the proofs that were omitted from the main paper.

Proposition A.1. *Assume that $\mathcal{V}_c \cup \mathcal{V}_u$ is exchangeable. Let $\pi(G) = \mathbf{\Pi} \in \triangle^{|\mathcal{V}|\times K}$ be a matrix where row v is the label distribution for node v predicted by any permutation equivariant GNN classifier $\pi(\cdot)$ trained on the entire graph G and only using labels for nodes in $\mathcal{V}_d$. Then the scores $s(v, y) = \mathbf{\Pi}_{vy}$ where $\mathbf{\Pi}_{vy}$ is the predicted probability for node v and class y, are exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$.*

Proof of Proposition 3.2. Let $g(\boldsymbol{X}, \boldsymbol{A})$ be the function that takes the entire graph, trains the model $\pi(\cdot)$ using only the labels for the nodes in $\mathcal{V}_d$, and returns the prediction only for the calibration and test nodes $\mathcal{V}_c, \mathcal{V}_u$. Since, we assume that $\pi(\cdot)$ is permutation equivariant, this implies that g is also permutation equivariant w.r.t. a *subset* of nodes. To see this, we construct the matrix $\mathbf{\Pi}$ as the prediction matrix $\mathbf{\Pi} = \pi(\boldsymbol{X}, \boldsymbol{A})$. This matrix consists of a block of labeled nodes, corresponding to the nodes in $\mathcal{V}_d$, and two blocks of nodes corresponding to $\mathcal{V}_c$ and $\mathcal{V}_u$ respectively. Without loss of generality, let the first $|\mathcal{V}_d|$ rows correspond to $\mathcal{V}_d$, the next $|\mathcal{V}_c|$ rows correspond to $\mathcal{V}_c$, and the final $|\mathcal{V}_u|$ rows correspond to $\mathcal{V}_u$, i.e. $\mathbf{\Pi} = [\mathbf{\Pi}^d, \mathbf{\Pi}^c, \mathbf{\Pi}^u]^T$ where $\mathbf{\Pi}^d, \mathbf{\Pi}^c, \mathbf{\Pi}^u$ correspond to the three blocks. Then, $g(\boldsymbol{X}, \boldsymbol{A}) = [\mathbf{\Pi}^c, \mathbf{\Pi}^u]^T$. Since π is permutation equivariant, we have that for any permutation ω, it holds $g(\omega\boldsymbol{X}, \omega\boldsymbol{A}) = \omega[\mathbf{\Pi}^c, \mathbf{\Pi}^u]^T$. Now, the result directly follows given the assumption that the nodes in $\mathcal{V}_c \cup \mathcal{V}_u$ are exchangeable and the fact that permutation equivariant functions preserve exchangeability Kuchibhotla (2020). □

Recall, that the while we do have and do use the labels for $\mathcal{V}_c$ for calibration, these labels are not used during training.

Proposition A.2. *Let $\boldsymbol{H} \in \mathbb{R}^{|\mathcal{V}|\times K}$ be any matrix where row v is the conformal scores for all classes y for node v, and let $\boldsymbol{H}$ be exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$. Then the diffused scores $\hat{\boldsymbol{H}}$ are also exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$.*

Proof of Proposition 3.3. Similar to the proof of Proposition 3.2, the diffusion of the scores defined in Eq. 35 that results in $\hat{\boldsymbol{H}}$ is permutation equivariant for all nodes, and hence also for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$. To see this notice that $\hat{\boldsymbol{H}} = (1-\lambda)\boldsymbol{H} + \lambda\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{H}$ is a special case of a message passing GNN layer. It follows that the diffused scores are also exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$. □

Theorem 2. Let $\boldsymbol{\pi}_i$ be the model's approximation of the ground-truth conditional probability vector $\boldsymbol{p}_i$, and let the diffused distribution be $\hat{\boldsymbol{\pi}}_i = (1-\lambda)\boldsymbol{\pi}_i + \frac{\lambda}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i}\boldsymbol{\pi}_j$. Assume that the $G = (\boldsymbol{X}, \boldsymbol{A})$ is constructed such that $\boldsymbol{A}_{ij} = 1$ iff $||\boldsymbol{p}_i - \boldsymbol{p}_j|| \leq \Delta$ where $||\cdot||$ is the total variation norm. Diffusion improves the approximation error $\epsilon_i = ||\boldsymbol{\pi}_i - \boldsymbol{p}_i||$, i.e. $||\hat{\boldsymbol{\pi}}_i - \boldsymbol{p}_i|| < \epsilon_i$ if $\frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i}\epsilon_j + \Delta < \epsilon_i$.

Proof of Theorem 3.4. To show $\|\boldsymbol{p}_i - \hat{\boldsymbol{\pi}}_i\| < \|\boldsymbol{p}_i - \boldsymbol{\pi}_i\| = \epsilon_i$ we use the definition of $\hat{\boldsymbol{\pi}}_i$

$$\|\boldsymbol{p}_i - \hat{\boldsymbol{\pi}}_i\| = \|\boldsymbol{p}_i - (1-\lambda)\boldsymbol{\pi}_i - \frac{\lambda}{|\mathcal{N}_i|} \cdot \sum_{j \in \mathcal{N}_i} \boldsymbol{\pi}_j\| \tag{78}$$

Leveraging the fact that $\boldsymbol{p}_i = (1 - \lambda \cdot \boldsymbol{p}_i) + \lambda \cdot \boldsymbol{p}_i$, we have

$$\left\| \boldsymbol{p}_i - (1-\lambda)\boldsymbol{\pi}_i - \frac{\lambda}{|\mathcal{N}_i|} \cdot \sum_{j \in \mathcal{N}_i} \boldsymbol{\pi}_j \right\| = \left\| (1-\lambda)(\boldsymbol{p}_i - \boldsymbol{\pi}_i) + \lambda \left(\boldsymbol{p}_i - \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \boldsymbol{\pi}_j \right) \right\| \tag{79}$$

$$\leq (1-\lambda)\|\boldsymbol{p}_i - \boldsymbol{\pi}_i\| + \frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \|\boldsymbol{p}_i - \boldsymbol{\pi}_j\| \tag{80}$$

$$\leq (1-\lambda)\epsilon_i + \frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \|\boldsymbol{p}_i - \boldsymbol{\pi}_j\| \tag{81}$$

$$\leq (1-\lambda)\epsilon_i + \frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (\|\boldsymbol{p}_i - \boldsymbol{p}_j\| + \|\boldsymbol{p}_j - \boldsymbol{\pi}_j\|) \tag{82}$$

$$\leq (1-\lambda)\epsilon_i + \frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (\Delta + \epsilon_j) \tag{83}$$

Where we repeatedly use the triangle inequality.

Conclusively, we want to prove that

$$(1-\lambda)\epsilon_i + \frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \epsilon_j + \lambda\Delta < \epsilon_i \tag{84}$$

$$\frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \epsilon_j + \lambda\Delta < \lambda\epsilon_i \tag{85}$$

By dividing everything by λ

$$\frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \epsilon_j + \Delta < \epsilon_i \tag{86}$$

Which is the basic assumption of the proposition. □

A.4 Synthetic Experiment with Access to the Ground-Truth Distribution

Additional to the provided theoretical insights on the effect of neighborhood diffusion, we also carried out a supplementary experiment utilizing synthetic data. In subsubsection 3.4.2 we discussed a perturbation of the ground-truth data. We use the following perturbation scheme:

$$\pi(x_i) = \pi_i = \begin{cases} p_i + u \cdot \epsilon_h & \text{if Bernoulli}(p_s) = 1 \\ p_i + u \cdot \epsilon_l, & \text{otherwise} \end{cases} \tag{87}$$

where ϵ_h is a high-magnitude perturbation coefficient, ϵ_l is a low-magnitude perturbation coefficient, u is a uniform random variable, and p_s defines the probability of

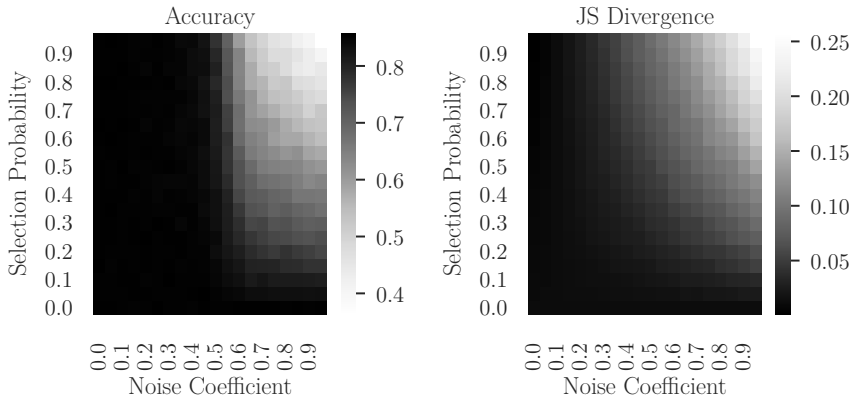

Figure A.3: Accuracy and Jensen Shannon divergence for various degrees of perturbation over the ground-truth label distribution. The x-axis shows the intensity of noise applied to highly perturbed nodes (ϵ_h) and the y-axis shows the probability of the node being highly perturbed (p_s). A highly accurate model does not imply a good approximation of the true label distribution.

a node being highly perturbed. We make sure that the resulting probability vector is normalized. The result is a perturbed distribution that aligns with ground truth for many nodes (with a very small shift) while highly perturbing the small proportion of selected nodes.

For the experiment in Fig. 3.2 we set $\epsilon_h = 0.7$, $\epsilon_l = 0.1$, and $p_s = 0.2$. First, we generate a dataset based on two different multivariate Gaussian distributions. Each point in the resulting dataset would be a conditional probability vector indicating how much the point is likely to belong to each class. Labels are also sampled from the same distribution. Here the graph is the k-NN graph with $k = 15$. In Fig. A.3 we study the effects of changing the high-magnitude perturbation coefficient ϵ_h from 0.0 to 0.9. We measure the effect on the accuracy and on the Jensen–Shannon divergence to the ground-truth probability distribution. We see that a perturbation up to a certain degree can cause a large JS divergence from the ground-truth while preserving the rank of the class with the highest probability (and thus the accuracy). This shows that an accurate model does not guarantee a good approximation of the label distribution.

Potential issue with smoothed probabilities. One potential pitfall of the diffusion approach which is also observable in the synthetic data experiment (see Fig. 3.2) is that it results in less confident probability vectors. This is a general phenomenon since there is an aggregation involved in the transformation. One can notice the resulting underconfidence in Fig. 3.1 where for DAPS we see a decrease at the end (highly-confident) part of the histogram. However, the denoising effect of diffusion makes such a valuable enhancement, that the effect of smoothing is negligible. This issue can be mitigated with temperature scaling, however we leave this for future work.

A.5 Additional Experiments and Experimental Details

A.5.1 Technical Details for the Hard Prediction Case

As mentioned in § 3.4 and further evaluated in § 3.5, one additional application of neighborhood diffusion is in cases where only the hard predictions of nodes are provided. Technically, in such cases methods like APS can not perform as expected since

the one-hot probability vector causes many ties in the score space. Even the built-in uniform randomization in APS can not overcome this problem. Inspecting the definition of APS, we see that the classes with probability of 0 are directly mapped to -1 which again results in ties despite the randomization. As a technical solution to this problem, we increase each non-top class by a small constant $\epsilon = 0.001$ and decrease the top class by $|C| \cdot \epsilon$. After this step, the resulting vectors can be used with APS and we can obtain valid coverage. However, APS results in large prediction sets for almost all inputs. This makes sense, since there is no information about the ranking of the (non-top) classes. As we discussed before, this is also the reason why RAPS is not applicable, even with the ϵ transformation. For a fair comparison, we apply the same transformation to DAPS even though it can work without it.

A.5.2 Combination of Scoring Functions

In addition to individually comparing DAPS and RAPS (see subsubsection A.5.9), another idea is to apply each of the approaches as an add-on to the other. This means that we can perform the regularization defined in RAPS on top of the diffusion in DAPS or vice versa. We evaluate the enhancement of such combined scores (relative to APS) in Fig. A.4. We see that diffusion provides additional benefits to RAPS. However, we do not recommend using these combinations since they inherit the instabilities of RAPS and its sensitivity to different initial splits and different α values.

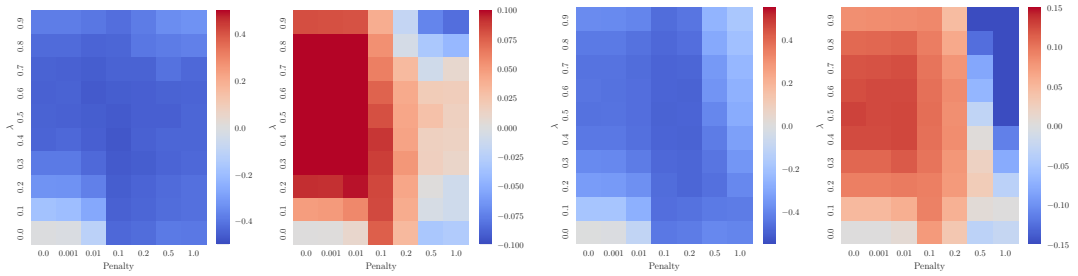

Figure A.4: Applying RAPS regularization over DAPS (left two plots), and diffusion over RAPS (right two plots). For each pair of plots the left subplot shows the enhancement in efficiency and the right subplot shows the enhancement in singleton hit ratio (the results are relative to APS). All plots of this figure have the $k = 0$ for RAPS.

A.5.3 Adaptive Coverage Guarantee

Many recent studies report their results on a fixed coverage guarantee. We argue that an adaptive coverage guarantee (which we define next) is more meaningful. In any case, we report results using both fixed and adaptive coverage. To make the coverage adaptive, for each dataset-model pair, we set $1-\alpha$ relative to the accuracy of the selected model for the selected dataset. In our examples, this value is set to a weighted average between the model accuracy and 100% with weights $(\frac{1}{3}, \frac{2}{3})$. This results in a value of $1-\alpha$ that is always larger than the accuracy. For example, if the model has accuracy of 97% it is less informative to use a fixed $1-\alpha = 0.9$ which is often the default. Here the adaptive coverage $1-\alpha = 0.99$ is more realistic. For all results in the main paper, except Fig. 3.7 (right), we use adaptive coverage, the concrete selected values are given in Table A.3. The value of α is important since the performance of CP is sensitive to the distance between $1-\alpha$ and the model's accuracy (see Fig. 3.4 as one of many examples).

In other words, assuming everything else is the same, a model with higher accuracy

is expected to exhibit superior performance under a fixed coverage guarantee. Hence, it is essential to examine different methods for both fixed and adaptive coverage. Fig. A.5 illustrates a significant difference between the two settings (fixed and adaptive coverage). Nonetheless, DAPS performs well in both settings. Since the accuracy of `Coauthor Physics` and `Coauthor CS` is relatively high (see Table A.3), all models including APS seem to perform well for fixed coverage, which is not true for the adaptive coverage.

A.5.4 Margin Scoring Function

In this study, we focused on applying diffusion on top of the baseline APS score function. However, this approach is equally applicable to any other score function as well. This is also true for the regularization idea behind RAPS presented in Angelopoulos et al. (2021). In § 3.4 we argued that APS is the most suitable choice. However, since Wijegunawardana et al. (2020) proposes a different scoring function called "margin scoring", we also evaluate our diffusion method on top of that as well. The score function is defined as $s(\boldsymbol{x}, y) = \pi(\boldsymbol{x})_y - \max_{y' \neq y} \pi(\boldsymbol{x})_{y'}$. Fig. A.6 presents the evaluation of the diffusion and regularization variants of the margin scoring function. Again, we see that both provide similar improvements on top of the margin baseline. Since the margin score has issues with undercovering hard examples and overcovering easy examples, similar to TPS, we do not advocate for its use even though it appears to have good efficiency.

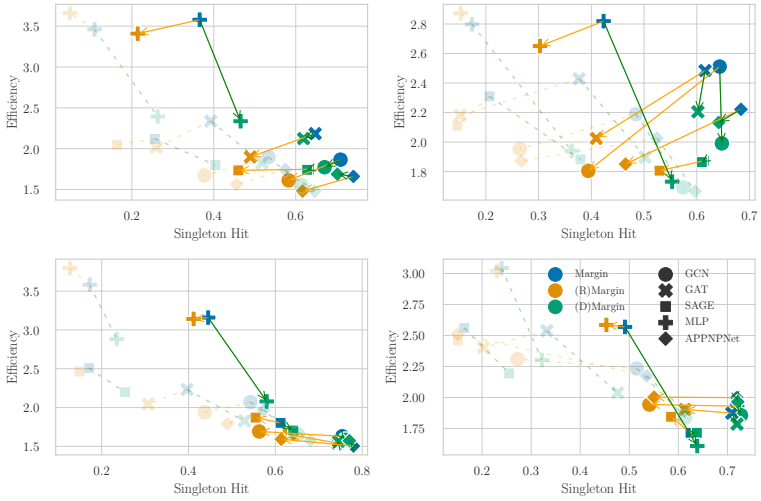

Figure A.6: Conformal prediction with the margin score for the `CoraML` (top), and `CiteSeer` (bottom) datasets. The evaluation is based on fixed 92% coverage (left) and adaptive coverage (right). The transparent plot recalls the result of using APS as a reference score.

A.5.5 Sensitivity to λ, Efficiency and Accuracy

As a supplementary discussion to subsubsection 3.5.1 we present the result of the same experiment as Fig. 3.6 over some other dataset/model pairs. Fig. A.7 shows that in almost every case the impact of the proposed diffusion on the accuracy is insignificant while it enhances the conformal set efficiency and singleton hit ratio. This gives a more intuitive sense that the diffusion framework results in more efficient sets by enhancing the approximation of the probabilities instead of increasing the model's accuracy.

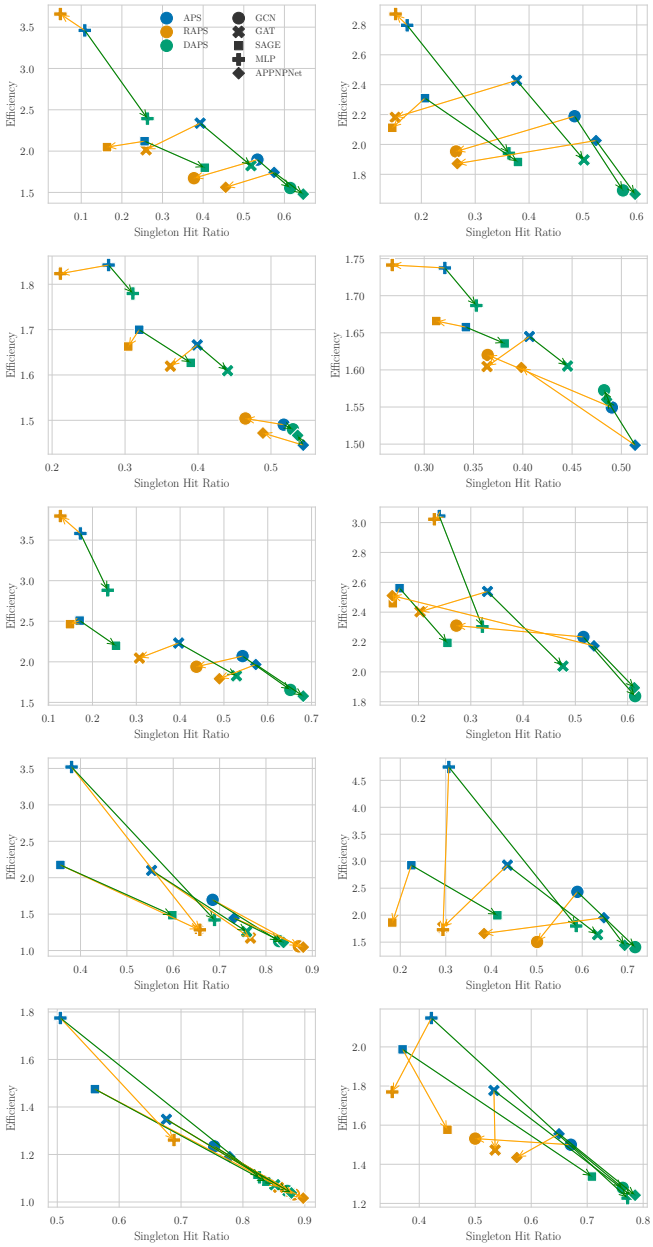

Figure A.5: The Pareto plot of different CP approaches for different datasets. From top to bottom, each row illustrates the evaluation of the approaches, namely APS, RAPS and DAPS, on `CoraML`, `PubMed`, `CiteSeer`, `Coauthor CS`, and `Coauthor Physics` respectively. The left plot in each row is regarding an experiment on 92% fixed coverage, and the right plot illustrates the result for adaptive coverage. DAPS performs best on average for both fixed and adaptive coverage.

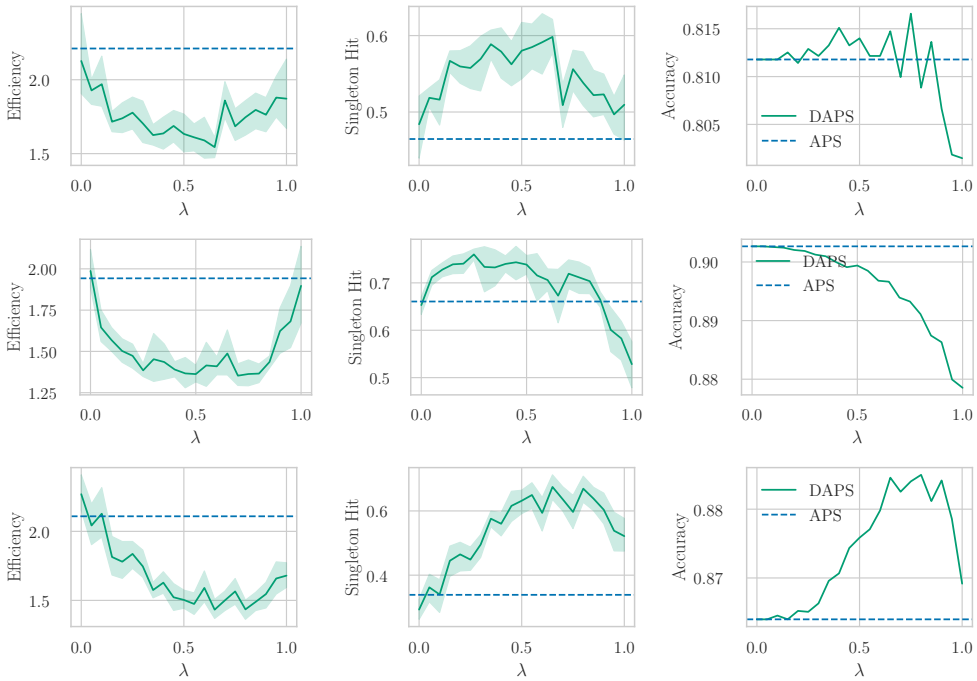

Figure A.7: The effect of different λ values on (left) efficiency and (middle) singleton hit ratio of conformal prediction alongside its impact on (right) the accuracy of the model. Rows refer to experiments conducted on (top) `CoraML/GCN`, (middle) `CoauthorCS/APPNP`, and (bottom) `Amazon-Photo/GraphSAGE` respectively.

A.5.6 Empty, Singleton and Multi-class Prediction Sets

For a given input, CP either returns a single class, a set of classes, or an empty set. One might prefer to increase the proportion of singleton sets as they can be applied without any further postprocessing. We compare the proportion of zero, singleton and multi-class prediction sets in Fig. A.8 for different coverages spanning from a threshold below the model's accuracy (which is a trivial area for CP) up to near 100% coverage. This experiment shows that DAPS results in fewer empty sets and more singleton sets, which aligns with the higher singleton hit ratio covered in § 3.5. As shown in the figure, all CP methods tend to increase the number of empty-set predictions as the coverage guarantee decreases (and becomes lower than the model's accuracy). It is likely that with lower values of coverage guarantees, CP tends to result in a smaller false positive ratio. Note, the result varies across different runs of the experiment which is a direct result of label scarcity and the small calibration set, but the order is the same in the majority of observations.

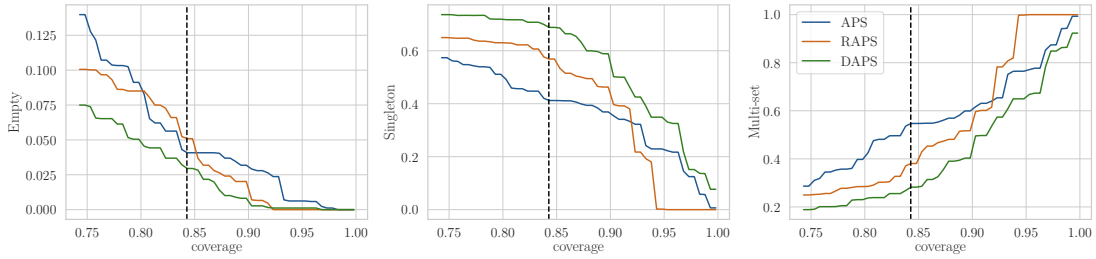

Figure A.8: Comparison of APS, RAPS, and DAPS over different coverage guarantees for `CoraML/GCN` by the proportion of empty (left), singleton (middle) and multi-set (right) prediction sets. The dashed line in all plots show the model's accuracy over the test set.

A.5.7 Empirical Evaluation of Conditional Coverage

Evaluating CP's deviation from conditional coverage requires access to ground truth $p(\boldsymbol{x}, y)$. However, Romano et al. (2020) propose an approximation that is adaptable to limited data. The procedure involves searching for a slab ($S_{v,a,b} = \{\boldsymbol{x} \in \mathbb{R}^p : a < \boldsymbol{v}^T\boldsymbol{x} < b\}$) in which the empirical coverage is at its lowest. With a finite number of datapoints, in order to avoid finite-sample negative bias, a valid slab must contain an acceptable proportion of data points (e.g. 10%). The slab's identifier vector $\boldsymbol{v}$ is chosen uniformly at random in the feature space and is normalized. We chose the optimal parameters a^*, b^*, and $\boldsymbol{v}^*$ on 25% of the test data, using the rest to evaluate the coverage. See Romano et al. (2020) for more details. Fig. A.9 shows that DAPS performs better than or on par with APS. This comparison is shown for different values of $1 - \alpha$.

A.5.8 Transformations for the Probability Space

Although we usually apply regularization (in RAPS), and diffusion as an enhancement on top of APS scores, we also examine the scenario where we apply those transformations over probability vectors (softmax outputs) as well. In both cases, we apply APS on top of the result and compare it with the conventional APS. Since APS accepts probability vectors as input, we need to represent the output of those transformations in a probability space. While, DAPS with $\lambda \in [0, 1]$ does not require any normalization to return a probability vector (since it is a convex combination), RAPS needs to be normalized (since the penalty changes the output range). To represent the regularization result in form of a probability vector, we apply a min-max normalization

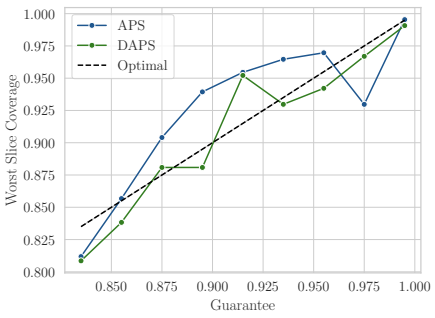

Figure A.9: Comparison of worst-slice coverage for APS and DAPS across different coverage guarantees $1 - \alpha$. Results are shown for `CoraML/GCN`. Note approaches that are closer to the optimal dashed line are better.

over elements, such that the minimum is equal to zero. Then we divide them by their summation. Fig. A.10 shows the comparison. With RAPS, we observed a significant decrease in efficiency and singleton hits while DAPS is similar to conventional APS. It is better to apply both RAPS and DAPS in the score rather than the probability space.

A.5.9 Transductive Semi-Supervised Node Classification

With a brief review of experimental results provided in § 3.5, this section presents a comprehensive report of all results obtained. We compare DAPS alongside the baseline APS, and RAPS in the form of Pareto plots where two different metrics (effective set size and singleton hit) are evaluated at the same time. Blue points in the plot show baseline values (APS results) for different dataset/model pairs. Corresponding orange and green points are respectively showing RAPS, and 1-hop DAPS. Each baseline is connected to the two other results by an arrow of the same color. Fig. A.5 shows the result on co-author networks and Fig. A.11 shows a similar evaluation on the co-purchase networks. We also conducted the same experiment on `CoraFull⋆` and `CoraML⋆`, for which the result is reported in Fig. A.12. Although we have shown the adaptability of DAPS to large networks like `OGBN Products` in Fig. 3.4, we also evaluated our method on another large `OGBN Arxiv` dataset. As shown in Fig. A.13, for `OGBN Arxiv` DAPS achieves a marginal enhancement and RAPS returns the most efficient sets among other methods. In spite of losing efficiency, DAPS outperforms RAPS in terms of singleton hit ratio for many coverage values. It is noteworthy that we did not expect a significant improvement for DAPS in this experiment since `OGBN Arxiv` does not have a high homophily score which is an essential requirement for this approach. For the same reason, we do not expect a significant enhancement in `CoraFull⋆` as well.

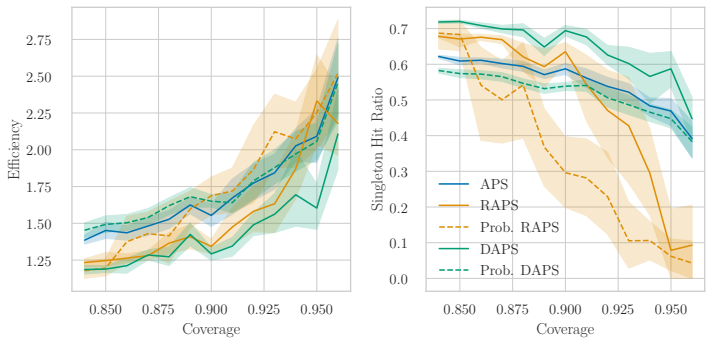

Figure A.10: Comparison of DAPS and RAPS in probability space (dashed lines) and in APS (solid lines) score space.

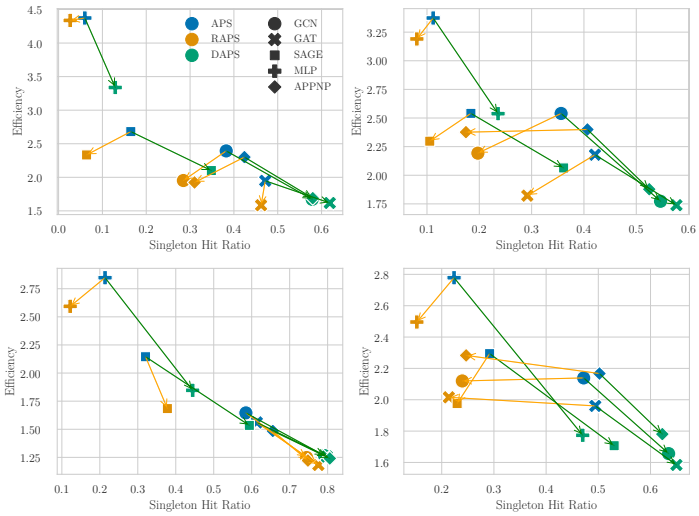

Figure A.11: Pareto plot of different CP approaches for co-purchase datasets; APS, RAPS, and DAPS. The first column shows the result for the fixed 92% coverage and the right column shows the result for adaptive coverage. Rows from above to below refer to (1) Amazon Computers and (2) Amazon Photo datasets.

A.5.10 Other Variants of Semi-Supervised Node Classification

While our focus is on the transductive setting, we conduct additional experiments for other settings as well. Here we provide our experimental results on inductive and simultaneous inductive settings. For both cases, we trained our model on the inductive subgraph restricted to only training and validation nodes. Definitions and theoretical analysis of exchangeability for these settings are provided in § 3.3. For the inductive setting, we add calibration nodes to the training graph (creating the induced subgraph of training/validation/calibration nodes) during CP's calibration step. The rest of the evaluation nodes (with their connections) are added one at a time. Upon each modification, we update the model predictions. The prediction set for each node is computed immediately upon its arrival. These updates in the graph structure lead to distribution shift and conclusive violation of exchangeability as shown in Fig. A.14. As a result, the $1 - \alpha$ coverage is not guaranteed anymore.

For the simultaneous inductive setting, we utilized the same model. However, this time all nodes in the rest of the network (consisting of calibration nodes and unlabeled nodes) were connected to the training graph simultaneously. After this we update the model predictions using the final graph. As shown in Fig. A.15 the coverage guarantee is still valid since exchangeability is not violated. The only difference between this setting and the transductive setting is the performance of the underlying model which is reflected in the conformal prediction metrics.

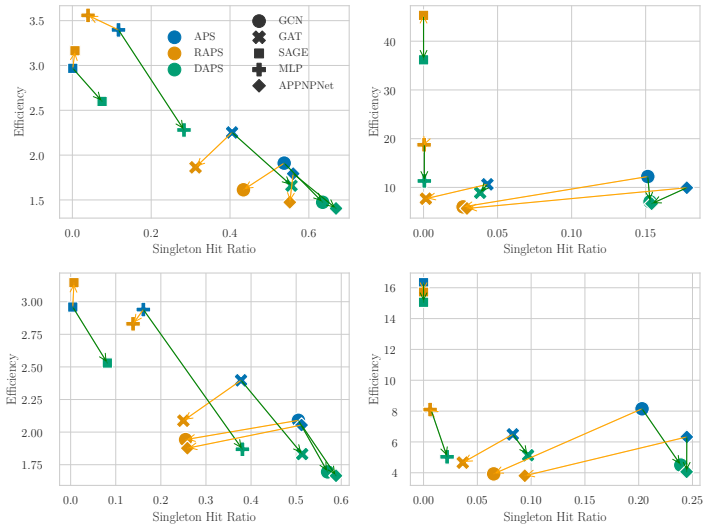

Figure A.12: Pareto plot on the effective set size and singleton hit over the datasets `CoraML`* (first column) and `CoraFull`* (second column). (First row) shows the result for a fixed coverage (92%) and (second row) over the adaptive coverage.

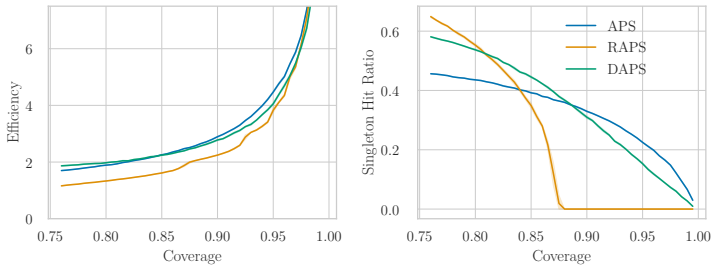

Figure A.13: Efficiency (left) and singleton hit ratio (right) for the `OGBN Arxiv` dataset. Since we have less homophily DAPS sacrifices efficiency to improve singleton hits.

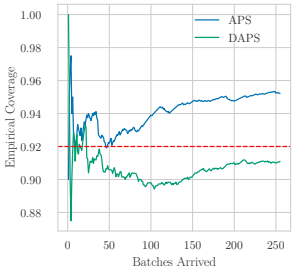

Figure A.14: Inductive evaluation setting for `Cora-Ml/GCN`. Accuracy is 82%.

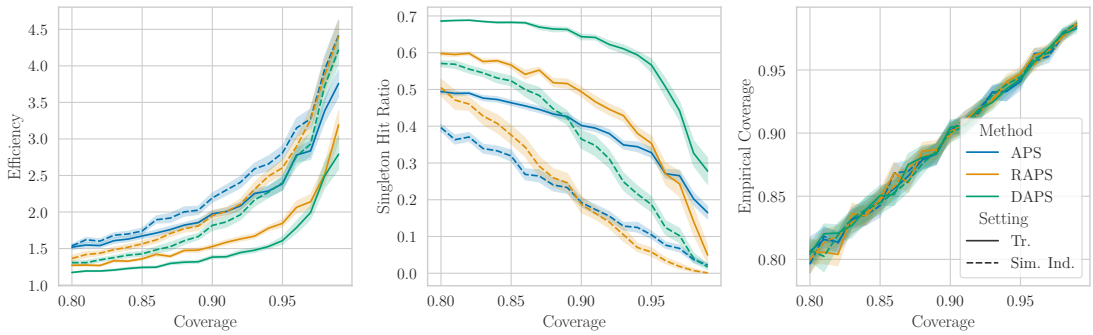

Figure A.15: Simultaneous inductive setting on `CoraML` with a `GCN`. Solid lines are recalling the same result for transductive setting while dashed lines show the results of the simultaneous inductive setting. DAPS always leads to an improvement.

A.5.11 Discussion of Neighborhood Adaptive Prediction Sets (NAPS)

For the inductive semi-supervised node classification, Clarkson (2022) proposes NAPS (Neighborhood APS) which is built upon Barber et al. (2022) to adapt conformal prediction "beyond exchangeability". Applying conformal prediction without exchangeability leads to a gap between the empirical (real) coverage and the specified $1 - \alpha$. This gap is bounded by

$$\text{Coverage Gap} \leq \frac{\sum_{i=1}^{n} w_i \cdot d_{TV}(Z, Z^i)}{1 + \sum_{i=1}^{n} w_i} \tag{88}$$

where w_i corresponds to a weight over i-th datapoint ($w_i \in [0, 1]$), and Z^i is the result of swapping i-th datapoint in the calibration set Z with the test point. Conclusively, a better weight assignment can result in a smaller coverage gap. In NAPS, a weight of $w_i = 1$ is assigned in case the test node is within k-hop distance of the i-th calibration node. The study suggests using NAPS only on large homophilous networks with $k = 1$ or 2. Note, the coverage gap is a theoretical property and it is not straightforward to compute or estimate it in practice.

As we discussed in § 3.3 NAPS is not applicable when the graph is sparse and the number of labeled nodes is limited. When the size of the calibration set is realistic, many test nodes will have no nodes with non-zero weights from the calibration set, making CP inapplicable to them. This happens regardless of the inductive or transductive setting. Even if somehow sparsity is not an issue, since NAPS assigns weights from $\{0, 1\}$, a substantial proportion of the calibration set will be effectively discarded for each node. Consequently, the smaller calibration set leads to a less concentrated coverage distribution (i.e. less concentrated Beta) and less statistical power. It is worth noting that the claim by Clarkson (2022) that NAPS (or any other CP score) is not to applicable for transductive settings does not hold as shown by Proposition 3.2.

In Table 3.2 we compared DAPS with NAPS in the transductive setting for different calibration set sizes. We see that as we approach a realistic labeling budget, a considerable amount of nodes are excluded from the CP procedure (due to all zero weights). This observation holds true even when the algorithm is applied in the inductive setting using the same dataset since the source of this limitation remains the same.

A.5.12 Comparison Over Training Checkpoints

As CP is built on top of a model, to evaluate CP, it becomes important how well the model is trained. The question is how the enhancement made by DAPS (in comparison to conventional APS) changes during training. Since the categorical cross-entropy loss encourages the model to predict a concentrated "one-hot" label distribution, it is expected that the predicted probabilities become more over-confident when training the model for too many epochs. This may be an issue for APS. DAPS uses structural information to propagate the scores and overcomes this issue.

To support this discussion we compared DAPS and APS for a GNN model during different checkpoints of the model's training. The results in Table A.1 show the enhancements made by DAPS has an increasing trend of improvement while the model becomes more accurate.

Table A.1: Comparison between DAPS and APS over different checkpoints during the model training.

Checkpoint	Acc	APS		DAPS		Difference	
		Eff Set Size	Singleton Hit	Eff Set Size	Singleton Hit	Eff Set Size	Singleton Hit
1	0.21	5.53	0.00	5.38	0.00	0.15	0
2	0.60	3.69	0.00	3.51	0.00	0.18	0
3	0.72	2.38	0.08	2.17	0.17	0.22	0.09
4	0.76	2.26	0.14	1.99	0.26	0.27	0.12
5	0.81	2.23	0.22	1.78	0.40	0.45	0.23
6	0.84	2.25	0.34	1.58	0.56	0.67	0.22

A.6 Complementarity with Other Methods for Uncertainty Quantification

Another interesting insight is that if we are provided with a good uncertainty estimation model, applying conformal prediction on top should return even better results. To show this, we evaluate CP with the Graph Posterior Network (`GPN`) Stadler et al. (2021) on `CoraML`. In particular, we compute the scores for the conformal prediction based on the class probabilities as a measure of the aleatoric uncertainty which is also the kind of uncertainty captured by conformal prediction. The results are shown in Fig. A.16. Comparing `GPN` and `GCN` and we can see that even though their performance is close, we can note an improvement for the APS and RAPS methods when using an uncertainty-aware model like `GPN`. DAPS is on par for both underlying models, which suggests that our method captures the aleatoric uncertainty regardless of the model.

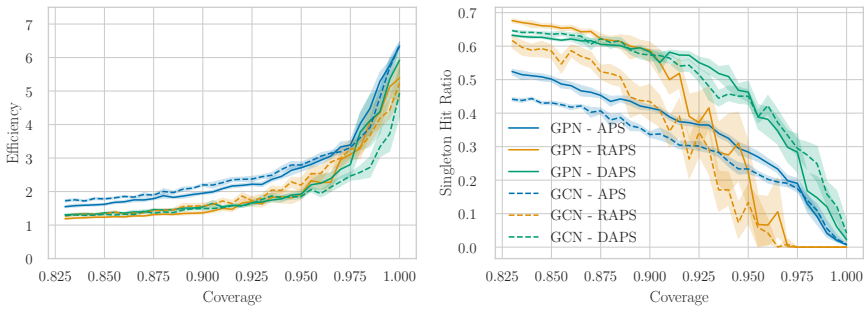

Figure A.16: Comparison of the approaches on `GPN` and `GCN` model over `CoraML` dataset. While `GPN` helps APS and RAPS, especially for singleton hit, our approach DAPS is already able to provide an uncertainty quantification with a vanilla `GCN`.

A.7 More Details on Datasets and Models

Table A.2 displays the statistics of the datasets used for the evaluation. The datasets marked by ⋆ refer to the largest component. Moreover, for `CoraFull`* we remove the classes (and the respective nodes) that have a number of samples less than 50 in order to have the same number of nodes per class in each train/validation split. Table A.3 summarizes the model's accuracies on every dataset, and the selected adaptive coverage as explained in subsubsection A.5.3.

Table A.2: Statistics of the datasets. The labeled node column includes all nodes that are assumed to be labeled in each experiment which is a summation of training, validation, tuning, and calibration nodes.

Dataset Name	Vertices	Attributes	Edges	Classes	Homophily	Labeled Nodes
CoraML	2995	2879	16316	7	78.85%	18.7%
CoraML*	2810	2879	15962	7	78.44%	14.95%
CoraFull*	18712	8710	124848	67	56.69%	20.41%
PubMed	19717	500	88648	3	80.23%	1.2%
CiteSeer	4230	602	10674	6	94.94%	10.8%
Coauthor CS	18333	6805	163788	15	80.80%	6.5%
Coauthor Physics	34493	8415	495924	5	93.14%	1.2%
Amazon Computers	13752	767	491722	10	77.72%	5.8%
Amazon Photo	7650	745	238162	8	82.72%	8.4%
OGBN Products	2449029	100	123718280	47	80.75%	11.24%
OGBN Arxiv	169343	128	1166243	40	65.51%	88.9%

Table A.3: Accuracy report for datasets and models involved in the analysis.

Dataset	Model	Accuracy	Best Accuracy	Adaptive Coverage
CoraML	GCN	82.3 ± 0.9	83.9	94.0
	GAT	79.8 ± 3.1	84.4	93.1
	SAGE	79.9 ± 1.7	82.2	93.2
	MLP	63.4 ± 1.9	65.6	87.6
	APPNPNet	83.5 ± 0.8	84.9	94.4
PubMed	GCN	79.5 ± 2.0	82.0	93. 0
	GAT	77.8 ± 3.2	82.3	92.5
	SAGE	75.7 ± 2.2	79.8	91.7
	MLP	69.9 ± 0.9	71.6	89.8
	APPNPNet	79.4 ± 2.3	82.2	93.0
CiteSeer	GCN	83.7 ± 1.4	85.9	94. 5
	GAT	83.2 ± 0.9	84.2	94.3
	SAGE	78.2 ± 2.3	80.8	92.6
	MLP	62.6 ± 1.5	65.2	87.3
	APPNPNet	84.9 ± 1.2	87.0	94.9
Coauthor CS	GCN	90.9 ± 0.8	91.8	96.9
	GAT	88.6 ± 1.1	90.3	96.1
	SAGE	88.6 ± 1.2	90.2	96.1
	MLP	88.0 ± 0.6	88.9	95.9
	APPNPNet	91.1 ± 0.5	91.7	97.0
Coauthor Physics	GCN	92.2 ± 1.2	93.4	97.3
	GAT	91.1 ± 1.1	92.9	97.0
	SAGE	92.0 ± 0.7	92.9	97.3
	MLP	87.1 ± 1.3	89.8	95.6
	APPNPNet	93.1 ± 0.9	94.5	97.7
Amazon Computers	GCN	80.2 ± 2.9	83.0	93.3
	GAT	81.8 ± 1.9	84.6	93.8
	SAGE	74.6 ± 3.1	78.1	91.4
	MLP	60.0 ± 5.3	67.3	86.4
	APPNPNet	80.5 ± 2.2	84.3	93.4
Amazon Photo	GCN	89.0 ± 2.3	91.0	96.3
	GAT	89.3 ± 1.2	91.3	96.4
	SAGE	82.4 ± 3.5	86.4	94.0
	MLP	74.2 ± 2.2	76.4	91.2
	APPNPNet	89.4 ± 1.6	91.7	96.4

B Supplementary Material: Conformal Inductive Graph Neural Networks

B.1 Algorithm for NodeEx CP and Weighted EdgeEx CP

At each timestep t with node-exchangeable graph $\mathcal{G}_t$, we first run the GNN model to extract conformity scores for all nodes in $\mathcal{V}_t$. We compute the α-quantile of the conformity scores conditional to the current graph, and for each new test node $v_{\text{test}} \in \mathcal{V}_{\text{eval}}^{(t)}$ we return any candidate label with a score larger than the conditional threshold. For an edge-exchangeable graph, the algorithm is still the same unless we compute the conditional weighted quantile with weights equal to $1/d(v)$ for all calibration vertices v. Algorithm 4 shows the pseudocode for NodeEx CP. The code is accessible at `github.com/soroushzargar/conformal-node-classification`.

Algorithm 4 NodeEx and EdgeEx CP for inductive node classification

Input: Graph $\mathcal{G}_t$ at timestep t; calibration vertex set $\mathcal{V}_{\text{cal}}$; permutation equivariant score function $s(\cdot, \cdot \mid \mathcal{G}_t)$; evaluation vertex set $\mathcal{V}_{\text{eval}}^{(t)}$
Output: $C^{(t)}(v_j)$ for each $v_j \in \mathcal{V}_{\text{eval}}^{(t)}$

Compute $s(v_i, y \mid \mathcal{G}_t)$ for all $v_i \in \mathcal{V}_t$, $y \in \mathcal{Y}$
if node-exchangeable sequence **then**
  $q_t \leftarrow \mathbb{Q}\left(\alpha; \{s(v_i, y_i \mid \mathcal{G}_t)\}_{v_i \in \mathcal{V}_{\text{cal}}}\right)$
else if edge-exchangeable sequence **then**
  $q_t \leftarrow \mathbb{Q}\left(\alpha; \{s(v_i, y_i \mid \mathcal{G}_t)\}_{v_i \in \mathcal{V}_{\text{cal}}}; \{1/\deg(v_i \mid \mathcal{G}_t)\}_{v_i \in \mathcal{V}_{\text{cal}}}\right)$
end if
for each $v_j \in \mathcal{V}_{\text{eval}}^{(t)}$ **do**
  $C^{(t)}(v_j) \leftarrow \{y : s(v_j, y \mid \mathcal{G}_t) \geq q_t\}$
end for
Return $\{C^{(t)}(v_j) : v_j \in \mathcal{V}_{\text{eval}}^{(t)}\}$

Naive CP. In naive CP we compute calibration scores before the arrival of any new node, on the inductive subgraph over $\mathcal{V}_{\text{tr}} \cup \mathcal{V}_{\text{cal}}$. Prior to the evaluation of any node, we compute the calibration quantile q from calibration scores. For any test node, the conformity score of all its classes is compared to q. Clarkson (2023) shows that this approach fails to provide valid prediction sets (the guarantee is broken due to the shift caused by the arrival of new nodes).

Computational complexity. Conformal prediction has two main computational routines in addition to model's inference: computing scores and quantiles. For the calibration set we only need the scores of the true class, but for test nodes, we should compute scores for all classes. Depending on the score function the complexity can be different. For standard CP we compute calibration scores once, but for NodeEx CP for each evaluation (timestep) we need to recompute the predictions and the conformity scores. Hence, Node(Edge)Ex CP has an overhead of $O(n \times t_s \times t)$ for t timesteps, n calibration nodes, and t_s the time for computing a conformity score for one class and one node. At each step we should also evaluate the model once if the model's prediction changes across time. Computing the quantile threshold takes $O(n)$ steps. For standard CP we have to compute this value once, but in NodeEx CP this values is updated upon

each evaluation. The total wall-clock overhead is less the a second. Additionally all mentioned complexities are for serial computation and some of CP procedures can run significantly faster via parallel computation.

Limitations of NodeEx and EdgeEx CP. Our study mainly focuses on approaches that can provide a valid CP for changing subgraphs under certain assumptions of node-or edge-exchangeability. There are three main limitations for using NodeEx (or EdgeEx) CP (i) The guarantee provided by NodeEx (and EdgeEx) CP is marginal. However, it is shown that exact conditional coverage $\mathbb{P}[y \in C(\boldsymbol{x}) \mid \boldsymbol{x}]$ is not achievable. Still, there are methods that tend to get better approximations of conditional coverage. (ii) In real-life graph sequences, it is hard to determine whether the graph sequence is node- or edge-exchangeable or neither. Here a follow-up works is to use the beyond exchangeability approach (Barber et al., 2022) to achieve valid coverage guarantees without the need for node- or edge-exchangeability assumption. (iii) Our result maintains validity as long as the selection evaluation timestep for each node is done without any knowledge of the prediction set. An adversarial selection can cause a significant deviation from the guaranteed coverage. For instance, an adversary can observe prediction sets for a particular node during various timesteps and pick the timestep of the smallest prediction set similar to the two false examples in subsubsection B.4.1. This results can lead to a significant miscoverage rate compared to the guarantee.

B.2 Addition to the Theory

Matrix $\boldsymbol{C}$. For timestep t and index i, $\boldsymbol{C}[i,t]$ indicates that whether node v_i if evaluated at timestep t is going to be covered or not. Hence this value can be either 1, indicating CP covers node v_i at timestep t, 0 showing that CP does not cover this node, or N/A showing that this node has not appeared at this timestep. This matrix is constructed using ground-truth labels and is shown only for the purpose of sanity check and evaluation. The matrix $\boldsymbol{C}$ is a very dense matrix since $1-\alpha$ entities at each column should be 1. That is why in Fig. 4.2 we showed $1-\boldsymbol{C}$ where each column is expected to have α-percentage of ones. A valid CP approach should result in a similar percentage of ones in each column. For an invalid CP, if the trend is gradually moving toward over-coverage, the matrix $1-\boldsymbol{C}$ becomes more sparse in later columns.

Node- and edge- exchangeability. A sequence $\mathcal{Z} = (z_1, \ldots, z_n)$ is called exchangeable if for any permutation μ, the joint probability of the permuted sequence remains the same, i.e. $\mathbb{P}[(z_1, \ldots, z_n)] = \mathbb{P}[(\mu(z_1), \ldots, \mu(z_n))]$. Before considering inductive graph sequences, we first define node- and edge-exchangeability on a fixed given graph $\mathcal{G}$.

Definition B.1 (Node-exchangeability)**.** Let $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{X}, \boldsymbol{y})$ be a graph where i-th row of $\boldsymbol{X}$ is the feature vector for node v_i, and similarly, i-th element of $\boldsymbol{y}$ is the label of that node. Let $\mu_V : \{1, \ldots, n\} \times \{1, \ldots, n\}$ with $n = |\mathcal{V}|$ be a permutation. We define $\mu_V(\mathcal{G})$ to be a relabeling of vertices $\mathcal{V}$ with $\mu_V(\mathcal{E}) = \big\{(\mu_V(v_i), \mu_V(v_j)) : \forall (v_i, v_j) \in \mathcal{E}\big\}$, $\mu_V(\boldsymbol{X})[i, \cdot] = \boldsymbol{X}[\mu_V(i), \cdot]$, and $\mu_V(\boldsymbol{y})[i] = \boldsymbol{y}[\mu_V(i)]$. The permuted graph is $\mu_V(\mathcal{G})(\mu_V(\mathcal{V}), \mu_V(\mathcal{G}), \mu_V(\boldsymbol{X}), \mu_V(\boldsymbol{y}))$. $\mathcal{G}$ is called node exchangeable if $\mathbb{P}[G = \mathcal{G}] = \mathbb{P}[G = \mu_V(\mathcal{G})]$, where G is sampled from a generative graph distribution.

Definition B.2 (Edge-exchangeability)**.** Let $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{X}, \boldsymbol{y})$ where $\mathcal{E} = \{e_1, \ldots e_m\}$ is the set of edges. Each edge e_k is defined as a pair $e_k = ((v_i, \boldsymbol{x}_i, y_i), (v_j, \boldsymbol{x}_j, y_j))$. For each node $v_i \in \mathcal{V}$ let $\boldsymbol{X}$ be the feature matrix with rows $\boldsymbol{x}_i$, and $\boldsymbol{y}$ be the vector of labels with entries y_i. Let $\mu : \{1, \ldots, m\} \times \{1, \ldots, m\}$ with $m = |\mathcal{E}|$ be a permutation. We define

$\mu(\mathcal{G})$ to be a relabeling of edges $\mathcal{E}$ with $\mu_E(\mathcal{E}) = \left\{e_{\mu_E(i)} : \forall e_i \in \mathcal{E}\right\}$. The permuted graph is $\mu_E(\mathcal{G})(\mathcal{V}, \mu_E(\mathcal{E}), \boldsymbol{X}, \boldsymbol{y})$. $\mathcal{G}$ is called edge exchangeable if $\mathbb{P}[G = \mathcal{G}] = \mathbb{P}[G = \mu_E(\mathcal{G})]$, where G is sampled from a generative graph distribution.

By an inductive sequence, we refer to a progressive sequence of graphs meaning that for each timestep t, the graph $\mathcal{G}_{t-1}$ is a subgraph of $\mathcal{G}_t$. Node- and edge-inductive sequences are defined as follows:

Definition B.3 (Node-inductive sequence)**.** A node-inductive sequence $\mathcal{G}_0, \mathcal{G}_1, \ldots$ is a sequence starting from an empty graph $\mathcal{G}_0 = (\emptyset, \emptyset)$. For each timestep t, the graph $\mathcal{G}_t$ is defined by adding a vertex v_t with all its connections to the vertices in the previous timestep. The vertex set is then $\mathcal{V}_t = \mathcal{V}_{t-1} \cup \{v_t\}$ and the edge set is the union of all previous edges $\mathcal{E}_{t-1}$ and any edge between v_t and $\mathcal{V}_{t-1}$; $\mathcal{E}_t = \mathcal{E}_{t-1} \cup (\cap_{i=1}^{\infty} \{e = (v_t, v_i) : e \in \mathcal{E}_i, v_i \in \mathcal{V}_{i-1}\})$.

Similarly an edge-inductive sequence is defined as follows:

Definition B.4 (Edge-inductive sequence)**.** An edge inductive sequence $\mathcal{G}_0, \mathcal{G}_1, \ldots$ starts from an empty graph $\mathcal{G}_0 = (\emptyset, \emptyset)$. For each timestep t, the graph $\mathcal{G}_t$ is defined by adding an edge $e_t = (v_i, v_j)$ to the graph at the previous timestep. Hence $\mathcal{E}_t = \mathcal{E}_{t-1} \cup \{e_t\}$, and $\mathcal{V}_t = \mathcal{V}_{t-1} \cup \left\{v_i, v_j\right\}$.

A node-inductive sequence is called node-exchangeable if for all graphs $\mathcal{G}_t$ in the sequence, $\mathcal{G}_t$ is node-exchangeable. Similarly, if all graphs in the sequence are edge-exchangeable, the sequence is also edge-exchangeable.

Note, an inductive sequence could be equivalentlty defined with respect to a "final" graph $\mathcal{G}_n$, where n can be infinity, by starting from an empty graph and adding a random node/edge at each timestep. If the final graph is node-exchangeable any node-inductive subgraph of it $\mathcal{G}' \subseteq \mathcal{G}_t$, $\mathcal{G}'$ is also node exchangeable. A similar argument holds for edge-inductive subgraphs of a "final" edge-exchangeable graph.

Permutation invariance and equivariance. A permutation invariant function f returns the same output for any permutation applied on its inputs, $f(z_1, \ldots, z_n) = f(z_{\mu(1)}, \ldots z_{\mu(n)})$ for any permutation μ. A function f is permutation-equivariant if permuting the input results in the same permutation on the output, i.e. for any $f(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$ we have $f(x_{\mu(1)}, \ldots x_{\mu(n)}) = (y_{\mu(1)}, \ldots y_{\mu(n)})$ for any μ. A permutation-equivariant GNN assigns the same predictions (and thus the same scores) to each node even when nodes/edges are relabeled.

Proof for Proposition 4.3. First assume $\mathcal{V}_{\text{eval}}^{(t)} = \mathcal{V}_t$ meaning that any node appeared at timestep t is either in the calibration or the evaluation set. Due to node-exchangeability, $\mathcal{V}_t$ and $\mathcal{V}_{\text{cal}}$ are exchangeable. Hence, we adapt the proof of Theorem 4.2. Now consider the general case where $\mathcal{V}' := \mathcal{V}_t \setminus \mathcal{V}_{\text{cal}} \neq \emptyset$. Any subset $\mathcal{A}'$ of an exchangeable set $\mathcal{A}$ is still exchangeable. For any permutation of $\mathcal{A}'$ there are k permutations in $\mathcal{A}$ all having the same ordering over elements of $\mathcal{A}'$. Since k is a constant for all permutations and function of $|\mathcal{A}|$ and $|\mathcal{A}'|$, any permutation in $\mathcal{A}'$ has the same probability. Which implies the exchangeability of its elements. This implies that with non-empty set $\mathcal{V}'$, $\mathcal{V}_{\text{cal}}$ and $\mathcal{V}_t \setminus \mathcal{V}'$ are still exchangeable. As a result, again we adapt Theorem 4.2 with the effect of $\mathcal{V}'$ being symmetric to the calibration and evaluation sets.

□

Proof for Theorem 4.4. We break the proof to two parts.

Part 1. For a fixed node v_i at timestep t_1 let $\alpha_i := \mathbb{E}[c_{i,t_1}]$. For any timestep t_2 with which v_i is existing in, we have $\mathbb{E}[c_{i,t_2}] = \alpha_i$.

Note that since v_i is fixed its expected coverage probability is not exactly $1-\alpha$. As in definition,

$$\alpha_i = \mathbb{P}\left[\mathbb{Q}\left(\alpha; \left\{s(v_j, y_j \mid \mathcal{G}_{t_1})\right\}_{v_j \in \mathcal{V}_{\text{cal}}}\right) 1 \leq s(v_i, y_i \mid \mathcal{G}_{t_1})\right]$$

Let $\mathcal{V}'_j = \mathcal{V}_{t_j} \setminus (\mathcal{V}_{\text{cal}} \cup \{v_i\})$. Node-exchangeability implies that $\mathcal{V}'_1$ and $\mathcal{V}'_2$ have same and also symmetric effect on all scores. Hence, with either of sets as context, the expected order of elements remains similar and $\mathbb{E}[c_{i,t_2}] = \alpha_i$. Moreover, for each node v_i and all t, $c_{i,t} \sim \text{Bernoulli}(\alpha_i)$.

Part 2. The coverage of any sub-partitioning $\mathbb{I}_T$ can be written as an average over a set of elements in $\boldsymbol{C}_T$. In other words define $\iota(v_i) = j \Leftrightarrow (v_i \in \mathcal{V}^j_{\text{eval}})$ we have

$$\text{Cov}(\mathbb{I}_T) = \frac{1}{|\mathbb{I}_T|}\sum_{i=1}^{|\mathbb{I}_T|} \boldsymbol{C}[i, \iota(v_i)] \overset{\text{Part 1}}{=} \sum_{i=1}^{|\mathbb{I}_T|} \boldsymbol{C}[i, T]$$

Following Proposition 4.3 we have

$$\mathbb{P}\left[\text{Cov}(\mathbb{I}_T) \leq \beta\right] = \frac{1}{|\mathbb{I}_T|}\sum_{i=1}^{|\mathbb{I}_T|} \boldsymbol{C}[i, \iota(v_i)] = 1 - \Phi_{\text{HG}}(i_\alpha; |\mathbb{I}_T| + n, n, i_\alpha + \lceil |\mathbb{I}_T| \cdot \beta \rceil)$$

□

Proof for Theorem 4.6. We divide the proof into two cases:

Graph where all nodes have degree 1. In this case $\mathcal{G}_t$ is a union of disjoint edges $(e_1, \ldots e_m)$ We divide $\mathcal{V}_t = \mathcal{V}_1 \cup \mathcal{V}_2$ including one and only one endpoint of each edge decided at random. The vertex set in this case has the same size as the edge set, and there is a one-to-one mapping between any permutation μ over the edge set to vertices in $\mathcal{V}_1$ or $\mathcal{V}_2$ since only one endpoint is present in each set. Hence $\mathbb{P}[\mathcal{E}] = \mathbb{P}\left[\mathcal{E}_\mu\right]$ implies $\mathbb{P}[\mathcal{V}_i] = \mathbb{P}\left[\mathcal{V}_{i,\mu}\right]_{i\in\{1,2\}}$. Similarly, selection $\mathcal{V}_{\text{cal}}$ decomposes to $\mathcal{V}_{\text{cal},1} \cup \mathcal{V}_{\text{cal},2}$ and again $\mathcal{V}_{\text{cal},i}$, and $\mathcal{V}_i \setminus \mathcal{V}_{\text{cal},i}$ are exchangeable due to node-exchangeability in $\mathcal{V}_i$. Any test node belongs to either of the subsequences which means is guaranteed via the CP applied to that subset.

The vertex set is divided into two subsets, and the intersection of the calibration set to each subset is exchangeable. Hence calibrating on each subset result in $1-\alpha$ coverage for all the nodes in the subset. Both calibrations have the same expected quantile threshold q. This is because each calibration node has $1/2$ probability of being included in each of the sets.

General graphs. We follow the same approach. We divide the endpoints of edges into two multi-sets. For each edge we run this division and this decision is made for each vertex $\deg(v_i)$ times. In each partition, each node v_i has the expected frequency $\deg(v_i)/2$. Via Lemma 4.5 we show that weighted CP is valid for each subsequence $\mathcal{V}_i$ and hence for $\mathcal{V}_t$. □

B.3 Limitations of NAPS

For the inductive scenario, NAPS (Clarkson, 2023) adapts the beyond exchangeability approach (Barber et al., 2022) without any computed bounds on the coverage gap (the distance between the coverage of the given non-exchangeable sequence and $1 - \alpha$). For each test node, calibration weights are assigned equal to 1 in case they are in the immediate neighborhood of the test node and 0 otherwise. In other words, this approach filters out the non-neighbor calibration nodes for each test node. This weight assignment is generalized to any k-hop neighbor while it is originally suggested to use $k = 1$ or 2. In a sparse graph, or with a limited calibration set, the probability of having calibration points in the immediate neighborhood is significantly low which leaves many of the test nodes with empty calibration sets. Fig. B.1 shows that for our experimental setup, until the end of the graph sequence, a notable proportion of the nodes are left without a prediction set. For the same reason, even for nodes in the neighborhood of the calibration set, the statistical efficiency is very low. Since this inapplicability is reported on a node-exchangeable sequence, it is independent of the prediction timestep – a node disconnected from the calibration set will remain disconnected as the graph grows.

Setting aside non-applicable nodes we compared NAPS with NodeEx CP in empirical coverage. Fig. B.2 shows that NAPS is significantly far from the coverage guarantee compared to NodeEx CP. Note that this experiment is very biased in favor of NAPS as it excludes nodes without a prediction set while evaluating the same nodes for NodeEx CP.

B.4 Additional Experiments

B.4.1 CP with Subgraph Sampling

Although the theory in § 4.4 concerns inductive graph sequences, we can leverage its results to transductive node-classification via subgraph sampling. For each test node v_{test}, we sample K inductive subgraphs all including $\{v_{\text{test}}\} \cup \mathcal{V}_{\text{cal}}$. Considering Theorem 4.4, in each subgraph, calibration scores and test scores are exchangeable. This leaves K prediction sets all with a coverage guarantee of $1 - \alpha$. We consider each prediction set as a vote for all its elements. Each label y' will be selected in the final prediction set with a Bernoulli experiment with parameter $p = \sum_{i=1}^{K} \mathbf{1}[y_j \in C_i(v_{\text{test}})]/K$. Again the resulting prediction set contains the true label with $1 - \alpha$ probability. It is important to note that Theorem 4.4 does not imply any result about the probability of a node being covered in all of K prediction sets. Specifically, the probability of a node being covered in all prediction sets simultaneously is less than (or equal to) the probability of the same event in each single CP. As shown in Fig. B.3 the subgraph Bernoulli prediction sets result in the same coverage as standard CP for transductive node-classification. Additionally, the union and intersection of CPs respectively result in over-coverage and under-coverage.

B.4.2 Set Size and Singleton Hit Ratio

In § 4.6 we focused on empirical coverage as the main evaluation criteria since our goal is to recover the conformal guarantee. When the guarantee holds, this metric is often reported as a sanity check. In that case, the average prediction set size (efficiency) is considered as a direct indicator of how efficient (and therefore useful) prediction sets are. Another important metric called *singleton hit ratio* is the frequency of singleton sets covering the true label. In Fig. B.4 we show the coverage matrix alongside the prediction

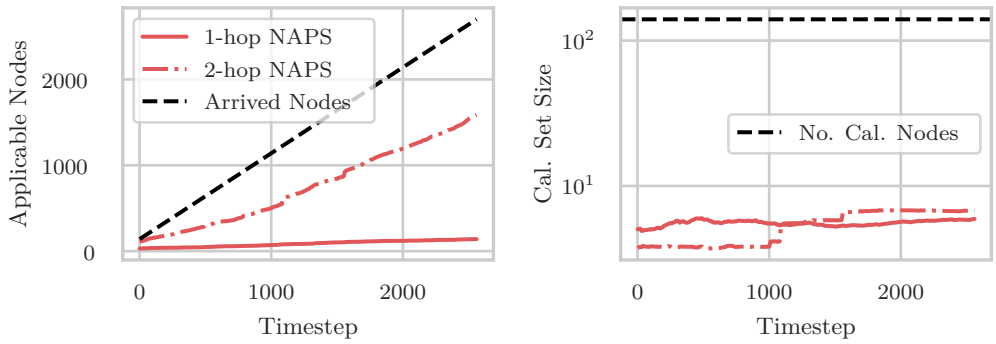

Figure B.1: [Left] The proportion of nodes without a prediction set due to empty calibration on `CoraML` dataset. The plot shows the inapplicability of k-hop NAPS with $k \in \{1, 2\}$. [Right] The average size of each node's filtered calibration set. Note that this plot excludes non-applicable nodes.

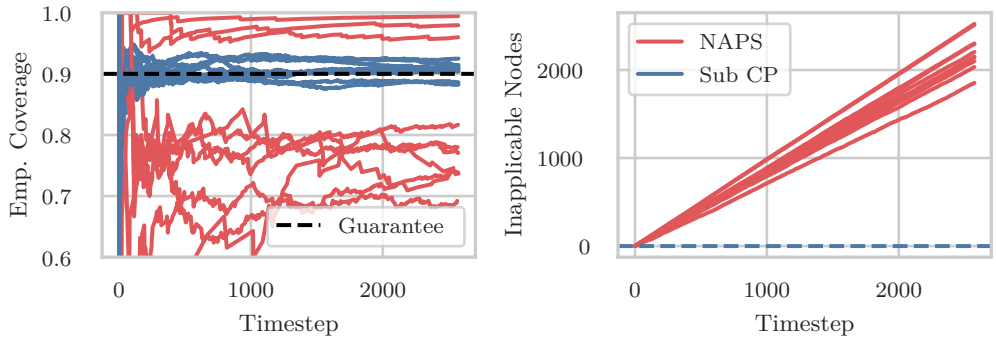

Figure B.2: [Left] Comparison of NAPS and NodeEx CP in empirical coverage for `CoraML` dataset and `GCN` model across different permutations of nodes. [Right] The number of nodes without prediction set for different permutations on the same dataset/model.

set size and singleton hits indicator for each node at each timestep. As shown in the figure, since the standard CP breaks the guarantee toward over-coverage, it results in larger prediction sets on average and hence, a lower number of singleton hits. The same result is also shown in Fig. 4.4 over different timesteps.

B.4.3 Different Score Functions

The threshold prediction sets (TPS) approach (Sadinle et al., 2018) directly takes the softmax output as a conformity score $s(\boldsymbol{x}, y) = \pi(\boldsymbol{x})_y$, where $\pi(\boldsymbol{x})_y$ is the predicted probability for class y. Although TPS produces small prediction sets, its coverage is biased toward easier examples leaving the hard examples under-covered (Angelopoulos and Bates, 2021). Romano et al. (2020) define *adaptive* prediction sets (APS) with a score function defined as $s(\boldsymbol{x}, y) := -\big(\rho(\boldsymbol{x}, y) + u \cdot \pi(\boldsymbol{x})_y\big)$. Here $\rho(\boldsymbol{x}, y) := \sum_{c=1}^{K} \pi(\boldsymbol{x})_c \mathbb{1}\left[\pi(\boldsymbol{x})_c > \pi(\boldsymbol{x})_y\right]$ 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 (Stutz et al., 2021). APS returns the smallest prediction sets satisfying conditional coverage if the model returns the ground truth conditional probability $p(y \mid \boldsymbol{x})$, otherwise Barber et al. (2019a) show that achieving conditional coverage is impossible without strong unrealistic assumptions. Here we

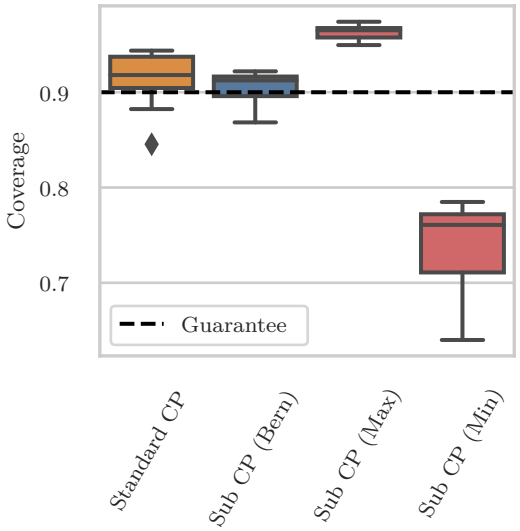

Figure B.3: Comparison between standard CP, Bernoulli prediction sets from NodeEx CP on random subgraphs, the union, and intersection of CPs.

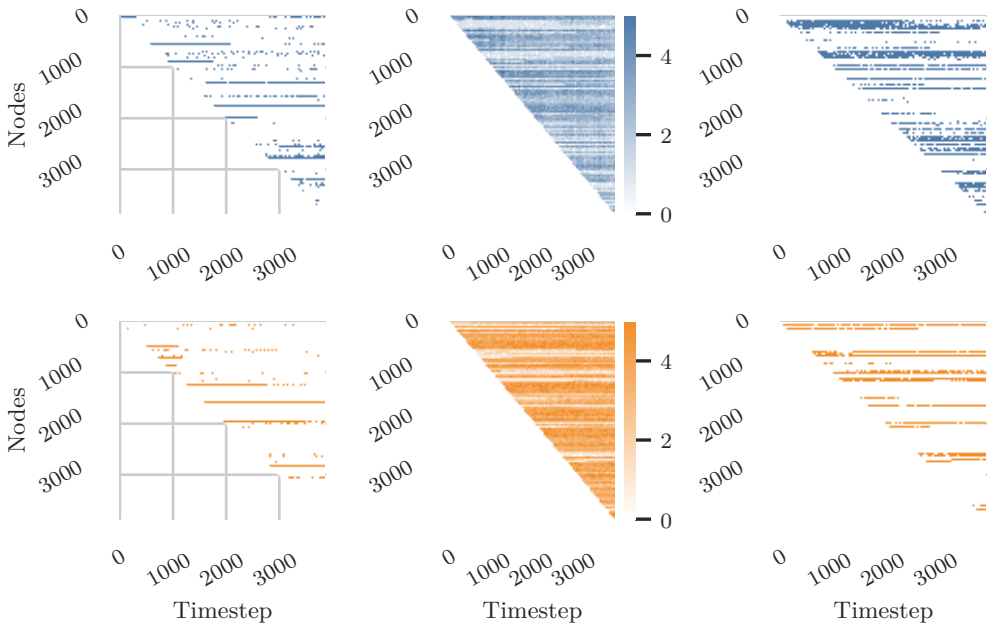

Figure B.4: [Left column] The coverage matrix $1 - \boldsymbol{C}$ (colored points show miscoverage). [Middle column] The prediction set size for each point at each timestep. [Right column The matrix of singleton hits. A cell is colored if the node at the timestep is predicted with a singleton set covering the true label. [Upper row] The NodeEx CP approach. [Lower row] The standard CP. The result is shown for `CiteSeer` dataset and `GCN` model.

use APS as baseline scoring function, but our method is orthogonal to this choice and works with any score.

Although we used APS as the base score function for evaluation, our results are general to any other scoring function. Here we evaluate our approach with two other score functions called threshold prediction sets (TPS) (Sadinle et al., 2018), and diffused adaptive prediction sets DAPS Zargarbashi et al. (2023a). TPS (Fig. B.5 - left) simply applies the softmax function on model's results (logits) and uses it as the conformity

score. Although it produces small prediction sets its coverage is biased toward easier examples (Angelopoulos and Bates, 2021). DAPS (Fig. B.5 - right) works as a structure-aware extension scoring over APS. It diffuses conformity scores over the network structure to leverage the uncertainty information in the neighborhood and produce a more efficient prediction set. Since DAPS also incorporates the structure in the scores' space (in addition to the implicit effect via message passing) it is even more influenced by the changes in the graph structure while using standard CP. However, as shown in the Fig. B.5, we can recover the coverage guarantee via NodeEx CP regardless of the utilized score function.

B.4.4 Different Models and Initial Splits

The coverage guarantee in CP holds regardless of the model structure. CP uses the model as a black-box and just performs the quantile calibration over the output of the model which is the input conformity score function. However, better model structure, training procedure, etc are reflected in other metrics like set size and singleton hits. As shown in Fig. B.6, NodeEx CP results in similar coverage values for all the models while standard CP results in different coverage for each model (note the significant distance between structure-aware models and MLP). As MLP does not take the adjacency structure into account, its empirical coverage is still guaranteed even with the standard CP. Note that different models result in different efficiencies in the prediction sets. Fig. B.7 shows the same experiment conducted on edge-exchangeable sampling. Again similar results are observed for edge-exchangeable sequences.

As pointed out in (Shchur et al., 2018), GNNs are sensitive to the initial train/validation sampling. This however does not impact the coverage of NodeEx (and EdgeEx) CP as it is guaranteed agnostic to the model and the initial split. However similar to different model architectures, different initial splits also affect the model's accuracy which is reflected in the efficiency of the prediction set. Fig. B.8 verifies this fir the node-exchangeable sampling. We also evaluated our method compared to standard CP for edge-exchangeable sequences over different initial sampling. The result is in Fig. B.9.

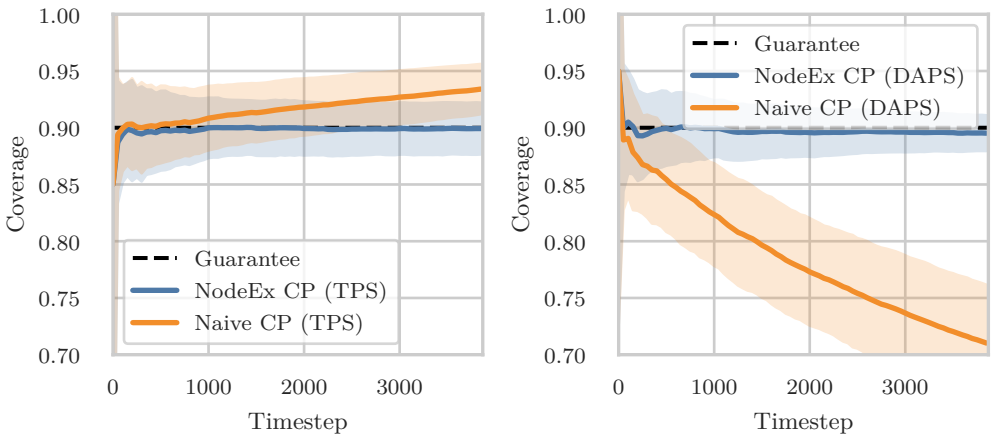

Figure B.5: [Left] Comparison of naive CP and NodeEx CP with TPS score function and [Right] DAPS score function. Results are for `CiteSeer` dataset and `GCN` model.

B.4.5 Other Experiments

Experiment corresponding to Fig. 4.1 We performed a node-exchangeable sampling on `CiteSeer` dataset. At each timestep, we ran the model (`GCN` trained on the train/val nodes) on the existing subgraph extracting conformal scores for all existing test nodes. The heatmap in Fig. 4.1 (left) shows the distribution of test scores (sorted) at each timestep. An oracle CP (with access to test nodes and their label) will choose the line labeled as "ground truth" as the conformal threshold since it is the exact α quantile of existing test scores. This is shown as the ideal reference to evaluate each CP approach with respect to it. We draw the threshold computed by standard CP and NodeEx CP. It is shown that the NodeEx CP picks thresholds close to the ideal line while the naive CP shifts from it due to the message passing with the new nodes.

We drew the distribution of conformity score for calibration nodes at some selected times (sampled with equal distance across from all timesteps) and compared it with calibration scores used by standard CP in Fig. 4.1(upper right). A distribution shift is clearly observable. We showed this shift in Fig. 4.1(upper right) by plotting the distibutions across various timesteps. The corresponding timestep for each distribution is marked in the left subplot with the same color. Here we computed the EMD (earth mover distance) between the scores of standard CP and conformity scores (of true labels) for calibration and test points. It is shown that the distribution shift is increasing by the number of new nodes introduced to the graph. This shift is almost similar in calibration and test. One source of the noise in thresholds and EMD is the uniform random value in APS scoring function. EMD is smoothed over 10 steps.

Large Datasets. We ran NodeEx CP and compared it with the baseline for two large datasets: Flickr (Young et al., 2014), Reddit2 (Zeng et al., 2019). Our results for Flicker and Reddit2 datasets under node exchangeable sampling are shown in Fig. B.10. Both NodeEx CP and naive CP show similar results. In the edge-exchangeable sampling Fig. B.11 shows a significant deviation from the guarantee for NodeEx CP and naive CP, while EdgeEx CP maintains a valid coverage.

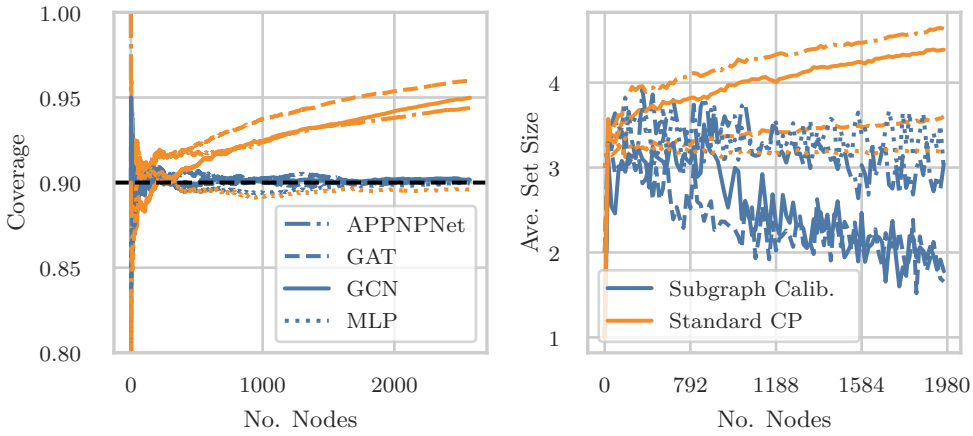

Figure B.6: [Left] The empirical coverage for different models. [Right] The average set size. The results are shown for the `CoraML` dataset with node-exchangeable sampling.

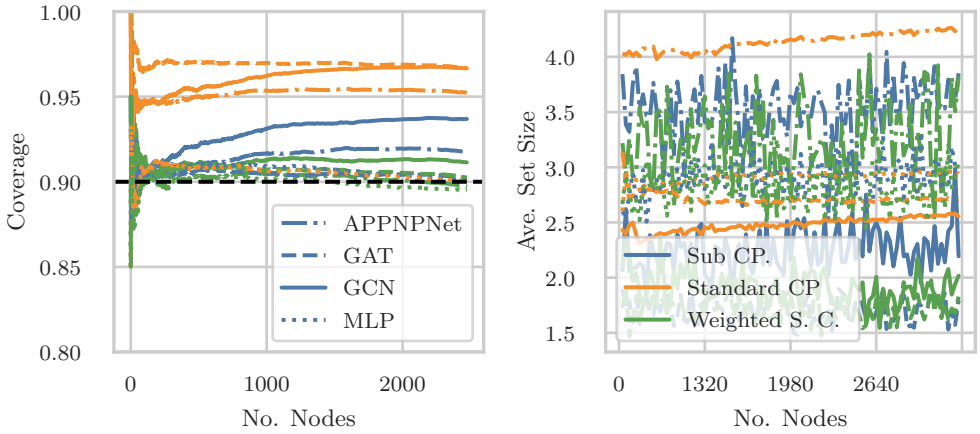

Figure B.7: [Left] The empirical coverage for different models. [Right] the average set size. The results are shown for the `CoraML` dataset with edge-exchangeable sampling.

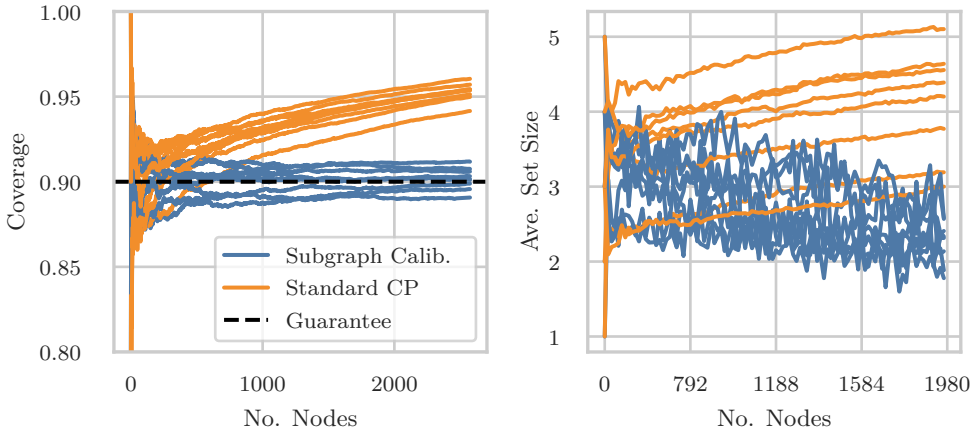

Figure B.8: [Left] The empirical coverage of standard CP and NodeEx CP with different initial train/val splits. [Right] The average set size. The result is shown for `CoraML` dataset and `GCN` model for node-exchangeable samplings.

Class-conditional coverage. Conformal prediction comes with a marginal guarantee which means that it does not guarantee conditional to group or class. In Fig. B.12 we compare this metric. In most of the classes, we observe the standard CP to be closer to the guaranteed line.

Effect of calibration set size. In non-graph data as mentioned in § 4.2 the coverage probability follows a Beta distribution. In graph data with a fixed number of test nodes (see Theorem 4.2) the distribution of coverage over test nodes follows a collection of hyper-geometric distributions. In both scenarios, there is a possible variance around predefined $1 - \alpha$ probability which is a function of calibration set size. As the number of calibration nodes increases, the distribution of coverage probability concentrates around $1 - \alpha$. Since in our experiments we followed a realistic setup (not allowing calibration set to be larger than training set) there is a variance observed around the guarantee line – each line converges to a value close to but not exactly equal to $1 - \alpha$. Fig. B.13 shows that as we increase the calibration set size the result concentrates around

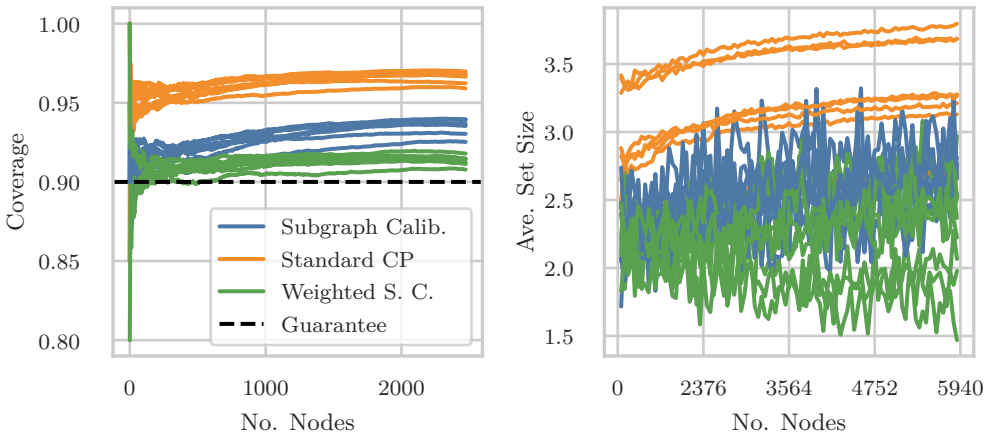

Figure B.9: [Left] The empirical coverage of standard CP and EdgeEx CP with different initial train/val splits. [Right] The average set size. The result is shown for `CoraML` dataset and `GCN` model for edge-exchangeable samplings.

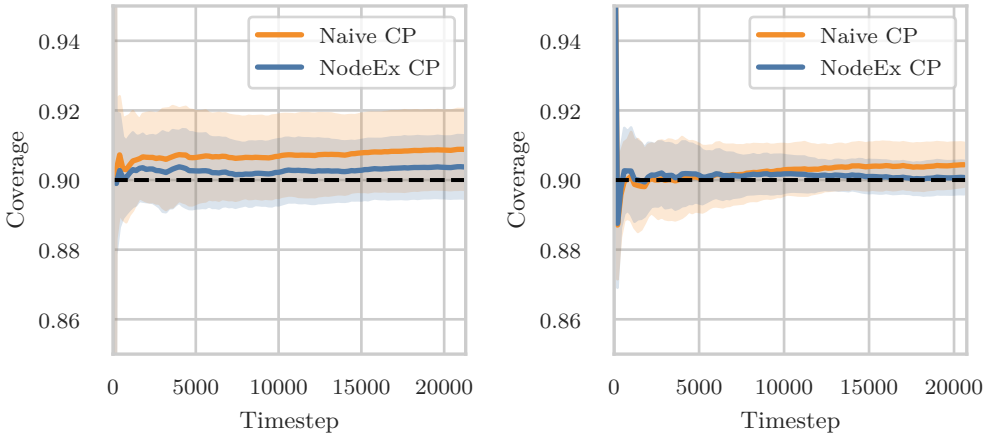

Figure B.10: Coverage Results for [Left] `Flickr`, and [Right] `Reddit2` dataset under node-exchangeable sampling. The results are shown for `GCN model`.

the guarantee.

B.5 Supplementary Details of Experiments

Datasets. Table B.1 provides specifications of datasets used in our experimental evaluations, including the number of nodes, edges, and homophily. The details of label sampling are provided in § 4.6. For each experiment, we ran each CP approach on 10 different sequences in node-exchangeable and 15 different sequences on edge-exchangeable setup. All transparent lines in plots show the result for one sequence. In those experiments, the solid line shows the average of sequences at each timestep.

Models. For all architectures, we built one hidden layer of 64 units and one output layer. We applied dropout on the hidden layer with probability 0.6 for `GCN`, and `GAT`, 0.5 for `APPNPNet`, and 0.8 for `MLP`. For `GAT` we used 8 heads. We trained all models with categorical cross-entropy loss, and `Adam` optimizer with L_2 regularization.

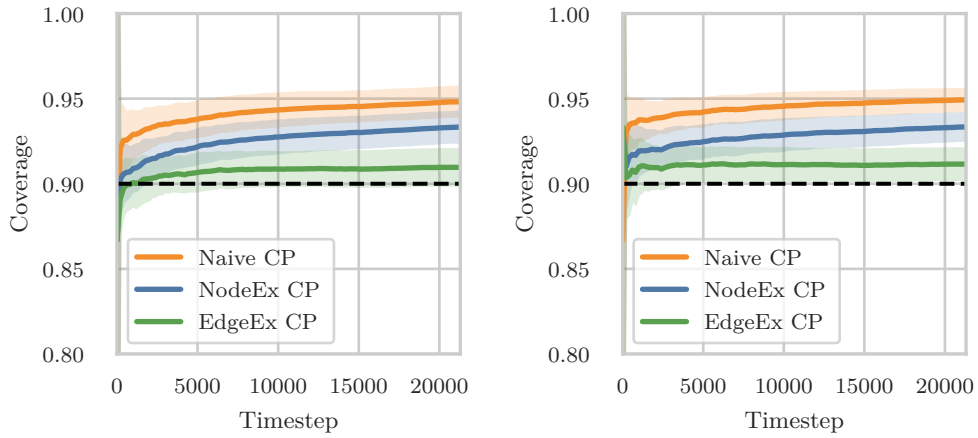

Figure B.11: Coverage Results for [Left] `Flickr`, and [Right] `Reddit2` dataset under edge-exchangeable sampling. The results are shown for `GCN model`.

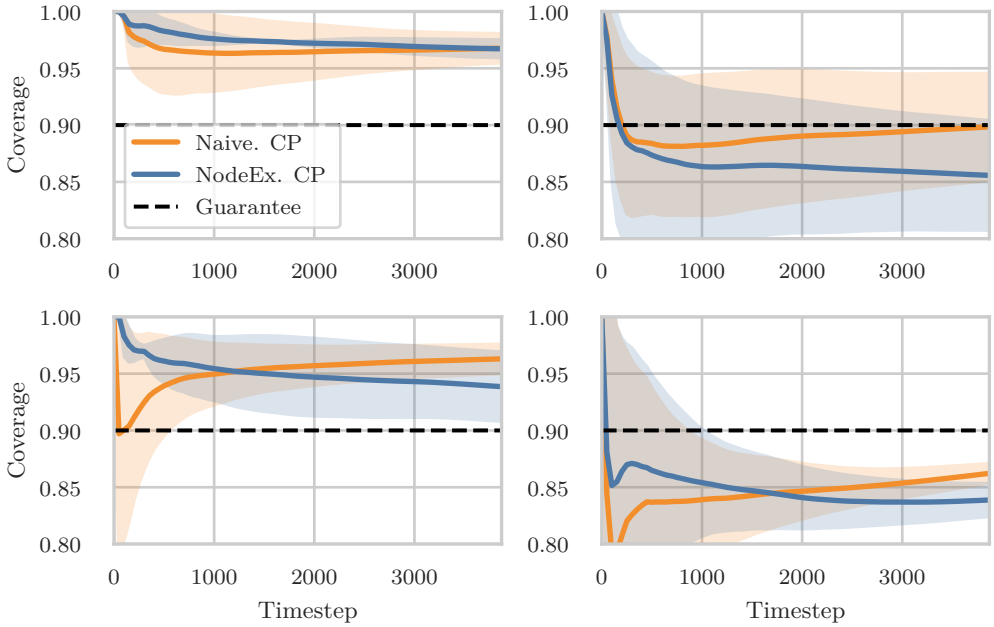

Figure B.12: NodeEx CP and standard CP in class-conditional coverage. The result is for `CiteSeer` dataset and `GCN` model. Plots show classes 1 and 2 [Upper row left to right], 3 and 4 [Lower row].

Adaptive and constant coverage. Despite many studies on CP, Zargarbashi et al. (2023a) chooses an adaptive value for $1 - \alpha$ which is conditional to the model accuracy. This is a suitable choice when the main comparison is based on set size and other efficiency-related metrics. The main supporting idea is that efficiency should be compared with a guarantee that is not trivially achievable by the normal model prediction. However, our main concern is to recover the guarantee meaning that the empirical coverage is the main comparison criteria. Hence we choose 0.9 as the required coverage for all datasets and models regardless of the accuracy. Furthermore, NodeEx CP works for any user-specified coverage guarantee including model-conditional values.

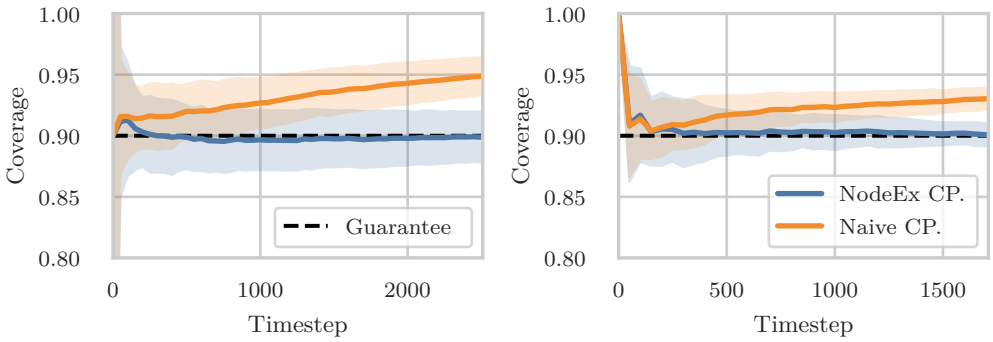

Figure B.13: Effect of different calibration set sizes on the concentration of coverage probability. The results are on `Cora-ML` dataset and for 200 [Left] and 1000 [Right] calibration nodes.

Table B.1: Statistics of the datasets.

Dataset Name	Vertices	Attributes	Edges	Classes	Homophily
`CoraML`	2995	2879	16316	7	78.85%
`PubMed`	19717	500	88648	3	80.23%
`CiteSeer`	4230	602	10674	6	94.94%
`Coaut. CS`	18333	6805	163788	15	80.80%
`Coauth. Physics`	34493	8415	495924	5	93.14%
`Amz. Comp.`	13752	767	491722	10	77.72%
`Amz. Photo`	7650	745	238162	8	82.72%

B.6 Related Works

Standard conformal prediction. CP is introduced by Vovk et al. (2005) and further developed in Lei and Wasserman (2014); Shafer and Vovk (2008); Barber (2020). Different variants of CP are yet defined ranging in the trade-off between statistical and computational efficiency. While full conformal prediction requires multiple training rounds for each single test point, Jackknife+ Barber et al. (2019b), and *split* conformal prediction sacrifice statistical efficiency, defining faster and hence more scalable CP algorithms. A comprehensive survey of CP can be found in Angelopoulos et al. (2024). In this study, we focus on split conformal prediction.

Further contributions in this area include generalization of the guarantee from including the true label to any risk function Angelopoulos et al. (2022), improving the efficiency (reducing the average set size) by simulating calibration during training Stutz et al. (2021), limiting the false positive rate Fisch et al. (2022), etc.

CP without exchangeability. Standard CP requires exchangeability for datapoints and assumes the model to treat datapoints symmetrically. The latter assumption ensures the exchangeability of datapoints even after observing the fitted model. Tibshirani et al. (2019b) extend CP to cases where exchangeability breaks via different $p(X)$ for calibration set and the test data, given $p(Y \mid X)$ is still the same. In this case, CP is adapted via reweighing the calibration set using the likelihood ratio to compare the training and test covariate distributions. This requires the high-dimensional likelihood to be known or well-approximated. Barber et al. (2022) does neither rely on known

likelihood between the original and shifted distribution nor the symmetry in the learning algorithm and proposes a coverage gap based on the total variation distance between calibration points and the test point. The main upperbound on the coverage requires a predefined fixed weight function. To adapt this bound to data-dependent weights, the d_{TV}-distance must be computed conditional to the assigned weights.

CP for graphs. Recently adapting CP to graphs got increasing attention. Wijegunawardana et al. (2020) adapted conformal prediction for node classification to achieve bounded error. Clarkson (2023) assumed exchangeability violated for node classification hence adapting weighted exchangeability without any lowerbound on the coverage, however, Zargarbashi et al. (2023a) and Huang et al. (2023a) proved the applicability of standard CP for transductive setting in GNNs. Moreover, Huang et al. (2023a) proposed a secondary GNN trained to increase the efficiency of prediction sets, and Zargarbashi et al. (2023a) stated that in homophily networks, diffusion of conformal score can increase the efficiency significantly.

For the inductive GNNs, the only work we are aware of is the neighborhood APS (NAPS) approach which is an adaptation of weighted CP Clarkson (2023), however, it is shown that NAPS can not be applied on a significant fraction of nodes if the network is sparse or the quantity of calibration nodes are limited Zargarbashi et al. (2023a) (this number increases to over 70% in some benchmark datasets). As our approach is adaptable to sparse networks and works for any calibration set size, we do not compare our approach with NAPS.

C Supplementary Material: Robust yet Efficient Conformal Prediction

C.1 More On Conformal Prediction

Conformity vs. non-conformity scores. As mentioned in § 5.2, conformal prediction requires defining 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, we include labels to the prediction set if their score is *less* than the threshold. 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 § 5.2 we mentioned that conformal prediction returns guaranteed sets regardless of the scoring function employed. Specifically, any score function maintaining the exchangeability (between calibration and test) is viable. In brief, the exchangeability of random variable $Z_1, \ldots, Z_n$ 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 $p(Z_1, \ldots, Z_n) = p(Z_{\psi(1)}, \ldots, Z_{\psi(n)})$. Assuming the calibration set to be exchangeably sampled from the data distribution, any permutation equivariant transformation on the data still preserves the 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., 2023a). 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. 42). 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. (2023a) show that the coverage probability, $\text{Cov}(\mathcal{D}) = (1/|\mathcal{D}|) \sum_{(\boldsymbol{x}_i, y_i)\in\mathcal{D}} \mathbf{1}[y_i \in C(\boldsymbol{x}_i)]$ is distributed as

$$\mathbb{P}\left[\text{Cov}(\mathcal{D}) \leq t\right] = 1 - \Phi_{\text{HG}}(\lfloor \alpha(n+1) \rfloor - 1; M+N, N, \lceil Mt \rceil + \lfloor \alpha(n+1) \rfloor) \tag{89}$$

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 $\hat{q}$ within the domain of the score function (as in poisoning), the new threshold will correspond to another quantile $\beta = \mathbb{Q}^{-1}(\hat{q}; \mathcal{D}_{\text{cal}})$ and the coverage will be similarly concentrated around $1-\beta$.

Access to a large calibration set (e.g. 1000 points) is unrealistic. Even in this case, for a large set of labeled points, there is an open question of whether to use a portion of it for training the model toward a 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 $1-\alpha$. 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.

C.1.1 Implementation Details

We based our implementation on PyTorch (Paszke et al., 2019) and Pytorch Geometric (Fey and 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).

Our code and reproducible experiments are provided in the supplementary material.

C.2 Faster Evasion-Robustness via Calibration-time Bound

The evasion-robust CP algorithm (see § 5.5) 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. 50) 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.3). 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 $O(n \times t_{\text{MC}})$ time for calibration, including the quantile computation. Then, for each test point we need $O(c \times t_{\text{MC}} \times t_{\text{bound}})$ time.

We define a computationally more efficient and robust alternative built upon Proposition 5.5 in which we offload the computational overhead from the test set to the calibration set. Proposition 5.5 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 $O(n \times t_{\text{MC}} \times t_{\text{bound}})$ for

calibration, and $O(c \times t_{\text{MC}})$ 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 C.1 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 C.1: 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 $O(n \times t_{\text{MC}})$	0.79 $O(n \times t_{\text{MC}} \times t_{\text{b}})$	204
Testing	2.93 $O(m \times c \times t_{\text{MC}} \times t_{\text{b}})$	0.15 $O(m \times c \times t_{\text{MC}})$	100
Total	3.08	0.94	

C.3 Technical Details On Randomized Smoothing

RSCP uses the closed-form solution to Eq. 49 as an upperbound on the score function within the L_2 perturbation radius (details are in § C.6). 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 $\mathbb{P}[\xi(\boldsymbol{x}) = \boldsymbol{z}]$ 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 $\delta \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ 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. 52 in § 5.4).

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

$$\mathbb{P}[\xi(\boldsymbol{x})[i] \neq \boldsymbol{x}[i]] = p_{\boldsymbol{x}[i]} \tag{90}$$

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

$$\mathbb{P}[\xi(\boldsymbol{x})_i = k] = \begin{cases} \left(\frac{p_0}{K-1}\right)^{(\boldsymbol{x}[i] \neq k)} (1 - p_0)^{(\boldsymbol{x}[i] = k)} & \boldsymbol{x}[i] = 0 \\ \left(\frac{p_1}{K-1}\right)^{(\boldsymbol{x}[i] \neq k)} (1 - p_1)^{(\boldsymbol{x}[i] = k)} & \boldsymbol{x}[i] \neq 0 \end{cases} \tag{91}$$

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. 49 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 $\frac{\mathbb{P}[\xi(\boldsymbol{x})=\boldsymbol{z}]}{\mathbb{P}[\xi(\tilde{\boldsymbol{x}})=\boldsymbol{z}]} = c_i$ for some constant c_i. Let $t_i = \mathbb{P}[\xi(\boldsymbol{x}) \in \mathcal{R}_i]$ and $\tilde{t}_i = \mathbb{P}[\xi(\tilde{\boldsymbol{x}}) \in \mathcal{R}_i]$ for for each region $\mathcal{R}_i$. Then Eq. 49 is equivalent to:

$$\max_{\boldsymbol{h}} \boldsymbol{h}^T \tilde{\boldsymbol{t}} \quad \text{s.t.} \quad \boldsymbol{h}^T \boldsymbol{t} = p, \quad 0 \leq \boldsymbol{h} \leq 1 \tag{92}$$

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

$$c_i = \left[\frac{p_0}{1-p_1}\right]^{i-r_d} \left[\frac{p_1}{1-p_0}\right]^{i-r_a} \tag{93}$$

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. 50 as the following linear program:

$$\max_{\boldsymbol{H}} b_m - \boldsymbol{H}\tilde{\boldsymbol{t}}\boldsymbol{d} \quad \text{s.t.} \quad \boldsymbol{H}\boldsymbol{t} = \boldsymbol{p}, \quad 0 \leq \boldsymbol{H} \leq 1 \tag{94}$$

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 $p_i = \mathbb{P}[s(\xi(\boldsymbol{x}), y) \leq b_i]$. 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. 94 can be decomposed into $m-1$ separate LPs similar to Eq. 92 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. 48. 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. 92 and similarly for Eq. 94.

C.4 Supplementary to Theoretical Support

C.4.1 Proofs

Proof of Proposition 5.3. Given the exchangeability of $(\boldsymbol{x}_{n+1}, y_{n+1})$ with the calibration set, Eq. 42 holds for the clean point. Since $\boldsymbol{x}_{n+1} \in \mathcal{B}(\tilde{\boldsymbol{x}}_{n+1})$ 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

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

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

Proof of Proposition 5.4. By definition $\mathcal{D}_{\text{cal}} \in \mathbb{B}_{k,\mathcal{B}}(\mathcal{D}_{\text{cal}})$ which means that q_α is a feasible solution to the problem and we have $q^* \leq q_\alpha$. It follows that $C_\alpha(\boldsymbol{x}) \subseteq \overline{C}_\alpha(\boldsymbol{x})$ where $C_\alpha(\boldsymbol{x}) = \{y_i : s(\boldsymbol{x}, y_i) \geq q_\alpha\}$ and $\overline{C}_\alpha(\boldsymbol{x}) = \{y_i : s(\boldsymbol{x}, y_i) \geq q^*\}$. Since $\mathbb{P}\left[y_{n+1} \in C_\alpha(\boldsymbol{x})\right] \geq 1 - \alpha$ due to exchangeability it follows that $\mathbb{P}\left[y_{n+1} \in \overline{C}_\alpha(\boldsymbol{x})\right] \geq 1 - \alpha$.

□

Proof of Proposition 5.5. Setting $\underline{q}_\alpha = \mathbb{Q}\left(\alpha; \left\{\underline{s}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}\right\}\right)$ we have:

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

Alternatively, The vanilla CP is calibrated with quantile

$$q_\alpha = \mathbb{Q}\left(\alpha; \{\hat{s}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}\}\right)$$

. The probability of a given potentially perturbed $\tilde{\boldsymbol{x}}_{n+1}$ being covered is:

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

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

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

□

Proof of Proposition 5.6. 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:

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

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 $\left\{\underline{s}_{\text{cdf}}(\boldsymbol{x}_i, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}\right\}$ 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 $\mathbb{P}\left[\underline{s}_{\text{cdf}}(\boldsymbol{x}_{n+1}, y_{n+1}) \geq \underline{q}_{\alpha}\right] \geq 1 - \alpha + \eta$. 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

$$\mathbb{P}\left[y_{n+1} \in \overline{C}_{\alpha+}\right] = \mathbb{P}\left[\hat{s}_{+}(\boldsymbol{x}_{n+1}, y_{n+1}) \geq \underline{q}_{\alpha+}\right] \geq 1 - \alpha$$

□

C.5 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;

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

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

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

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 $F(b_i) = \mathbb{P}[z \leq b_i]$ and $F_m(b_i) = \sum_{j=1}^{m} \mathbf{1}[z_j \leq b_i]$,

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

The above inequality holds simultaneously for all b_i.

For Eq. 49 we use the Bernstein inequality as is has shown a better empirical result compare to Hoeffding's inequality. For Eq. 50 we use the DKW inequality to find confidence intervals the empirical CDF.

Error correction in Eq. 49 and Eq. 50 To find the upper (or lower) bound in Eq. 49, 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. 50 we use the Dvoretzky–Kiefer–Wolfowitz inequality to find an upper (or lower) CDF. Since in the Eq. 51 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.

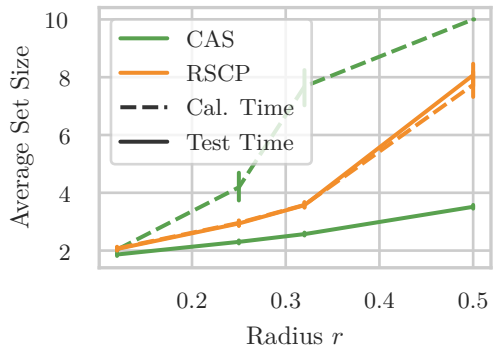

Figure C.1: Comparison of CAS and RSCP for faster (calibration-time) and test-time error correction.

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. 49 (RSCP) or Eq. 50 (CAS) with the error corrected estimate. For both methods, the following holds:

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

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. C.1, CAS benefits significantly from calibration-time robustness. The reason is that the CDF bound (Eq. 50) performs significantly better than the mean bound (Eq. 49) 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. 5.4 we choose the best performance of each method in either calibration- or test-time robustness with error correction.

C.6 Details on RSCP

C.6.1 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:

$$s_{\text{RSCP}}(x, y) = \Phi^{-1}(\mathbb{E}[s(\xi(x), y)]) \tag{95}$$

RSCP compute the α-quantile q_α of the new calibration scores (Eq. 95) and compares each score with the modified threshold $q_\alpha - r/\sigma$, where r is the radius of the l_2 ball and σ is the scale of the smoothing distribution. We can equivalently add the additional r/σ term to test scores and compare the augmented score with 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

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

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 $p = \mathbb{E}[s(\xi(\tilde{\boldsymbol{x}})]$ we see that this is equivalent to the Gaussian $\overline{s}_{\text{mean}}$ upper-bound defined in § 5.4.

C.6.2 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 $\overline{s}_{\text{cdf}}(\tilde{\boldsymbol{x}}, y)$.

C.7 Technical Details on Poisoning Certificate

Feature poisoning. The solution of the optimization problem in Eq. 45 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.[16] 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 § 5.4. This is a sound relaxation and the optimal q^* of the relaxed problem is smaller or equal than the q^* 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:

$$
\begin{aligned}
q^* = \quad & \min_{s_i, q} \quad q \\
s.t. \quad & \forall \tilde{s}_i : \underline{s}_i \leq s_i \leq \overline{s}_i \\
& t_i := \mathbf{1}[s_i \leq q], \quad \sum_{i=1}^{n} z_i \leq \lfloor \alpha n \rfloor, \quad \text{and} \quad \sum_{i=1}^{n} (1 - t_i) \leq \lceil (1-\alpha) n \rceil \\
& b_i := \mathbf{1}[s_i \neq \tilde{s}_i], \quad \sum_{i=1}^{n} b_i \leq k
\end{aligned}
\tag{96}
$$

In Eq. 97, 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. 47 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

$$
\begin{aligned}
q^* = \quad & \min_{q, \boldsymbol{C} \in \{0,1\}^{n \times c}} \quad q \\
\text{s.t.} \quad & \boldsymbol{C}\mathbf{1}^{c \times 1} = \mathbf{1}^{n \times 1} \\
& \boldsymbol{r} = (\boldsymbol{C} \odot \boldsymbol{S}) \times \mathbf{1}^{c \times 1} \\
& \sum_i \boldsymbol{C}[i, y_i] \geq n - k \\
& z_i := \mathbf{1}[r_i \leq q], \quad \sum_{i=1}^{n} z_i \leq \lfloor \alpha n \rfloor, \quad \text{and} \quad \sum_{i=1}^{n} (1 - z_i) \leq \lceil (1-\alpha) n \rceil
\end{aligned}
\tag{97}
$$

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.

[16] Note, there is a typo in Eq. 45 where $\mathbf{1}[\tilde{\boldsymbol{x}}_i \neq \boldsymbol{x}] \leq k$ should be $\sum_i \mathbf{1}[\tilde{\boldsymbol{x}}_i \neq \boldsymbol{x}_i] \leq k$.

C.8 Robustness to Poisoning and Evasion Attacks Combined

In subsubsection 5.3.2 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. 45, and Eq. 47, 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 $C(\boldsymbol{x}) \subseteq \overline{C}(\tilde{\boldsymbol{x}})$.

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 $\hat{q}$ 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.

C.9 Time and Space Complexity

Our robust CP approach breaks down into four subroutines (i) computing the score function, (ii) estimating expectations for randomized smoothing, (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 $O(1)$ step. The computation of the conformity score depends on the choice of this function. TPS takes $O(K)$ (K is the number of classes) to compute the categorical distribution via softmax function. APS score function takes an additional $O(K)$ steps to sort the class probabilities and compute the summation of confidences (see § 5.2 for the definition of the score function). 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 $O(n)$ steps (median computation) and the prediction sets take $O(C)$ 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 $O(N \times M)$ steps to compute the mean of M Monte-Carlo samples.

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 $O(S \times R)$ 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 $O(NM)$ additional steps for calibration on smooth scores and $O(MK + K \cdot t_b)$ for constructing the prediction sets. We also

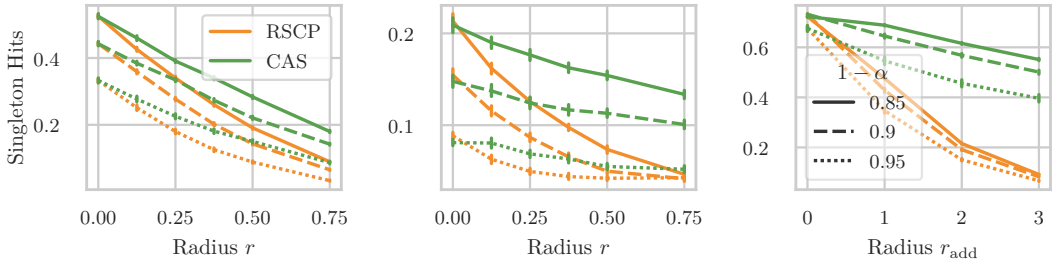

Figure C.2: Singleton hit ratio of CAS and RSCP under evasion for (from left to right) `CIFAR-10` with APS, `ImageNet` with TPS, and `Cora` with APS.

proposed a faster way to provide robust prediction sets in § C.2. 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 $O(MK)$ for test time (per test datapoint), and increases the calibration time complexity to $O(N \cdot t_b)$. This procedure decreases the number of steps in total. Table C.1 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 $O(NM + M \cdot t_b)$ 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.

C.10 Supplementary To Experiments

C.10.1 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 L2 norm smoothing paradigm and applied various noise levels, identifying the model that delivered optimal performance based on findings from Cohen et al. (2019). In the primary text results, a skip parameter of 10 was set for the `CIFAR-10` dataset, leading to the evaluation of roughly 1000 samples. Meanwhile, 500 data points are used from the sampling of every 100-image from the `ImageNet` dataset. Noise variance settings used were 0.25 sigma for `CIFAR-10` and 0.5 sigma for `ImageNet`. During the Monte Carlo sampling phase, each sample was processed through 10,000 iterations to calculate the expected probability for each data point. The appendix contains an in-depth analysis of the `CIFAR-10` results, covering all 10,000 samples. For the `ImageNet` dataset, a broader subset of 1,000 samples was examined. An in-depth ablation study, delving into the impact of Monte Carlo sampling frequency, is detailed in § C.5.

For our experiments on the `CoraML` 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 and 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, earmarking 10% of the dataset for calibration. The quantile was iteratively computed 100 times, with each round using a random calibration subset and the average performance was recorded over all these iterations.

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 radius 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. C.2 shows that CAS outperform RSCP on all datasets.

Proportion of Empty, Singleton, and Multi-sets. In the manuscript, we reported the effective set size as the average size of non-empty prediction sets. It is also important to report the proportion of empty, singleton, and multi-prediction sets. 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. Hence we expect more multi-sets in more conservative approaches like RSCP compared to CAS since it is based on a higher upper-bound. Fig. C.3 verifies this.

Effect of σ. We comprehensively examined the impact of various σ-s for randomized smoothing. Approving the theoretical results, CAS consistently outperforms RSCP for various σ-levels. Fig. C.4 shows smaller prediction sets and higher singleton hits ratio for CAS across various smoothing levels and radii within the range $r, \sigma \in \{0.12, 0.25, 0.50\}$. For this experiment, we set the coverage guarantee to 90%.

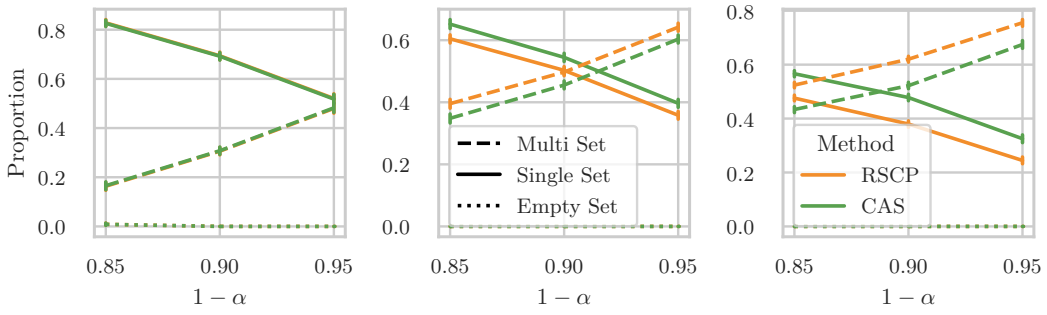

Figure C.3: The proportion of singleton, empty and multi-sets for RSCP and CAS across radii (left) $r = 0$, (middle) $r = 0.12$, and (right) $r = 0.25$.

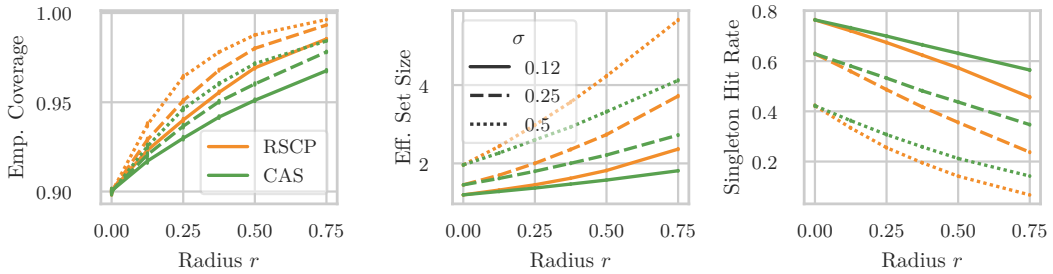

Figure C.4: Comparison of RSCP and CAS across various smoothing levels (0.12, 0.25, 0.50) and radii on the `CIFAR-10` dataset, highlighting the consistent superiority of CAS.

Different Score Functions. As mentioned in § 5.2 (and in § C.1 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. C.5 compares RSCP with CAS applied on TPS and APS score functions. In all scores, and all metrics CAS shows an improved result.

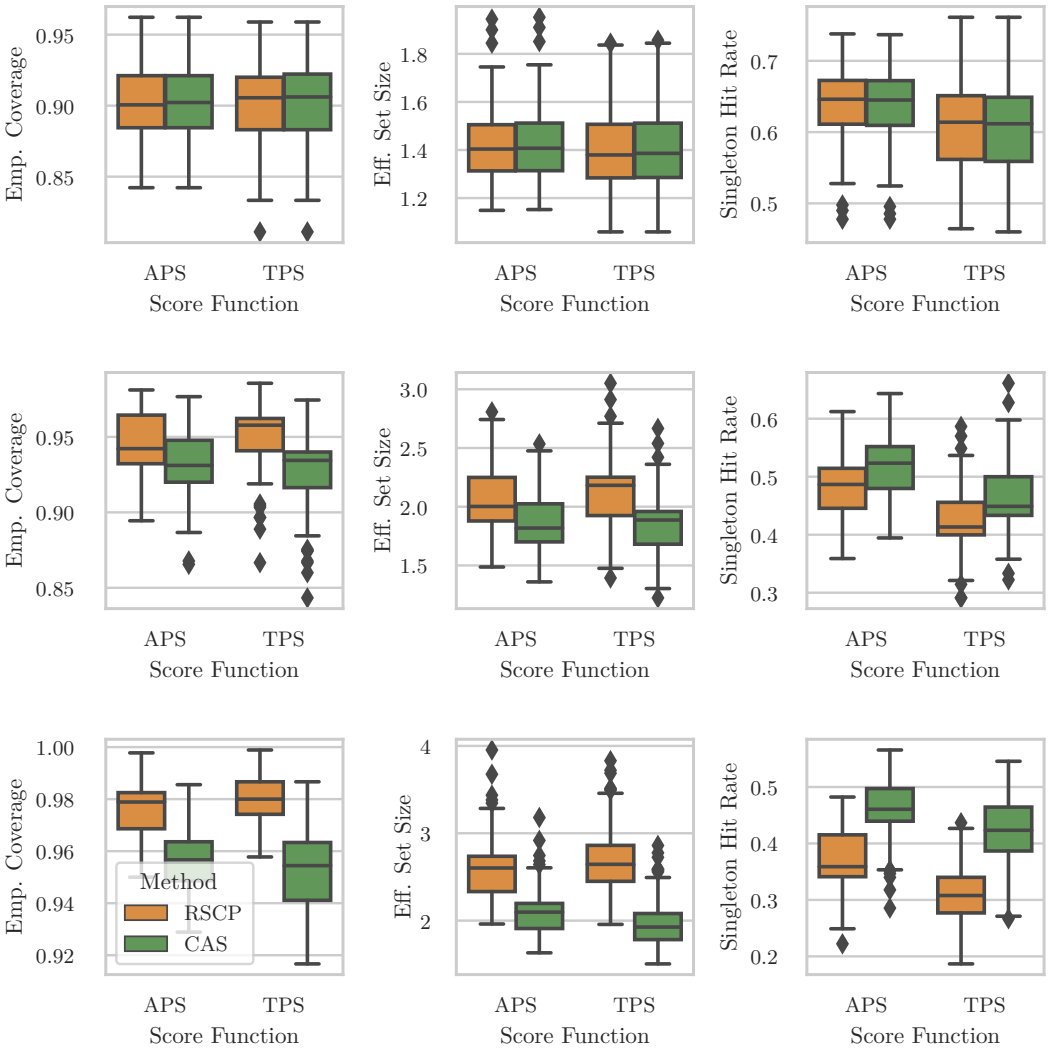

Figure C.5: Comparison of RSCP and CAS for smooth APS and TPS score across various radii for (left column) empirical coverage (middle column) set size, and (right column) singleton hits. From upper to lower row results are respectively for $r = 0$, $r = 0.12$, and $r = 0.25$. All results are for `CIFAR-10`, and smoothing with $\sigma = 0.25$

D Supplementary Material: Robust Conformal Prediction with a Single Binary Certificate

D.1 Algorithm for Robust (and Vanilla) BinCP

Here we provide the algorithm for BinCP in both p-fixed, and τ-fixed setups. The two setups differ in calibration and finite sample correction, while computing the certificate, and returning prediction sets is similar in both. Note that in p-fixed version after computing the quantile τ_α we correct for finite samples which results in a lower $p_\alpha^\downarrow$.

Note that in both algorithms we set m, and η as fixed hyper-parameters defining the number of random samples per each datapoint and the collective failure probability of the confidence intervals. Therefore ClopperPearson(p) actually refers to the Clopper-Pearson interval with $p \cdot m$ success out of m samples and failure probability of $\eta/(k + |\mathcal{D}_{\text{cal}}|)$.

Algorithm 5 BinCP with τ-fixed setup

Input: Score function $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$; Calibration set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$; Smoothing ξ; Threat model $\mathcal{B}$; Fixed threshold τ; Test point $\tilde{\boldsymbol{x}}_{n+1}$
Output: Prediction set $\bar{C}_{\text{bin}}(\tilde{\boldsymbol{x}}_{n+1})$ with $1 - \alpha$ robust coverage

▷ Calibration phase
for each $(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}$ **do**
  Sample from $\boldsymbol{x}_i + \boldsymbol{\epsilon}$ for m times
  $q_i \leftarrow \frac{1}{m} \sum_{j=1}^m \mathbb{I}[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i) \geq \tau]$
  if Exact Certificate **then**
    $q_i^\downarrow \leftarrow q_i$
  else
    $q_i^\downarrow \leftarrow \text{ClopperPearson}_{\text{low}}(q_i)$
  end if
end for
$p_\alpha^\downarrow \leftarrow \mathbb{Q}\left(\alpha; \{q_i^\downarrow\}_{i=1}^n\right)$
Compute $\text{c}^\downarrow[p_\alpha^\downarrow, \mathcal{B}]$ from Eq. 60

▷ Test phase
for each $y \in \mathcal{Y}$ **do**
  Sample from $\xi(\tilde{\boldsymbol{x}}_{n+1})$ for m times
  $q_{n+1,y} \leftarrow \frac{1}{m} \sum_{j=1}^m \mathbb{I}[s(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, y) \geq \tau]$
  if Exact Certificate **then**
    $q_{n+1,y}^\uparrow \leftarrow q_{n+1,y}$
  else
    $q_{n+1,y}^\uparrow \leftarrow \text{ClopperPearson}_{\text{high}}(q_{n+1,y})$
  end if
end for

Return $\{y : q_{n+1,y}^\uparrow \geq \text{c}^\downarrow[p_\alpha^\downarrow, \mathcal{B}]\}$

Score functions. Throughout the paper we reported results for TPS score – directly setting the softmax values as the score $s(\boldsymbol{x}, y) = \pi(\boldsymbol{x}, y)$ (Sadinle et al., 2018). In vanilla CP

Algorithm 6 BinCP with p-fixed setup

Input: Same as Algorithm 5. Fixed probability p_α
Output: Same as Algorithm 5

$p_\alpha \leftarrow \lceil p_\alpha \cdot m \rceil / m$ ▷ Account for discrete samples

▷ Calibration phase
for each $(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}$ **do**
  Sample from $\boldsymbol{x}_i + \boldsymbol{\epsilon}$ for m times
  $\tau_i \leftarrow \mathbb{Q}\left(p_\alpha; \{s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i)\}_{j=1}^{m}\right)$
  if Exact Certificate **then**
    $p_\alpha^{\downarrow} \leftarrow p_\alpha$
  else
    $p_\alpha^{\downarrow} \leftarrow \text{ClopperPearson}_{\text{low}}(p_\alpha)$
  end if
end for
$\tau_\alpha \leftarrow \mathbb{Q}\left(\alpha; \{\tau_i\}_{i=1}^{n}\right)$
Compute $\text{c}^{\downarrow}[p_\alpha^{\downarrow}, \mathcal{B}]$ from Eq. 60

▷ Test phase
for each $y \in \mathcal{Y}$ **do**
  Sample from $\xi(\tilde{\boldsymbol{x}}_{n+1})$ for m times
  $q_{n+1,y} \leftarrow \frac{1}{m} \sum_{j=1}^{m} \mathbb{I}[s(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, y) \geq \tau_\alpha]$
  if Exact Certificate **then**
    $q_{n+1,y}^{\uparrow} \leftarrow q_{n+1,y}$
  else
    $q_{n+1,y}^{\uparrow} \leftarrow \text{ClopperPearson}_{\text{high}}(q_{n+1,y})$
  end if
end for

Return $\{y : q_{n+1,y}^{\uparrow} \geq \text{c}^{\downarrow}[p_\alpha^{\downarrow}, \mathcal{B}]\}$

($r = 0$), TPS tends to over-cover easy examples and under-cover hard ones (Angelopoulos and Bates, 2021). Alternatively "adaptive prediction sets" (APS) aiming for conditional coverage uses the score function defined as $s(\boldsymbol{x}, y) := -\left(\rho(\boldsymbol{x}, y) + u \cdot \pi(\boldsymbol{x})_y\right)$ where $\rho(\boldsymbol{x}, y) := \sum_{c=1}^{K} \pi(\boldsymbol{x})_c 1\left[\pi(\boldsymbol{x})_c > \pi(\boldsymbol{x})_y\right]$ 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). Another approach is to directly use the logits of the model as the score. This is not applicable in CAS and RCSP+ (Zargarbashi et al., 2024; Yan et al., 2024) as they work with bounded scores. BinCP can also work with unbounded score, hence, we also report the results on CP with logits. All three score functions are reported in Fig. 6.6-left for BinCP. Interestingly we do not see any significant difference in set size between APS and TPS when smoothed. We report BinCP with APS score in § D.5 over all σ, and radii. Orthogonally, while RSCP with either scores quickly breaks to returning trivial sets, RSCP+ refines the score function through a biased temperature scaling. This can also be considered as another score function (or transformation over any score function) tailored for robust setup. We compare BinCP with RSCP+ in Fig. D.7 (§ D.5).

D.2 Computing Certificate Optimization

Canonical view. Turns out that for isotropic Gaussian and sparse smoothing, we can always attain this minimum at canonical points – for any test point $\boldsymbol{x}_{n+1}$ we can translate the function, and the points to the origin, and the worst case to the border of $\mathcal{B}(\boldsymbol{x}_{n+1})$. Formally, there is a pair $(\boldsymbol{u}, \tilde{\boldsymbol{u}})$ such that $\rho_{\boldsymbol{u},\tilde{\boldsymbol{u}}} = \rho_{\boldsymbol{x}_{n+1},\tilde{\boldsymbol{x}}_{n+1}}$ for any $\boldsymbol{x}_{n+1}$, and $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$. Namely for the continuous ball $\mathcal{B}_r$ the canonical vectors are $\boldsymbol{u} = \mathbf{0}$ and $\tilde{\boldsymbol{u}} = [r, 0, 0, \dots]$. For the binary $\mathcal{B}_{r_a,r_d}$ we have the canonical $\boldsymbol{u} = [0, \dots, 0, 1, \dots, 1]$ and $\tilde{\boldsymbol{u}} = \mathbf{1} - \boldsymbol{u}$ where $\|\boldsymbol{u}\|_0 = r_d$ and $\|\tilde{\boldsymbol{u}}\|_0 = r_a$. Intuitively it is due to the symmetry of the ball and the smoothing distribution. To avoid many notations, we again use the $\boldsymbol{x}$, and $\tilde{\boldsymbol{x}}$ in the rest of the discussion that refers to the canonical points.

To obtain an upper or lower bound (Eq. 60 as maximization or minimization) we partition the space $\mathcal{X}$ to regions where the likelihood ratio between $(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ is constant; formally $\mathcal{X} = \cup_{i=1}^{k} \mathcal{R}_i$ where $\forall \boldsymbol{z} \in \mathcal{R}_i : \Pr[\xi(\boldsymbol{x}) = \boldsymbol{z}]/\Pr[\xi(\tilde{\boldsymbol{x}}) = \boldsymbol{z}] = c_i$. For any function h we can find an equivalent piecewise-constant $\hat{h}$ where inside each region it is assigned to the expected value of h in that region. Let $t_i = \Pr[\xi(\boldsymbol{x}) = \boldsymbol{z}]$, and $\tilde{t}_i = \Pr[\xi(\tilde{\boldsymbol{x}}) = \boldsymbol{z}]$ then Eq. 60 simplifies to the following linear programming

$$\min_{\boldsymbol{h} \in [0,1]^k} \boldsymbol{h}^\top \tilde{\boldsymbol{t}} \quad \text{s.t.} \quad \boldsymbol{h}^\top \boldsymbol{t} = p_\alpha \tag{98}$$

Where $\boldsymbol{h}$, $\boldsymbol{t}$, and $\tilde{\boldsymbol{t}}$ are vectors that include the values h_i, t_i, $\tilde{t}_i$ for each region. The optimum solution to the simplified linear programming is obtained by sorting regions based on the likelihood ratio and greedily assigning h to the possible maximum in each region until the budget $\boldsymbol{h}^\top \boldsymbol{t} = p_\alpha$ is met. The rest of the regions are similarly assigned to zero. This problem is equivalent to fractional knapsack. For isotropic Gaussian smoothing Cohen et al. (2019) show that the optimal solution has a closed form $\rho_\alpha = \Phi_\sigma(\Phi_\sigma^{-1}(p_\alpha) - r)$ where Φ_σ is the Gaussian CDF function of the Gaussian distribution with standard deviation σ. For sparse smoothing, following Bojchevski et al. (2020) we solve the greedy program on at most $r_a + r_d + 1$ distinct regions. The runtime is linear w.r.t. to the add and delete budget. For more detailed explanation see (Lee et al., 2019; Bojchevski et al., 2020).

D.3 Supplementary to Theory

D.3.1 Vanilla BinCP

Conformal risk control. We use conformal risk control (CRC) (Angelopoulos et al., 2022) to prove the coverage guarantee in BinCP. Here we succinctly recall it before the proof of Proposition 6.2.

Theorem D.1 (Conformal Risk Control - rephrased)**.** *Let λ be a parameter (larger λ yields more conservative output), and $L_i : \Lambda \to (-\infty, B]$ for $i = 1, \ldots, n+1$ be exchangeable random functions. If (i) L_is are non-increasing right-continuous w.r.t. λ, (ii) for $\lambda_{\max} = \sup \Lambda$ we have $L_i(\lambda_{\max}) \leq \alpha$, and (iii)* $\sup_\lambda L_i \leq B < \infty$*, then we have:*

$$\mathbb{E}[L_{n+1}(\hat{\lambda})] \leq \alpha \quad \textit{for} \quad \hat{\lambda} = \inf\left\{\lambda : \frac{\sum_{i=1}^{n} L_i(\lambda)}{n+1} + \frac{B}{n+1} \leq \alpha\right\} \tag{99}$$

In case that $B = 1$, by simplifying Eq. 99, we have $\hat{\lambda} = \inf\left\{\lambda : \sum_{i=1}^{n} L_i(\lambda) \leq \alpha(n+1) - 1\right\}$ We use this framework to prove the guarantee in BinCP.

Proof to Proposition 6.2. We prove the theorem through re-parameterizing of the conservativeness variable in each case. For fixed p we set $\tau = 1 - \lambda$; similarly for fixed τ we set $p = -\lambda$. In both cases, the risk is defined as

$$L_i(\tau, p) = 1 - \text{accept}(\boldsymbol{x}_i, y_i; p, \tau)$$

which for simplicity we define $\text{reject}(\boldsymbol{x}_i, y_i; p, \tau) = 1 - \text{accept}(\boldsymbol{x}_i, y_i; p, \tau)$ and by definition we have $\text{reject}(\boldsymbol{x}_i, y_i; p, \tau) = \mathbb{I}[\Pr[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y) < \tau] > 1 - p] = \mathbb{I}[\Pr[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y) \geq \tau] < p]$. We show that the risk function satisfies the properties for a risk function feasible to the setup in Theorem D.1.

1. **Non-increasing to** λ. In both cases the risk L_i is non-inscreasing to λ; for fixed p we have

$$\begin{aligned} \lambda_1 < \lambda_2 &\Rightarrow 1 - \lambda_1 > 1 - \lambda_2 \\ &\Rightarrow \Pr[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y) < 1 - \lambda_1] \geq \Pr[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y) < 1 - \lambda_2] \\ &\Rightarrow \text{reject}(\boldsymbol{x}, y; p, 1 - \lambda_1) \geq \text{reject}(\boldsymbol{x}, y; p, 1 - \lambda_2) \end{aligned}$$

   Now for fixed τ, let $p_{\boldsymbol{x}} = \Pr[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y) \geq \tau]$ then we have

$$\begin{aligned} \lambda_1 < \lambda_2 &\Rightarrow p_1 > p_2 \text{ means that } \mathbb{I}[p_{\boldsymbol{x}} \leq p_1] \geq \mathbb{I}[p_{\boldsymbol{x}} \leq p_2] \\ &\Rightarrow \text{reject}(\boldsymbol{x}, y; -\lambda_1, \tau) \geq \text{reject}(\boldsymbol{x}, y; -\lambda_2, \tau) \end{aligned}$$

   Intuitively by adapting the definition of the rejection (risk) function $\text{reject}(\boldsymbol{x}_i, y_i; p, \tau) = \mathbb{I}[\Pr[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y) \geq \tau] < p]$, if we increase λ which means decreasing p, the chance of rejecting a label decreases. This is because, we require the same probability mass to be lower than a smaller value.

2. **Right continuous**. Formally the function accept is

$$\text{accept}(\boldsymbol{x}, y; p, \tau) = \begin{cases} 1 & \text{if } \Pr[s(\boldsymbol{x} + \boldsymbol{\epsilon}, y) \geq \tau] \geq p \\ 0 & \text{otherwise} \end{cases}$$

   Across the domain (for either p or τ) this function has two values and it is just non-continuous in the jump between the values. For both p and τ this function is left continuous due to the $\geq$ comparison. Therefore for fixed p the function $\text{reject}(\boldsymbol{x}, y; p, 1 - \lambda)$ is right continuous to $\lambda = 1 - \tau$. Similar argument follows for fixed τ.

3. **Feasibility of risks less than** α. For fixed $p > 0$ if we set $\lambda = 1 - \tau$ to ∞ ($\tau = -\infty$), for all $\boldsymbol{x}_i$, we have $\text{accept}(\boldsymbol{x}_i, y_i; p, 0) = 1$; i.e. the risk is 0 for every data. Similarly by approaching p to zero in fixed τ setup, we decrease the risk to 0 for everyone. To avoid corner cases we can restrict τ to $\max s(\boldsymbol{x} + \boldsymbol{\epsilon}, y)$ for $x \in \mathcal{X}$ from above.

4. **Limited upperbound risk**. For any parameter and any input the highest possible risk is in case of rejection which is 1 ($B = 1$).

Fixed p. The risk function $L_i(\lambda) = \text{reject}(\boldsymbol{x}_i, y_i; p, 1 - \lambda)$ which means that the prediction set $C(\boldsymbol{x}_i; p, 1 - \lambda)$ excludes y_i. We have

$$\mathbb{E}[\text{reject}(\boldsymbol{x}_{n+1}, y_{n+1}; p, 1 - \hat{\lambda})] \leq \alpha$$

$$\text{for } \hat{\lambda} = \inf_{\lambda} \left\{ \lambda : \sum_{i=1}^{n} \text{reject}(\boldsymbol{x}_i, y_i; p, 1 - \lambda) \leq \alpha(n+1) - 1 \right\}$$

Setting back the $\tau = 1 - \lambda$, and rewriting the expectation as a probability form, we have

$$\Pr\left[y_{n+1} \in C(\boldsymbol{x}_{n+1}; p, \tau_p)\right] \geq 1 - \alpha$$

$$\text{for } \tau_p = \sup_{\tau} \left\{ \tau : \sum_{i=1}^{n} \text{accept}(\boldsymbol{x}_i, y_i; p, \tau) \geq (1 - \alpha)(n+1) \right\}$$

In the above, we used the fact that if a test fails on $\alpha(n+1) - 1$ variables among the total of n variables, it passes on $n - [\alpha(n+1) - 1]$ and $(1 - \alpha)(n+1) = n - [\alpha(n+1) - 1]$.

Fixed τ. Similarly, we define the risk function as $L_i(\lambda) = \text{reject}(\boldsymbol{x}_i, y_i; -\lambda, \tau)$. We have

$$\mathbb{E}[\text{reject}(\boldsymbol{x}_{n+1}, y_{n+1}; p_\tau, \tau)] \leq \alpha$$

$$\text{for} \quad p_\tau = \inf_{\lambda} \left\{ \lambda : \sum_{i=1}^{n} \text{reject}(\boldsymbol{x}_i, y_i; -\lambda, \tau) \leq \alpha(n+1) - 1 \right\}$$

□

Proof to Lemma 6.3. The function $\text{accept}(\boldsymbol{x}, y; p, \tau)$ is non-increasing in both p and τ. Therefore the term $\sum_{i=1}^{n} \text{accept}(\boldsymbol{x}_i, y_i; p, \tau)$ is also non-increasing in p and τ and its range is the integer numbers between 0 and n (or $[n]$). For a fixed p, let $\tau_\alpha(p)$ be the solution to Eq. 57, then by definition it satisfies that

$$\sum_{i=1}^{n} \text{accept}(\boldsymbol{x}_i, y_i; p, \tau_\alpha(p)) \geq (1 - \alpha)(n+1)$$

This implies that p satisfies the same condition for $p_\alpha(\tau_\alpha(p))$. Therefore $p_\alpha(\tau_\alpha(p)) \geq p$ as p is a feasible solution in Eq. 58. The supremum search for $\tau_\alpha(p)$ directly implies that for any positive δ we have

$$\sum_{i=1}^{n} \text{accept}(\boldsymbol{x}_i, y_i; p, \tau_\alpha(p)) \geq \sum_{i=1}^{n} \text{accept}(\boldsymbol{x}_i, y_i; p, \tau_\alpha(p) + \delta) - 1$$

which intuitively means that increasing the $\tau(p)$ by any small margin fails at least in one more accept for calibration points. Since $\sum_{i=1}^{n} \text{accept}(\boldsymbol{x}_i, y_i; p, \tau_\alpha(p))$ is the sum of n non-increasing functions, there is one index i for which

$$\text{accept}(\boldsymbol{x}_i, y_i; p, \tau_\alpha(p)) = 1 \quad \text{and} \quad \text{accept}(\boldsymbol{x}_i, y_i; p, \tau_\alpha(p) + \delta) = 0$$

For any small positive δ. Using the definition of the accept function we have

$$\Pr[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i) \geq \tau_\alpha(p)] \geq p \quad \text{and} \quad \Pr[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i) \geq \tau_\alpha(p) + \delta] < p$$

Due to the continuous strictly increasing CDF for S_i we have $\Pr[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i) \geq \tau(p)] = p$. Therefore for any small positive δ

$$\text{accept}(\boldsymbol{x}_i, y_i; p, \tau_\alpha(p)) = 1 \quad \text{and} \quad \text{accept}(\boldsymbol{x}_i, y_i; p + \delta, \tau_\alpha(p)) = 0$$

which means that the accept function for $\boldsymbol{x}_i$ fails by adding a small number to p. Since all other accept functions are also non-increasing we have $\sum_{i=1}^{n} \text{accept}(\boldsymbol{x}_i, y_i; p, \tau_\alpha(p)) \leq (1-\alpha)(n+1) - 1$. This implies that p is also the supremum for $Eq.$ 57 with parameter $p_\alpha(\tau)$. □

D.3.2 Robust BinCP

Proof to Lemma 6.4. With $f_{\text{true}}(\boldsymbol{x}_i) = \mathbb{I}[s(\boldsymbol{x}_i, y_i) \geq \tau_\alpha]$ for true y_i, the calibration-time robust prediction set is defined as $\bar{C}_{\text{cal}}(\tilde{\boldsymbol{x}}_{n+1}) = \{p(\tilde{\boldsymbol{x}}_{n+1}, y; \tau_\alpha) \geq \mathbb{Q}\left(\alpha; \{\text{c}^{\downarrow}[\bar{f}_{\text{true}}(\boldsymbol{x}_i), \mathcal{B}]\}_{i=1}^{n}\right)\}$. By definition we have $p_\alpha = \mathbb{Q}\left(\alpha; \{\bar{f}_{\text{true}}(\boldsymbol{x})\}_{i=1}^{n}\right)\}$. Both lower bound and upper bound functions are non-decreasing. As a result, the ranks, and hence the quantile index in $\{\bar{f}_{\text{true}}(\boldsymbol{x}_i)\}_{i=1}^{n}$ and $\{\text{c}^{\downarrow}[\bar{f}_{\text{true}}(\boldsymbol{x}_i), \mathcal{B}]\}_{i=1}^{n}$ are the same. Therefore, $\mathbb{Q}\left(\alpha; \{\text{c}^{\downarrow}[\bar{f}_{\text{true}}(\boldsymbol{x}_i), \mathcal{B}]\}_{i=1}^{n}\right)\}$ $= \text{c}^{\downarrow}[p_\alpha, \mathcal{B}]$.

The test-time robust prediction set is defined as $\bar{C}_{\text{test}} = \{y : \text{c}^{\uparrow}[\bar{f}_y(\tilde{\boldsymbol{x}}_{n+1}), \mathcal{B}^{-1}] \geq p_\alpha\}$, let $\tilde{p}_y = \bar{f}_y(\tilde{\boldsymbol{x}}_{n+1})$ then it follows

$$\begin{aligned} \text{c}^{\uparrow}[\bar{f}_y(\tilde{\boldsymbol{x}}_{n+1}), \mathcal{B}^{-1}] \geq p_\alpha \Leftrightarrow \text{c}^{\downarrow}[\text{c}^{\uparrow}[\bar{f}_y(\tilde{\boldsymbol{x}}_{n+1}), \mathcal{B}^{-1}], \mathcal{B}] \geq \text{c}^{\downarrow}[p_\alpha, \mathcal{B}] \\ \Leftrightarrow \bar{f}_y(\tilde{\boldsymbol{x}}_{n+1}) \geq \text{c}^{\downarrow}[p_\alpha, \mathcal{B}] \end{aligned}$$

By definition $\text{accept}(\tilde{\boldsymbol{x}}_{n+1}, y; \text{c}^{\downarrow}[p_\alpha, \mathcal{B}], \tau_\alpha) = \mathbb{I}[\bar{f}_y(\tilde{\boldsymbol{x}}_{n+1}) \geq \text{c}^{\downarrow}[p_\alpha, \mathcal{B}]]$. □

In the above, we proved that BinCP results in a valid conformal prediction. Here we prove the validity of robust BinCP to adversarial data within the bounded threat model.

Proof to Lemma 6.5. For each of the mentioned smoothing schemes we have:

Gaussian smoothing. In both cases since p norm is symmetric for any point $\tilde{\boldsymbol{x}}$ it holds that $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_p \leq r$. In other words, from any perturbed point the clean point is within $\mathcal{B}_r(\tilde{\boldsymbol{x}})$. Therefore $\mathcal{B}_r^{-1} = \mathcal{B}_r$.

For simpler notation let $\overline{p} = \text{c}^{\uparrow}[p, \mathcal{B}_r]$. Given the closed from solution $\text{c}^{\downarrow}[p, \mathcal{B}_r] = \Phi_\sigma(\Phi_\sigma^{-1}(p) - r)$, and $\text{c}^{\uparrow}[p, \mathcal{B}_r] = \Phi_\sigma(\Phi_\sigma^{-1}(p) + r)$ we have

$$\begin{aligned} \overline{p} = \Phi_\sigma(\Phi_\sigma^{-1}(p) + r) \Leftrightarrow \Phi_\sigma^{-1}(\overline{p}) = \Phi_\sigma^{-1}(p) + r \Leftrightarrow \Phi_\sigma^{-1}(\overline{p}) - r = \Phi_\sigma^{-1}(p) \\ \Leftrightarrow \Phi_\sigma(\Phi_\sigma^{-1}(\overline{p}) - r) = p \Leftrightarrow \text{c}^{\downarrow}[\overline{p}, \mathcal{B}] = p \end{aligned}$$

Uniform smoothing. For the uniform smoothing from Levine and Feizi (2021) we have that the smooth classifier is $1/(2\lambda)$-Lipschitz continuous. From that the certified lower and upper bounds are defined as $\text{c}^{\downarrow}[p, \mathcal{B}_r] = p - r \cdot 1/(2\lambda)$, and $\text{c}^{\uparrow}[p, \mathcal{B}_r] = p + r \cdot 1/(2\lambda)$ Therefore

$$\text{c}^{\downarrow}[\text{c}^{\uparrow}[p, \mathcal{B}_r^{-1}], \mathcal{B}_r] = \text{c}^{\downarrow}[p + \frac{r}{2\lambda}, \mathcal{B}_r] = p + \frac{r}{2\lambda} - \frac{r}{2\lambda} = p$$

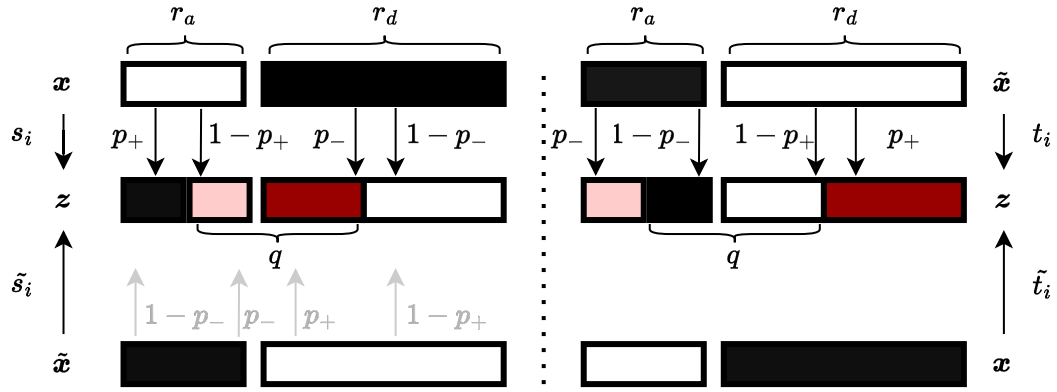

Figure D.1: Illustration of likelihood ratio in sparse smoothing for both $\mathcal{B}_{r_a,r_d}$, and $\mathcal{B}_{r_d,r_a}$

A similar argument can be applied to any certificate that directly adds (or subtracts) the Lipschitz constant of the smooth classifier to the base probability.

Sparse smoothing. Any $\tilde{\boldsymbol{x}} \in \mathcal{B}_{r_a,r_d}(\boldsymbol{x})$ has at most r_a zero bits, and r_d one bits toggled from $\boldsymbol{x}$. By toggling those bit back we can reconstruct $\boldsymbol{x}$. The maximum needed toggles is therefore r_d zero bits and r_a one bits which is the definition of $\mathcal{B}_{r_d,r_a}$.

As discussed in § D.2, canonical points for $\mathcal{B}_{r_a,r_d}$ are $\boldsymbol{x} = [0,\dots,0,1,\dots,1]$ and $\tilde{\boldsymbol{x}} = \mathbf{1} - \boldsymbol{x}$ where $\|\boldsymbol{x}\|_0 = r_d$ and $\|\tilde{\boldsymbol{x}}\|_0 = r_a$. For $\mathcal{B}^{-1}_{r_a,r_d}$ the canonical points are $\boldsymbol{u}$, $\tilde{\boldsymbol{u}}$ where $\|\boldsymbol{u}\|_0 = r_a$. By applying a permutation over $\boldsymbol{u}$, $\tilde{\boldsymbol{x}}$ and every other point in all regions we can set $\boldsymbol{u} = \tilde{\boldsymbol{x}}$, and $\tilde{\boldsymbol{u}} = \boldsymbol{x}$. For computing both $\mathcal{B}_{r_a,r_d}$, and $\mathcal{B}^{-1}_{r_a,r_d}$ there are $r_a + r_d + 1$ regions of constant likelihood ratio, each including all points that have the same number of total flips from the source $\boldsymbol{x}$, or $\boldsymbol{u}$; formally $\mathcal{R}_q = \{\boldsymbol{z} : \|\boldsymbol{x} - \boldsymbol{z}\|_0 = q\}$. The same region can also defined to preserve $r_d + r_a - q$ bits from $\tilde{\boldsymbol{x}}$. With $\frac{s_q}{\tilde{s}_q}$ as the likelihood ratio of a point $\boldsymbol{z}$ in $\mathcal{B}_{r_a,r_d}$ and $q = q_a + q_d$ as the number of changes in 1 and 0 bit, we have $s_q = (p_+)^{q_a}(1-p_+)^{r_a-q_a}(p_-)^{q_d}(1-p_-)^{r_d-q_d}$, and similarly $\tilde{s}_q = (p_-)^{q_a}(1-p_-)^{r_a-q_a}(p_-)^{q_d}(1-p_-)^{r_d-q_d}$. Then the likelihood ratio is simplified to

$$\frac{s_q}{\tilde{s}_q} = \left[\frac{p_+}{1-p_-}\right]^{q-r_d}\left[\frac{p_-}{1-p_+}\right]^{q-r_a} \tag{100}$$

As illustrated in Fig. D.1 regions for $\mathcal{B}^{-1}_{r_a,r_d}$ are same as $\mathcal{B}_{r_a,r_d}$ only with reverse order. In other word, let t_i, $\tilde{t}_i$ be the probability of visiting region $\mathcal{R}_q$ from $\boldsymbol{u}$ and $\tilde{\boldsymbol{u}}$, then $t_i = \tilde{s}_{r_a+r_d+1-q}$, and $\tilde{t}_i = s_{r_a+r_d+1-q}$. For a fixed $\boldsymbol{z}$ the probability to visit $\boldsymbol{z}$ from $\boldsymbol{x}$ is the probability of toggling $q = \|\boldsymbol{z} - \boldsymbol{x}\|_0$ bits which is the same as toggling q bits from $\tilde{\boldsymbol{u}}$ as $\tilde{\boldsymbol{u}} = \boldsymbol{x}$.

Solutions to $\mathrm{c}^{\downarrow}[p, \mathcal{B}_{r_a,r_d}]$ and $\mathrm{c}^{\uparrow}[p, \mathcal{B}^{-1}_{r_a,r_d}]$ are obtained from the following optimization functions:

$$\mathrm{c}^{\downarrow}[p, \mathcal{B}_{r_a,r_d}] = \min_{\boldsymbol{h}\in[0,1]^{r_a+r_d+1}} \boldsymbol{h}^\top \tilde{\boldsymbol{t}} \quad \text{s.t.} \quad \boldsymbol{h}^\top \boldsymbol{t} = p$$
$$\mathrm{c}^{\uparrow}[p, \mathcal{B}^{-1}_{r_a,r_d}] = \max_{\boldsymbol{h}\in[0,1]^{r_a+r_d+1}} \boldsymbol{h}^\top \tilde{\boldsymbol{s}} \quad \text{s.t.} \quad \boldsymbol{h}^\top \boldsymbol{s} = p$$

The solution to the lower bound optimization is obtained by a greedy algorithm. We visit each in increasing order w.r.t. $\frac{s_q}{\tilde{s}_q}$, we assign $h_q = 1$ until the budget $\boldsymbol{h}^\top \boldsymbol{s}$ is met and we set $h_q = 0$ for the remaining regions (fractional knapsack problem). For the maximization we do the same but in a decreasing order.

We want to prove $\mathrm{c}^{\downarrow}[\mathrm{c}^{\uparrow}[p, \mathcal{B}^{-1}_{r_a,r_d}], \mathcal{B}_{r_a,r_d}] = p$. This is the solution to

$$\min_{\boldsymbol{h}\in[0,1]^{r_a+r_d+1}} \boldsymbol{h}^\top \tilde{\boldsymbol{t}} \quad \text{s.t.} \quad \boldsymbol{h}^\top \boldsymbol{t} = \mathrm{c}^{\uparrow}[p, \mathcal{B}^{-1}_{r_a,r_d}] = \boldsymbol{h}'^\top \tilde{\boldsymbol{s}}$$

Let $\overleftarrow{\tilde{\boldsymbol{s}}}$ be the vector $\tilde{\boldsymbol{s}}$ in reverse order. Then $\boldsymbol{t} = \overleftarrow{\tilde{\boldsymbol{s}}}$. From the problem definition we have that $\overleftarrow{\boldsymbol{h}'}$ (the solution from maximization problem in reverse order) is a feasible solution. Given that the solution of the optimization (reduced to fractional knapsack problem) is always in form of $[1, \dots, 1, \delta, 0, \dots, 0]$ for some $\delta \in [0, 1]$, any vector of this form that satisfies the constraint is optimal. Therefore $\overleftarrow{\boldsymbol{h}'}$ is the solution to the maximization greedy problem. So the optimal solution is $\overleftarrow{\boldsymbol{h}'}^\top \tilde{\boldsymbol{t}} = \overleftarrow{\boldsymbol{h}'}^\top \overleftarrow{\boldsymbol{s}} = p$. □

For any ℓ_p with the same argument as ℓ_2 ball we have $\mathcal{B}^{-1}_r = \mathcal{B}_r$. Similar to isotropic Gaussian smoothing, the Lipschitz continuity in DSSN-smoothed distribution shows that Lemma 6.5 applies to ℓ_1 ball and this distribution as well.

D.3.3 Correction for Finite Sample Monte-Carlo Estimation

Proof to Proposition 6.7. With $p_i = \Pr[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i) \geq \tau_\alpha]$ as the true probability of crossing τ_α for each true score distribution in calibration set. We have $p_\alpha = \mathbb{Q}\left(\alpha; \{p_i\}_{i=1}^n\right)$. For all i we have $q_i^{\downarrow} \leq p_i$ from which follows $p_\alpha^{\downarrow} \leq p_\alpha$. The probability of failure in each calibration datapoint is $\eta/(|\mathcal{D}_{\text{cal}}| + k)$; as a result, from the union bound the probability of failure $q_i^{\downarrow} \leq p_i$ for all $i \in \{1, ..., n\}$ and therefore the quantile is $|\mathcal{D}_{\text{cal}}|\eta/(|\mathcal{D}_{\text{cal}}| + k)$.

For all classes of the test point we have $\tilde{q}^{\uparrow}_{n+1,y} \geq \tilde{p}_{n+1,y}$ with $\eta/(|\mathcal{D}_{\text{cal}}| + k)$. Therefore, for the true class we have $\tilde{q}_{n+1} \geq \tilde{p}_{n+1}$ with $k\eta/(|\mathcal{D}_{\text{cal}}| + k)$.

Conformal guarantee implies that with $1 - \alpha + \eta$ probability we have $p_{n+1} \geq p_\alpha$. The robustness certificate implies that $\tilde{p}_{n+1} \geq \mathrm{c}^{\downarrow}[p_\alpha, \mathcal{B}]$. Following holds by using the mentioned inequality:

$$\tilde{q}^{\uparrow}_{n+1} \underset{1-\frac{k\eta}{n+k}}{\geq} \tilde{p}_{n+1} \underset{1-\alpha+\eta}{\geq} \mathrm{c}^{\downarrow}[p_\alpha, \mathcal{B}] \underset{1-\frac{n\eta}{n+k}}{\geq} \mathrm{c}^{\downarrow}[p_\alpha^{\downarrow}, \mathcal{B}]$$

From the union bound it follows that the total failure probability is less than α. □

We estimate the probabilities in Algorithm 5 following Proposition 6.7 directly. There, we estimate the $p_i = \Pr_{\boldsymbol{\epsilon}}[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i) \geq \tau]$ via Monte Carlo samples resulting in $q_i = \frac{1}{m}\mathbb{I}[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i) \geq \tau]$. Via Clopper Pearson bounds we find $q_i^{\downarrow}$ for which we have $q_i^{\downarrow} \leq p_i$ with adjusted $1 - \eta$ probability.

For the p-fixed approach (Algorithm 6), we compute the discrete quantile $\tau_i = \mathbb{Q}(\lceil p_\alpha \cdot m \rceil / m; \{s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i)\}_{i=1}^m$ over the randomly sampled scores. By definition of discrete quantile function, without counting again, we have $\frac{1}{m}\mathbb{I}[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i) \geq \tau_i] \geq p_\alpha$ (p_α proportion of the binary variables $\mathbb{I}[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i) \geq \tau_i]$ are 1). Now, we bound a Bernoulli parameter that is the proportion of the distribution exceeding τ_i. Again we use Clopper Pearson bound but this time on p_α which is known. Therefore after computing τ_α we use $p_\alpha^{\downarrow}$ instead which is a minimum support for all τ_i-s set as the p_α quantile of the sampled scores (with adjusted $1 - \eta$ confidence).

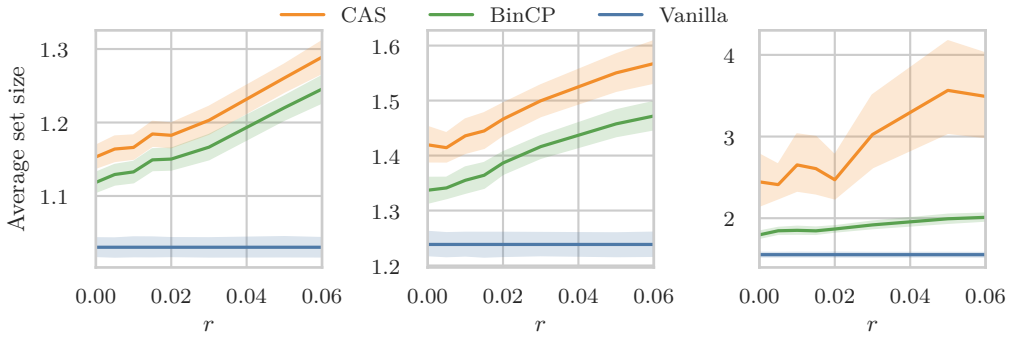

Figure D.2: Comparison between BinCP and CAS on CIFAR-10 dataset with $\sigma = 0.25$ and small values of r. The nominal coverage $1 - \alpha$ is set to [From left to right] 85%, 90%, and 95%.

D.4 Supplementary Discussion

D.4.1 High-level understanding of robustness certificates

A certificate of robustness is a formal guarantee that the model predicts the same class for any perturbation within the specified threat model - within the ball around the input. In other words, if the function f is certified to be robust for the point $\boldsymbol{x}$ w.r.t. $\mathcal{B}$, for any $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$ we have $\arg\max_y f_y(\boldsymbol{x}) = \arg\max_y f_y(\tilde{\boldsymbol{x}})$. This certificate ensures that the top label remains the same within the threat model (binary certificate). A similar (confidence, or soft) certificate guarantees a lower (and upper) bound on the model confidence within the threat model given the predictive probability. One way to attain such certificates is through verifiers. Verifiers need white-box access (knowledge about the model structure and weights) and they work efficiently only on a limited class of models. However, our robust conformal guarantee is black-box.

A common approach for black-box certification is through randomized smoothing. A randomly smoothed classifier results from inference given the input augmented with random noise. For example $g(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})}[f(\boldsymbol{x} + \boldsymbol{\epsilon})]$ – model g returns the expected output of f given randomly augmented $\boldsymbol{x}$ where the noise comes from an isotropic Gaussian distribution with scale σ. The randomization function is smooth even if the original function changes rapidly, which is the effect of the expectation. It is also Lipschitz continuous, meaning that we can bound the output based on the distance of $\tilde{\boldsymbol{x}}$ from $\boldsymbol{x}$. The latter allows us to provide formal guarantees that the top class probability (or confidence) remains high (changes slowly) even if $\tilde{\boldsymbol{x}} \in \mathcal{B}$ is passed to the model instead of $\boldsymbol{x}$.

Ultimately a randomized smoothing-based certificate returns a lower (or upper) bound probability (or score) on the expected output (given the randomized $\boldsymbol{x}$). In robust CP we use these bounds to answer "if instead of the clean input $\boldsymbol{x}_{n+1}$ which is already exchangeable with the calibration set, the model received the worst case $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$ how much lower the conformity score has become". Or in other words "if the model is queried with $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$ (which has a lower conformity score in order not to be covered) how much higher the conformity score of the clean input can be". Technical details of smoothing-based certificates are mentioned in § 6.4. For a more detailed discussion see (Cohen et al., 2019; Kumar et al., 2020).

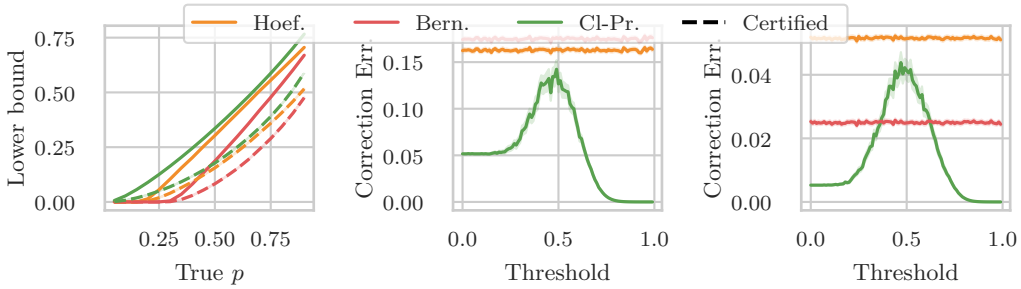

Figure D.3: [Left] Confidence lower bound and the corresponding certified lower bound for scores derived from Beta and Bernoulli distributions. [Middle and right] Correction error (lower bound subtracted from the theoretical mean) of the scores distributed from the Gaussian distribution both in continuous case (mean lower bound) and binarized case (lower bound on the Bernoulli parameter) for [Middle] 100 and [Right] 1000 samples. Details of the experiments are in subsubsection D.4.2.

D.4.2 Comparison of confidence intervals

As discussed in § 6.5, BinCP, CAS, and RSCP, all require true probability, CDF, and mean from the distribution of scores which is intractable to compute (except in the case of de-randomized BinCP). Therefore we use confidence bounds that are lower (or higher) than the true values with collective probability $1-\eta$ (which is taken into account while calibrating). CAS, and RSCP are defined on continuous scores that are bounded by Hoeffding, Bernstein, or DKW inequalities. BinCP is defined through binarized scores, and the final parameter is the success probability of a Bernoulli distribution which can bounded by the Clopper-Pearson interval which is exact (Clopper and Pearson, 1934). The width of all mentioned confidence intervals is decreasing w.r.t. the sample size. Therefore a tighter interval can result in the same or better efficiency (correction error) with fewer samples; e.g. For scores sampled from a Gaussian $\mathcal{N}(0.5, 0.1)$ Clopper Pearson error (for $z \geq 0.6$) with 100 samples is still lower than Bernstein's error with 250 samples.

To illustrate this we conducted two experiments. First, to compare the tightness of each concentration inequality, we sampled from a Beta distribution with mean p to have continuous score values between $[0, 1]$. The distribution for a fixed β is Beta$(\frac{p}{\beta(1-p)}, \beta)$. Then for the continuous score, we computed both Hoeffding's and Bernstein's lower bound on the mean, alongside the Clopper-Pearson bound for the given parameter p and the same sample size. As shown in Fig. D.3-left (with $\beta = 1$) the binary lower bound is always higher (better). Since the certified lower bound is an increasing function of the given probability, the certified lower bound for the binary values is again higher.

In another experiment shown in Fig. D.3-middle and right we sample scores from a Gaussian distribution $\mathcal{N}(0.5, 0.1)$, and computed the lower-bound mean given both Hoeffding and Bernstein's inequalities. Then for various thresholds, we computed the probability of scores passing that threshold and lower bounded this probability by Clopper Pearson concentration inequality. As shown in the figure for lower sample rates, binarization results in less error compared to the theoretical mean. Even with higher sample rates Clopper Pearson interval is significantly tighter than the other two for low and high thresholds (which is a parameter of BinCP).

Proposition D.2. *Let* $X \sim \text{Beta}(a, b)$ *and* $x_1, \ldots, x_m$ *be* m *i.i.d. samples of* X*. Given the empirical mean* $\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$ *the upper bound for the true mean* $\mu = \mathbb{E}[X]$ *is given by the*

Hoeffding's inequality as $\mu \leq \bar{x} + b_{\text{hoef}}$*, where* $b_{\text{hoef}} = \sqrt{\frac{\ln\left(\frac{1}{\eta}\right)}{2m}}$*. For any user-specified* $\tau \in (0,1)$*, let* $Y = \mathbb{I}[X > \tau]$*. The Clopper-Pearson upper bound* p_u *for the true* $p = \mathbb{E}[Y] = \Pr[X \geq \tau]$ *is:*

$$p_u = \Phi^{-1}_{\text{Beta}}(1-\eta; 1 + \sum_{i=1}^{m} \mathbb{I}[x_m > \tau], m - \sum_{i=1}^{m} \mathbb{I}[x_m > \tau_\mu])$$

Each upper bound holds with probability $1-\eta$*. For any number of samples* m*, and any significance level* η*, the probability that the CP bound is tighther is* $\Pr[p_u - \mathbb{E}[Y] \leq b_{\text{hoef}}]$ *and equals:*

$$\Pr[p_u - \mathbb{E}[Y] \leq b_{\text{hoef}}] = \Phi_{\text{Binom}}(\hat{m}; m, \mathbb{E}[Y]) \tag{101}$$

where $\hat{m}$ *is defined in Eq. 103.*

Proof. The variable Y is distributed as $Y \sim \text{Bernoulli}(p)$ where $p = \mathbb{E}[Y] = 1 - \Phi_{\text{Beta}}(\tau; a, b)$ and Φ_{Beta} is the CDF of the beta distribution with parameters a and b.

Let $m_+ = \sum_{i=1}^{m} \mathbb{I}[x_m > \tau]$. We will compute the probability that the inequality

$$p_u - \mathbb{E}[Y] \leq \sqrt{\frac{\ln\left(\frac{1}{\eta}\right)}{2m}} \tag{102}$$

holds. Substituting the definition of p_u and $\mathbb{E}[Y]$ we get:

$$\Phi^{-1}_{\text{Beta}}(1-\eta; 1+m_+, m-m_+) - (1 - \Phi_{\text{Beta}}(\tau; a, b)) \leq \sqrt{\frac{\ln\left(\frac{1}{\eta}\right)}{2m}} \Leftrightarrow$$

$$1 - \eta \leq \Phi_{\text{Beta}}\left(\sqrt{\frac{\ln\left(\frac{1}{\eta}\right)}{2m}} + 1 - \Phi_{\text{Beta}}(\tau; a, b); 1 + m_+, m - m_+\right) \Leftrightarrow$$

$$\eta \geq 1 - \Phi_{\text{Beta}}\left(\sqrt{\frac{\ln\left(\frac{1}{\eta}\right)}{2m}} + 1 - \Phi_{\text{Beta}}(\tau; a, b); 1 + m_+, m - m_+\right)$$

Define $\hat{m}$ as the break-point after which the Clopper-Pearson bound becomes looser than the Hoeffding bound:

$$\hat{m} = \sup\left\{ m_+ : \mathbb{I}\left[\eta \geq 1 - \Phi_{\text{Beta}}\left(\sqrt{\frac{\ln\left(\frac{1}{\eta}\right)}{2m}} + 1 - \Phi_{\text{Beta}}(\tau; a, b); 1 + m_+, m - m_+\right)\right]\right\} \tag{103}$$

In other words, $m_+ > \hat{m} \Leftrightarrow p_u - \mathbb{E}[E] > b_{\text{hoef}}$. Since Φ is monotonic, it follows that:

$$\Pr[p_u - \mathbb{E}[Y] > b_{\text{hoef}}] = \Pr[m_+ > \hat{m}] = 1 - \Phi_{\text{Binom}}(\hat{m}; m, \mathbb{E}[Y]) \tag{104}$$

Where Φ_{Binom} is the CDF of the Binomial distribution with the specified parameters.

□

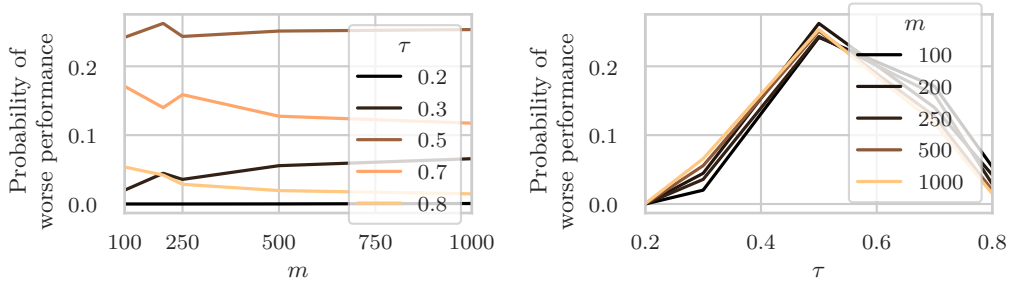

Figure D.4: Probability of observing higher upper bound from Clopper Pearson confidence interval in comparison with Hoeffding's interval. The result is for Beta(2, 2), and $\eta = 0.01$.

Similarly, we can compare Clopper-Pearson bound with the Bernstein bound we use $\mu \leq \bar{x} + b_{\text{bern}}$ where

$$b_{\text{bern}} = \sqrt{2\sigma_m^2 \frac{\ln\left(\frac{2}{\eta}\right)}{m}} + \frac{7\ln\left(\frac{2}{\eta}\right)}{3(m-1)}$$

By replacing b_{hoef} with b_{bern} in Proposition D.2 we can derive a similar result. We choose a Beta distribution to simulate the fact that conformity scores such as TPS and APS are bounded. Moreover, we need bounded scores to be able to apply Hoeffding's inequality. Any other distribution (after some transformation that ensures bounded scores) could be used, as long as we can compute its CDF.

In Fig. D.4 we show the probability of Hoeffding bound being tighter than Clopper-Pearson bound (it's complement is defined in Eq. 104) for $X \sim \text{Beta}(2, 2)$ for different values of m and τ. There is a choice of τ such that the probability is effectively 0 for all values of m, i.e. the CP bound is always better. Interestingly, at worst, both in terms of the number of samples m and τ, we see that it is less than 25%. That is, Clopper-Pearson bound is better on average for all configurations.

To get some additional intuition, instead of the exact Clopper-Pearson bound for p we can use the following bound derived from a Normal approximation which approximately holds with probability $1 - \eta$:

$$p \leq \hat{p} + \frac{z_\eta}{\sqrt{m}}\sqrt{\hat{p}(1-\hat{p})}$$

where $\hat{p} = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}[x_m > \tau]$ and z_η is the $1 - \eta$ quantile of the standard normal distribution. It is not difficult to verify that for all values of $\hat{p} \in [0, 1]$ we have that

$$\frac{z_\eta}{\sqrt{m}}\sqrt{\hat{p}(1-\hat{p})} \leq \sqrt{\frac{\ln\left(\frac{1}{\eta}\right)}{2m}}$$

To see this, note that the $\sqrt{m}$ term cancels, and for $\eta = 0.05$ $z_\eta \approx 1.64$, $\sqrt{\frac{\ln\left(\frac{1}{\eta}\right)}{2}} \approx 1.22$. Since $\sqrt{\hat{p}(1-\hat{p})} \in [0, 0.5]$, even in the worst-case $1.64 \cdot 0.5 \leq 1.22$. This analysis again confirms that CP gives tighter bounds. Proposition D.2 can be analogues extended to lower bounds.

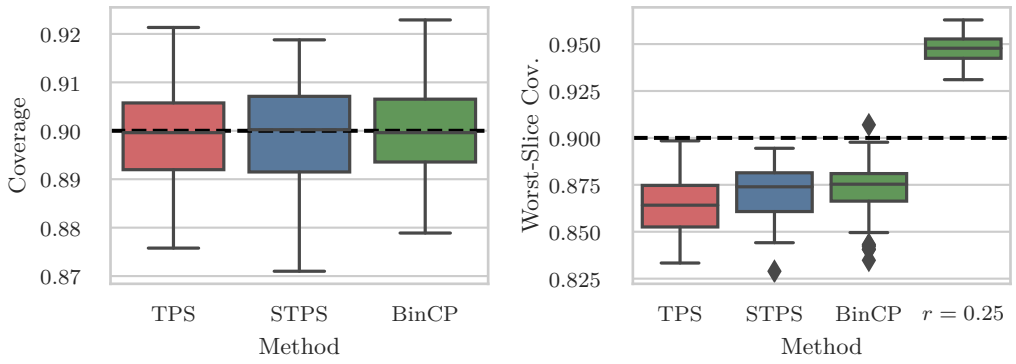

Figure D.5: Comparison of coverage [Left] and worst-slice coverage [Right]. Here the STPS refers to the smooth TPS which is the average of 2000 randomly smooth inferences per point. The results are for CIFAR-10 dataset and r= 0 unless specified.

Finite sample correction for fixed τ setup. What we showed in Proposition 6.7 adds MC sample correction to BinCP with fixed τ computation. We can correct for finite samples in a fixed p setup in a similar way. First, we compute the $\tau_i(p)$ for each of the calibration points. In an asymptomatically valid setup this implies that for $\tau_i(p)$ we have $\Pr[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}, y_i) \geq \tau_i(p)] \geq p$. To account for finite samples we reduce p to $p^{\downarrow}$ ($p^{\downarrow} \leq p$). Again this holds for each calibration with $1/(|\mathcal{D}_{\text{cal}}| + k)$ probability, and the conformal threshold is $(\tau_p, p^{\downarrow})$. In the test time the setup is identical to the fixed-τ setup.

D.4.3 Realistic setup to evaluate GNN robustness

Concurrent to our work Gosch et al. (2024) shows that transductive setup is flawed to evaluate GNN robustness. This is since assuming that the clean graph is accessible to the defender during training, can lead to perfect robustness by simply remembering the clean graph. Many robust- and self-training GNNs proposed for robustness also exploit this flaw.

To evaluate GNN node-classification for robust CP, following prior works, we assumed the transductive setup where the clean graph is given during the training and calibration. Specifically, we assume that the defender trains, and computes the calibration scores on the clean graph. Perturbations in test nodes are then applied after calibration (in evasion setup). This again is based on similar assumption made by many other GNN robustness works. Therefore our robust node-classification results are only representing the comparison of BinCP and other approaches on a sparse binary certificate.

A more realistic setup is the inductive setup where the defender is given a clean subgraph $\mathcal{G}_{\text{tr}}$ for training (and calibration), and the adversary perturbs the rest of the graph upon arrival; i.e. the defender does not know the clean test graph. In vanilla setup, the conformal guarantee is still valid in the inductive setting via re-computing calibration nodes (Anonymous, 2024b). However in evasion, robust CP is not as easily applicable by recomputing the quantile and computing upper bounds on scores, since the calibration nodes are also affected by the adversarial test nodes through the message passing. Therefore robustness in GNNs under realistic setups is challenging which we leave it to future works.

D.5 Additional Experiments

Various models and smoothing maginitudes (σ). In Table D.2 we compare the result between SOTA CAS and BinCP for CIFAR-10 dataset. The results are reported across various data smoothing σ values, and models trained with different noise augmentations (data augmented during training with different σ values). We call them smoothing σ, and model σ respectively. In the robustness certificate for classification, it is considered best practice to use the same σ in both model's noise augmentation, and the smoothing process. Similarly, in robust conformal prediction mismatching smoothing and model σ results in a larger prediction set. Interestingly this adverse effect is much less observed in BinCP although it remains present. Overall, across all smoothing parameters, model σ values, coverage rates, and perturbation radii, BinCP consistently outperforms CAS.

Performance on small radii. For completeness, in Fig. D.2, we report the performance of BinCP on small values of r. As Jeary et al. (2024) reports ~ 4.45 average set size for $r = 0.02$ (Table 1 in (Jeary et al., 2024)) our report shows more than twice smaller sets for the same r. As in Table D.2 we observe the same average set size for $r \sim 0.5$ ($\geq 20\times$ higher radius) for smallest $\sigma = 0.12$. As we discussed, one effect of this eye-catching difference is the inherent robustness of the randomized smooth prediction. As shown in Fig. 6.6, the empirical coverage of non-smooth prediction drastically decreases to 0 for small radii, while in smooth prediction the coverage decreases slowly.

Table D.1: Comparison of smoothing-based robust CP methods on APS score

		$1-\alpha = 0.9$				$1-\alpha = 0.95$			
		CAS		UNKNOWNMETHOD!!!		CAS		UNKNOWNMETHOD!!!	
σ	r	**Coverage**	**Set Size**	**Coverage**	**Set Size**	**Coverage**	**Set Size**	**Coverage**	**Set Size**
0.12	0.06	0.954	1.635	0.946	1.529	0.990	4.022	0.980	2.151
	0.12	0.971	1.939	0.963	1.757	0.996	6.435	0.985	2.389
	0.18	0.987	2.879	0.978	2.076	1.000	9.745	0.991	2.876
	0.25	0.998	7.454	0.986	2.510	1.000	10.000	0.995	3.405
	0.37	1.000	10.000	1.000	10.000	1.000	10.000	1.000	10.000
	0.50	1.000	10.000	1.000	10.000	1.000	10.000	1.000	10.000
	0.75	1.000	10.000	1.000	10.000	1.000	10.000	1.000	10.000
0.25	0.06	0.955	2.108	0.944	1.894	0.986	3.316	0.976	2.677
	0.12	0.964	2.309	0.954	2.054	0.989	3.682	0.980	2.857
	0.18	0.970	2.495	0.961	2.227	0.993	5.038	0.986	3.181
	0.25	0.980	2.900	0.972	2.537	0.997	6.004	0.989	3.444
	0.37	0.991	3.795	0.982	3.047	1.000	9.360	0.994	4.035
	0.50	0.999	7.430	0.991	3.729	1.000	10.000	0.997	4.850
	0.75	1.000	10.000	1.000	10.000	1.000	10.000	1.000	10.000
0.50	0.06	0.956	2.738	0.942	2.479	0.981	3.864	0.975	3.342
	0.12	0.962	2.890	0.951	2.635	0.984	4.077	0.978	3.508
	0.18	0.968	3.078	0.959	2.801	0.986	4.277	0.981	3.658
	0.25	0.973	3.304	0.966	2.994	0.989	4.546	0.984	3.899
	0.37	0.980	3.684	0.974	3.302	0.993	5.193	0.988	4.300
	0.50	0.986	4.153	0.979	3.663	0.996	5.868	0.991	4.733
	0.75	0.995	5.441	0.989	4.584	0.999	8.026	0.995	5.542

APS score function. Although we observe similar comparison between CAS and BinCP given APS score function, for completeness we report the performance of both methods in Table D.1.

Conditional and class-conditional coverage. Following Romano et al. (2020), we approximated the conditional coverage gap as the worst coverage among n different slices. Each slice is defined as $\mathcal{X}_s = \{\boldsymbol{x}_i \in \mathcal{D} : a \leq \boldsymbol{x}_i \cdot \boldsymbol{v} \leq b\}$ for the random vector $\boldsymbol{v}$ and random scalers a, b (Romano et al., 2020). For that, we sampled 200 random vectors $\boldsymbol{v}$ and among all the scalers randomly sampled a, b from the set $\{\boldsymbol{x}_i \cdot \boldsymbol{v}, \boldsymbol{x}_i \in \mathcal{D}\}$.

We report the result over 100 different calibration samplings. In each iteration of the experiment, we exclude the slices with less than 200 points of support.

BinCP has smoothing, binarization and accounting for perturbation radius. For a better intuition we observe the effect of each step separately: vanilla TPS (without smoothing) as the baseline score, vanilla smooth TPS (labeled as STPS) which is an average of TPS scores over randomized sampled (same as CAS), BinCP without robustness (set $r = 0$) which reflects the effect of binarization (we did not correct for finite sample due to validity in vanilla setup), and robust CP via BinCP. As shown in Fig. D.5, the smooth model has a better worst-slice coverage than vanilla TPS. Though binarization although the average worst-slice coverage remains the same, there is a slight decrease in the variance of this metric. Note that sampling correction and making CP robust increases the empirical coverage guarantee, therefore the worst slice coverage is increased due to the inherent increase in marginal coverage.

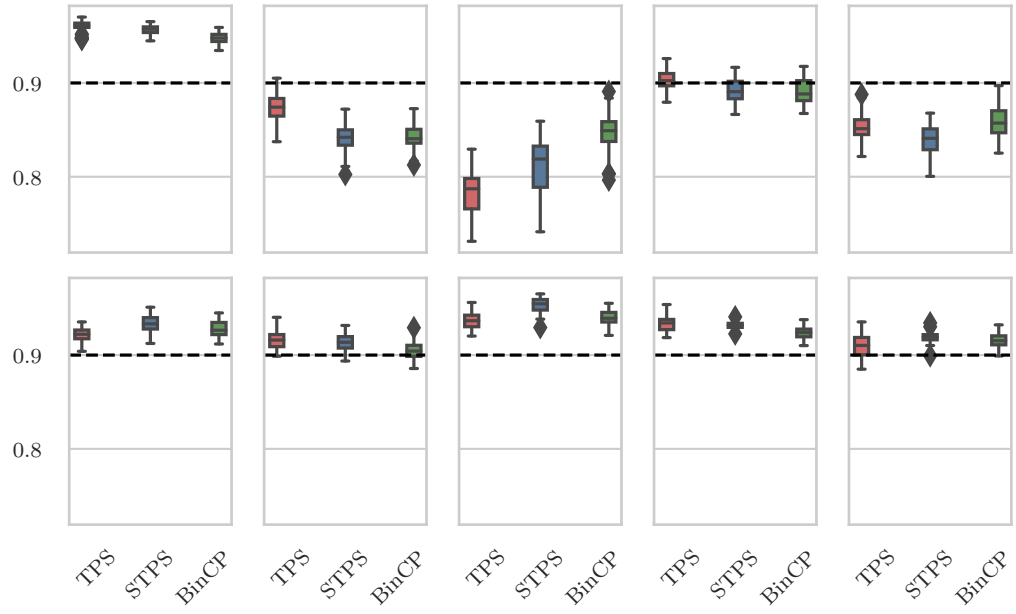

Figure D.6: Comparison of methods in class-conditional coverage for all classes of CIFAR-10, note that here BinCP is used without sampling correction. That is because the correction slightly increases empirical coverage which can be misleading.

We also reported the result of the class-conditional coverage in Fig. D.6. Empirically in almost all classes, BinCP is closer to the nominal guarantee compared to normal smoothing. Ultimately both smooth prediction and BinCP are not comparable with vanilla TPS.

Comparison with RSCP+. Yan et al. (2024) shows a flaw of RSCP (Gendler et al., 2021) indicating that the score function is not corrected for finite sample estimation. They show that by adding finite sample correction to RSCP, it becomes significantly inefficient and produces trivial sets $C(x_{n+1}) = \mathcal{Y}$. They remedy that by designing a ranking-based transformation on top of the given score function which defines a new score as

$$s_{\text{ppt}}(x, y) = \sigma\left(\frac{1}{T|\mathcal{D}_{\text{tune}}|}\text{rank}(s(x, y); \{s(x_j, y_j)\}_{(x_j,y_j)\in\mathcal{D}_{\text{tune}}}) - \frac{b}{T}\right) \tag{105}$$

Where $\mathcal{D}_{\text{tune}}$ is a holdout tuning index, T is the temperature parameter, b is a bias parameter, and σ is the sigmoid function. The original experiment from Yan et al. (2024)

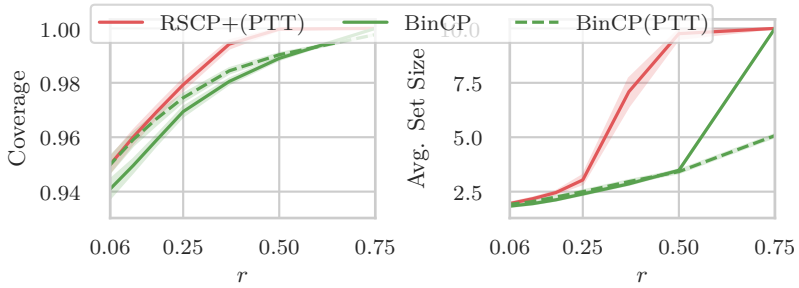

Figure D.7: Comparison between BinCP and RSCP+ (PPT, Eq. 105) and BinCP with Eq. 105. The result is on CIFAR-10 dataset with $\sigma = 0.25$.

has several issues, which we resolved and compared with it: (i) The scores have possible ties; i.e. two different data points can have the same score value. To remedy that we added an unnoticeable random number $\delta \sim \text{Uniform}[0, 1/|\mathcal{D}_{\text{tune}}|]$ to the scores. (ii) The tuning set and the calibration set in the experiments are significantly large. Yan et al. (2024) use 5250/10000 test datapoints as a tuning and calibration set. This unrealistic holdout labeled set contradicts the sparse labeling assumption. In our reproduction of their results we used a total of ~ 380 datapoints where 200 of them are for tuning. As shown in Fig. D.7, BinCP still outperforms RSCP+(PPT). As the score function in Eq. 105 is also a valid score, we can use BinCP on top which shows slightly better efficiency for larger radii compared to BinCP combined with TPS score. Here we set $b = 1 - \alpha$, and $T = 0.001$, and report the results on 2000 MC samples. The reported result is on CIFAR-10 dataset.

			$1-\alpha = 0.9$				$1-\alpha = 0.95$			
			CAS		**UNKNOWNMETHOD!!!**		**CAS**		**UNKNOWNMETHOD!!!**	
Model σ	**Data** σ	r	Coverage	Set Size	Coverage	Set Size	Coverage	Set Size	Coverage	Ave Set Size
0.12	0.12	0.06	0.950	1.581	0.942	1.483	0.987	3.353	0.976	2.009
		0.12	0.968	1.839	0.959	1.671	0.996	6.731	0.986	2.387
		0.18	0.985	2.761	0.974	1.946	0.999	9.417	0.990	2.666
		0.25	0.997	7.078	0.985	2.369	1.000	10.000	0.994	3.213
	0.25	0.06	1.000	10.000	0.943	4.911	1.000	10.000	0.977	6.500
		0.12	1.000	10.000	0.953	5.328	1.000	10.000	0.984	6.880
		0.18	1.000	10.000	0.961	5.711	1.000	10.000	0.991	7.366
		0.25	1.000	10.000	0.974	6.323	1.000	10.000	0.994	7.628
		0.37	1.000	10.000	0.988	7.133	1.000	10.000	0.998	8.387
		0.50	1.000	10.000	0.996	7.990	1.000	10.000	0.999	9.049
		0.75	1.000	10.000	1.000	10.000	1.000	10.000	1.000	10.000
	0.50	0.06	1.000	10.000	0.950	8.836	1.000	10.000	0.980	9.310
		0.12	1.000	10.000	0.960	8.985	1.000	10.000	0.984	9.402
		0.18	1.000	10.000	0.965	9.075	1.000	10.000	0.986	9.450
		0.25	1.000	10.000	0.974	9.222	1.000	10.000	0.990	9.535
		0.37	1.000	10.000	0.983	9.403	1.000	10.000	0.996	9.656
		0.50	1.000	10.000	0.991	9.557	1.000	10.000	0.999	9.789
		0.75	1.000	10.000	0.999	9.820	1.000	10.000	1.000	9.947
0.25	0.12	0.06	0.954	2.570	0.937	2.232	0.991	7.016	0.968	2.992
		0.12	0.969	3.049	0.949	2.395	0.998	9.042	0.976	3.301
		0.18	0.984	4.573	0.956	2.562	1.000	9.845	0.981	3.567
		0.25	0.999	9.510	0.969	2.908	1.000	10.000	0.986	3.895
	0.25	0.06	0.953	2.051	0.941	1.836	0.984	3.307	0.974	2.551
		0.12	0.960	2.183	0.950	1.951	0.991	4.077	0.981	2.832
		0.18	0.969	2.411	0.959	2.126	0.994	5.242	0.984	3.054
		0.25	0.979	2.790	0.969	2.394	0.997	6.749	0.988	3.295
		0.37	0.991	3.867	0.981	2.858	1.000	9.660	0.994	3.888
		0.50	0.999	7.824	0.989	3.480	1.000	9.948	0.996	4.564
	0.50	0.06	1.000	10.000	0.947	6.762	1.000	10.000	0.979	7.837
		0.12	1.000	10.000	0.956	7.024	1.000	10.000	0.983	8.016
		0.18	1.000	10.000	0.960	7.212	1.000	10.000	0.988	8.221
		0.25	1.000	10.000	0.969	7.523	1.000	10.000	0.992	8.430
		0.37	1.000	10.000	0.981	7.935	1.000	10.000	0.996	8.763
		0.50	1.000	10.000	0.990	8.350	1.000	10.000	0.998	9.040
		0.75	1.000	10.000	0.997	9.008	1.000	10.000	0.999	9.494
0.50	0.12	0.06	0.948	3.701	0.923	3.060	0.994	8.485	0.965	4.196
		0.12	0.961	4.230	0.929	3.159	0.999	9.564	0.971	4.457
		0.18	0.980	5.843	0.937	3.330	1.000	9.960	0.973	4.538
		0.25	0.998	9.329	0.947	3.550	1.000	10.000	0.977	4.753
	0.25	0.06	0.943	3.152	0.925	2.792	0.990	7.254	0.969	4.013
		0.12	0.951	3.417	0.933	2.929	0.995	7.818	0.973	4.172
		0.18	0.961	3.771	0.942	3.112	0.996	8.476	0.976	4.276
		0.25	0.970	4.095	0.948	3.246	0.998	8.916	0.978	4.416
		0.37	0.987	5.773	0.959	3.586	1.000	9.958	0.984	4.753
		0.50	0.999	9.121	0.970	3.952	1.000	10.000	0.988	5.171
	0.50	0.06	0.957	2.767	0.943	2.428	0.984	3.995	0.974	3.288
		0.12	0.964	2.948	0.949	2.558	0.986	4.165	0.977	3.405
		0.18	0.968	3.071	0.955	2.673	0.988	4.351	0.980	3.542
		0.25	0.974	3.272	0.962	2.835	0.991	4.850	0.983	3.806
		0.37	0.982	3.721	0.973	3.182	0.993	5.160	0.986	4.044
		0.50	0.987	4.215	0.980	3.524	0.997	6.439	0.990	4.511
		0.75	0.996	5.708	0.987	4.285	1.000	9.287	0.994	5.316

Table D.2: Comparison of CAS and UNKNOWNMETHOD!!! for model trained with various smoothing σ, and input data with different smoothing σ. Results are for CIFAR-10 dataset.

E Supplementary Material: One Sample is Enough to Make Conformal Prediction Robust

E.1 More on Conformal Prediction

Our default score function in the manuscript is TPS (threshold prediction sets) where the score function is directly set to the softmax; $s(\boldsymbol{x}, y) = \text{Softmax}_y(f(\boldsymbol{x}))$ for the prediction model f. Another choice is to use the logits of the model as the conformity score $s(\boldsymbol{x}, y) = f(\boldsymbol{x})_y$. Similar to BinCP we are using a single binary certificate which do not rely on bounded score function. Another conformal prediction method called as adaptive prediction sets (APS) uses the accumulated softmax up to label y as the conformity score; formally $s(\boldsymbol{x}, y) = -[\pi(\boldsymbol{x}, y) \cdot u + \sum_{k=1}^{|\mathcal{Y}|} \pi(\boldsymbol{x}, y) \cdot \mathbb{I}[\pi(\boldsymbol{x}, y_k) > \pi(\boldsymbol{x}, y)]]$ where $\pi(\boldsymbol{x}, y) = \text{Softmax}_y(f(\boldsymbol{x}))$ and $u \sim \text{Uniform}[0, 1]$. While this score results in larger sets, it increases adaptivity – approximate conditional coverage.

We report the result using these score functions in Fig. E.1 (comparison of BinCP and RCP1 for each score) and Table E.1 (the set size of RCP1 for both score functions). As expected, similar trend as TPS is observed for APS as well.

Same as BinCP we also do not need the score function to be bounded. However in the end, using an unbounded score function (like using logits directly) did not show to improve over the existing APS and TPS.

E.2 Supplementary to Theory

E.2.1 Robust Conformal Prediction Guarantee

The guarantee in Eq. 62 doesn't take the randomness in score function and prediction set into account. This is while many conformal scores have a random variable inside, for instance APS Romano et al. (2020) multiplies the probability of each class by a uniform random value to break the ties (and enable the exact $1 - \alpha$ coverage). The original guarantee taken from Ghosh et al. (2023); Zargarbashi et al. (2024) is:

$$\begin{aligned} &\Pr_{\mathcal{D}_{\text{cal}}, \boldsymbol{x}_{n+1} \sim \mathcal{D}} [y_{n+1} \in C_{\mathcal{B}}(\tilde{\boldsymbol{x}}_{n+1}), \forall \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})] \geq 1 - \alpha \\ \equiv &\Pr_{\mathcal{D}_{\text{cal}}, \boldsymbol{x}_{n+1} \sim \mathcal{D}} [\inf_{\forall \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})} \mathbb{I}[y_{n+1} \in C_{\mathcal{B}}(\tilde{\boldsymbol{x}}_{n+1})] = 1] \geq 1 - \alpha \end{aligned}$$

This formulation fails to capture the stochasticity in the score function (and hence in the prediction set). Since with a very small probability to miscover a point (which

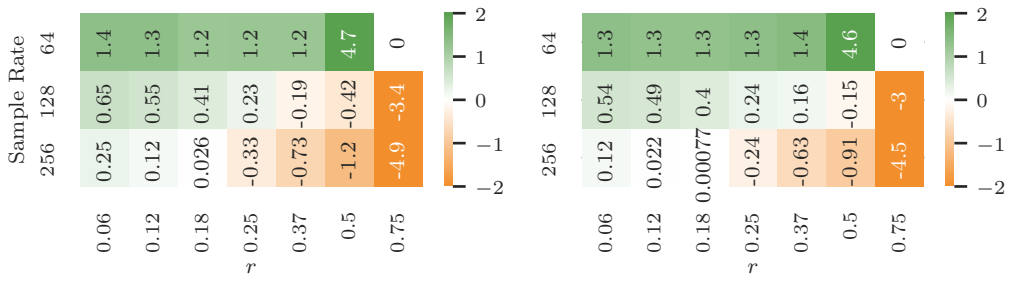

Figure E.1: Comparison of BinCP and RCP1 for [Left] TPS, and [Right] APS score function on CIFAR-10 dataset with $\sigma = 0.9$.

Table E.1: Set size of RCP1 for TPS and APS score across radii (r) and target coverage guarantees.

r	Coverage	TPS		APS	
		Avg Set Size	Emp. Cov.	Avg Set Size	Emp. Cov.
0.06	0.85	2.17 ± 0.02	0.88 ± 0.00	2.52 ± 0.03	0.88 ± 0.00
	0.90	2.70 ± 0.03	0.92 ± 0.00	2.98 ± 0.02	0.92 ± 0.00
	0.95	3.74 ± 0.01	0.97 ± 0.00	3.91 ± 0.04	0.97 ± 0.00
0.12	0.85	2.44 ± 0.03	0.90 ± 0.00	2.76 ± 0.02	0.90 ± 0.00
	0.90	2.93 ± 0.04	0.94 ± 0.00	3.23 ± 0.01	0.94 ± 0.00
	0.95	3.96 ± 0.06	0.97 ± 0.00	4.07 ± 0.04	0.97 ± 0.00
0.18	0.85	2.70 ± 0.04	0.92 ± 0.00	2.99 ± 0.01	0.92 ± 0.00
	0.90	3.25 ± 0.05	0.95 ± 0.00	3.44 ± 0.03	0.95 ± 0.00
	0.95	4.48 ± 0.09	0.98 ± 0.00	4.48 ± 0.02	0.98 ± 0.00
0.25	0.85	3.03 ± 0.01	0.94 ± 0.00	3.33 ± 0.04	0.94 ± 0.00
	0.90	3.70 ± 0.02	0.97 ± 0.00	3.89 ± 0.03	0.97 ± 0.00
	0.95	4.81 ± 0.01	0.99 ± 0.00	4.82 ± 0.04	0.99 ± 0.00
0.37	0.85	3.62 ± 0.06	0.96 ± 0.00	3.91 ± 0.01	0.97 ± 0.00
	0.90	4.48 ± 0.03	0.98 ± 0.00	4.55 ± 0.09	0.98 ± 0.00
	0.95	6.24 ± 0.14	0.99 ± 0.00	6.49 ± 0.04	1.00 ± 0.00
0.50	0.85	4.51 ± 0.03	0.98 ± 0.00	4.55 ± 0.11	0.98 ± 0.00
	0.90	5.32 ± 0.04	0.99 ± 0.00	5.34 ± 0.03	0.99 ± 0.00
	0.95	10.00 ± 0.00	1.00 ± 0.00	10.00 ± 0.00	1.00 ± 0.00
0.75	0.85	6.28 ± 0.19	0.99 ± 0.00	6.31 ± 0.11	0.99 ± 0.00
	0.90	10.00 ± 0.00	1.00 ± 0.00	10.00 ± 0.00	1.00 ± 0.00
	0.95	10.00 ± 0.00	1.00 ± 0.00	10.00 ± 0.00	1.00 ± 0.00

is non-zero in methods like APS) the indicator evaluates to false. The worst-case indicator $\inf_{\forall \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})} \mathbb{I}[y_{n+1} \in C_{\mathcal{B}}(\tilde{\boldsymbol{x}}_{n+1}) = 1]$ becomes zero whenever there exists any probability that $\boldsymbol{x}_{n+1}$ is not covered. Consider the non-robust case where no perturbations are applied; i.e., evaluating the coverage guarantee of standard conformal prediction. Using the APS score function and shrinking the perturbation space to an infinitesimal ball $\mathcal{B}_r$ as $r \to 0^+$ (and therefore $\tilde{\boldsymbol{x}}_{n+1} \to \boldsymbol{x}_{n+1}$), the coverage probability should exceed $1 - \alpha$, since APS already satisfies this guarantee. However many of the datapoints that have a small probability to exclude the true class from the prediction set will not pass the worst-case indicator. Consider a datapoint with one hot conditional probability, still the top label will be in the prediction set with probability $-q$ where q is the threshold.

While the shortcoming in Eq. 62 excludes any CP with randomness in the score, still previous smoothing-based RCP methods (at least in assymptotically valid setup) satisfy its conditions independent of the score function used. This is since all previous methods systematically remove all the randomness from the score and return a deterministic prediction set. Given any base score function, these methods defined their own score as a statistic (e.g. mean, or quantile) over the distribution of the base score on $\boldsymbol{x} + \boldsymbol{\epsilon}$. As the distribution already includes the inherent randomness in the base score itself, the statistics like mean Gendler et al. (2021); Zargarbashi et al. (2024) and the quantile Zargarbashi and Bojchevski (2025) are deterministic (excluding any randomness). This is an orthogonal to the probabilistic nature of estimating these statistics from Monte Carlo samples (the validity of confidence intervals). As a result the final set based on

these scores exclude the inherent randomness of the base score function.

E.2.2 Proofs

Proof of Lemma 7.2. The Lagrangian form of Eq. 64 is:

$$\mathcal{L}(\beta,\lambda) = \min_{h\in\{0,1\}^{|\mathcal{X}|}} \Pr_{\boldsymbol{\epsilon}}[h(\tilde{\boldsymbol{t}}+\boldsymbol{\epsilon})=1] - \lambda\left(\Pr_{\boldsymbol{\epsilon}}[h(\boldsymbol{t}+\boldsymbol{\epsilon})=1]-\beta\right)$$
$$= \min_{h\in\{0,1\}^{|\mathcal{X}|}} E_{\boldsymbol{z}\sim q}[h(\boldsymbol{z})] - \lambda\cdot E_{\boldsymbol{z}\sim p}[h(\boldsymbol{z})-\beta] = \lambda\cdot\beta + \min_{h\in\{0,1\}^{|\mathcal{X}|}} E_{\boldsymbol{z}\sim q}[h(\boldsymbol{z})] - E_{\boldsymbol{z}\sim p}[h(\boldsymbol{z})]$$
$$= \lambda\cdot\beta + \min_{h\in\{0,1\}^{|\mathcal{X}|}} \int_{\mathcal{X}} (q(\boldsymbol{z}) - \lambda\cdot p(\boldsymbol{z}))\cdot h(\boldsymbol{z})\mathrm{d}\boldsymbol{z}$$

where p and q are smoothing distributions centered at $\boldsymbol{t}$ and $\tilde{\boldsymbol{t}}$ respectively. The worst classifier (the minimizer of the problem) can be derived as follows

$$h(\boldsymbol{z}) = \begin{cases} 0 & \text{if} \quad q(\boldsymbol{z}) - \lambda\cdot p(\boldsymbol{z}) \geq 0 \\ 1 & \text{otherwise} \end{cases}$$

Intuitively, to minimize the term $\int_{\mathcal{X}}(q(\boldsymbol{z}) - \lambda\cdot p(\boldsymbol{z}))\cdot h(\boldsymbol{z})\mathrm{d}\boldsymbol{z}$ we look at each point independently. For each point if the term $(q(\boldsymbol{z}) - \lambda\cdot p(\boldsymbol{z}))$ is positive we cancel it by $h=0$ and if negative we keep it to decrease the total integral value. The resulting dual with the dual variable $\lambda\geq 0$ is then:

$$\mathcal{L}(\beta,\lambda) = \lambda\cdot\beta + \int \min\{0, p(\boldsymbol{z}) - \lambda\cdot q(\boldsymbol{z})\mathrm{d}\boldsymbol{z}\} = \lambda\cdot\beta + l(\lambda)$$

Here $l(\lambda) := \int \min\{0, p(\boldsymbol{z}) - \lambda\cdot q(\boldsymbol{z})\mathrm{d}\boldsymbol{z}\}$ is only a function of λ. Maximizing over λ we get the optimal dual soltuion which equals the optimal primal since it was shown that strong duality holds (Zhang et al., 2020):

$$\mathrm{c}^{\downarrow}[\beta,\mathcal{B}] = \max_{\lambda} \lambda\cdot\beta + l(\lambda)$$

Which is pointwise maximum of affine functions and therefore convex in β.

The monotonicity w.r.t. β directly follows from the definition. By increasing β the feasible space reduces to a nested subset of the previous problem which means that the solution will be greater than or equal to the original solution. □

E.2.3 Choosing the conservative $1-\alpha'$

In general to obtain $1-\alpha$ robust coverage guarantee, we should choose the nominal $1-\alpha'$ in Algorithm 2 such that $\mathrm{c}^{\downarrow}[1-\alpha,\mathcal{B}]\geq 1-\alpha$. This nominal probability can be found via binary search due to the non-decreasing nature of $\mathrm{c}^{\downarrow}$. But in many cases including the Gaussian distribution, where the canonical points can be used interchangeably (choosing $(\boldsymbol{t},\tilde{\boldsymbol{t}})$ as the pair of clean, and noisy canonical points doesn't differ from the opposite $(\tilde{\boldsymbol{t}},\boldsymbol{t})$, the following lemma allows us to set the $1-\alpha' = \mathrm{c}^{\uparrow}[1-\alpha,\mathcal{B}^{-1}]$.

Lemma E.1. *If for a smoothing scheme, and a perturbation ball $\mathcal{B}$, canonical points $\boldsymbol{t}$, and $\tilde{\boldsymbol{t}}$ can be used interchangeably; then we have $\mathrm{c}^{\uparrow}[\mathrm{c}^{\downarrow}[p,\mathcal{B}],\mathcal{B}^{-1}] = p$. Using the canonical points interchangeably means that for $\mathrm{c}^{\uparrow}$, and (similarly $\mathrm{c}^{\downarrow}$) both of the optimizations*

$$\max_{h\in\mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[h(\tilde{\boldsymbol{t}}+\boldsymbol{\epsilon})=1] \quad s.t. \quad \Pr_{\boldsymbol{\epsilon}}[h(\boldsymbol{t}+\boldsymbol{\epsilon})=1] = p$$

and

$$\max_{h\in\mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[h(\boldsymbol{t}+\boldsymbol{\epsilon})=1] \quad s.t. \quad \Pr_{\boldsymbol{\epsilon}}[h(\tilde{\boldsymbol{t}}+\boldsymbol{\epsilon})=1]=p$$

yield the same solution.

Proof. The term $\mathrm{c}^{\uparrow}[\mathrm{c}^{\downarrow}[p,\mathcal{B}],\mathcal{B}^{-1}]$ is expressed as the following optimization problem:

$$\begin{aligned} p^*_{\text{high}} &= \max_{h\in\mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[h(\tilde{\boldsymbol{t}}+\boldsymbol{\epsilon})=1] \quad \text{s.t.} \quad \Pr_{\boldsymbol{\epsilon}}[h(\boldsymbol{t}+\boldsymbol{\epsilon})=1]=p^*_{\text{low}} \\ p^*_{\text{low}} &= \min_{h'\in\mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[h'(\tilde{\boldsymbol{t}}+\boldsymbol{\epsilon})=1] \quad \text{s.t.} \quad \Pr_{\boldsymbol{\epsilon}}[h'(\boldsymbol{t}+\boldsymbol{\epsilon})=1]=\Pr_{\boldsymbol{\epsilon}}[f(\boldsymbol{t}+\boldsymbol{\epsilon})]=p \end{aligned} \tag{106}$$

We swap $\tilde{\boldsymbol{t}}$, and $\boldsymbol{t}$ since we can use the canonical points interchangeably. We have

$$p^*_{\text{high}} = \max_{h\in\mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[h(\boldsymbol{t}+\boldsymbol{\epsilon})=1] \quad \text{s.t.} \quad \Pr_{\boldsymbol{\epsilon}}[h(\tilde{\boldsymbol{t}}+\boldsymbol{\epsilon})=1]=p^*_{\text{low}}$$

h^*_{low} as the solution to the inner problem in Eq. 106 (defining p^*_{low}) is a feasible solution to the outer optimization (the first line); therefore

$$p^*_{\text{high}} = \max_{h\in\mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[h(\boldsymbol{t}+\boldsymbol{\epsilon})=1] \geq \Pr_{\boldsymbol{\epsilon}}[h^*_{\text{low}}(\boldsymbol{t}+\boldsymbol{\epsilon})=1]=\Pr_{\boldsymbol{\epsilon}}[f(\boldsymbol{t}+\boldsymbol{\epsilon})]=p$$

Both functions $\mathrm{c}^{\uparrow}$, and $\mathrm{c}^{\downarrow}$ (and therefore both minimization and maximization) are non-decreasing to the value in their constraint. Assuming $p^*_{\text{high}} > p$ ($p^*_{\text{high}} \neq p$), we have $p = \mathrm{c}^{\uparrow}[p'_{\text{low}}, \mathcal{B}^{-1}]$ that $p'_{\text{low}} < p^*_{\text{low}}$ (due to non-decreasing nature of $\mathrm{c}^{\uparrow}$). We have

$$p = \max_{h\in\mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[h(\boldsymbol{t}+\boldsymbol{\epsilon})=1] \quad \text{s.t.} \quad \Pr_{\boldsymbol{\epsilon}}[h(\tilde{\boldsymbol{t}}+\boldsymbol{\epsilon})=1]=p'_{\text{low}}$$

with a maximizer function h'_{high}; i.e. $p = \Pr_{\boldsymbol{\epsilon}}[h'_{\text{high}}(\boldsymbol{t}+\boldsymbol{\epsilon})=1]$. We rewrite inner problem in Eq. 106

$$p^*_{\text{low}} = \min_{h'\in\mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[h'(\tilde{\boldsymbol{t}}+\boldsymbol{\epsilon})=1] \quad \text{s.t.} \quad \Pr_{\boldsymbol{\epsilon}}[h'(\boldsymbol{x}+\boldsymbol{\epsilon})=1]=\Pr_{\boldsymbol{\epsilon}}[f(\boldsymbol{t}+\boldsymbol{\epsilon})]=p$$

The maximizer function h'_{high} satisfies the constraint, and therefore $p^*_{\text{low}} < p'_{\text{low}}$ which is a contradiction. Therefore $p^*_{\text{high}} = p$. □

E.2.4 Lower and Upper Bounds for All Shapes and Sizes

Lemma E.2. *For a binary classifier $f(\boldsymbol{x})$, let $g(\boldsymbol{x}) = \Pr_{\boldsymbol{\epsilon}}[f(\boldsymbol{x}+\boldsymbol{\epsilon})=1]$ and*

$$g(\tilde{\boldsymbol{x}}) \geq \mathrm{c}^{\downarrow}_{g}[p,\mathcal{B}] := \min_{h\in\mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[h(\tilde{\boldsymbol{x}}+\boldsymbol{\epsilon})=1] \quad s.t. \Pr_{\boldsymbol{\epsilon}}[h(\boldsymbol{x}+\boldsymbol{\epsilon})=1]=g(\boldsymbol{x})=p$$

Similarly let

$$g(\tilde{\boldsymbol{x}}) \leq \mathrm{c}^{\uparrow}_{g}[p,\mathcal{B}] := \max_{h\in\mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[h(\tilde{\boldsymbol{x}}+\boldsymbol{\epsilon})=1] \quad s.t. \Pr_{\boldsymbol{\epsilon}}[h(\boldsymbol{x}+\boldsymbol{\epsilon})=1]=g(\boldsymbol{x})=p$$

Both be obtainable at the same canonical points. We have $\mathrm{c}^{\uparrow}_{g}[p,\mathcal{B}] = 1-\mathrm{c}^{\downarrow}_{g}[1-p,\mathcal{B}]$.

Proof. For simpler notation let $\overline{g}(\boldsymbol{x}) = \mathrm{c}^{\uparrow}_{g}[g(\boldsymbol{x}),\mathcal{B}]$, $\underline{g}(\boldsymbol{x}) = \mathrm{c}^{\downarrow}_{g}[g(\boldsymbol{x}),\mathcal{B}]$, then we have

$$\begin{aligned} 1-\overline{g}(\boldsymbol{x}) &= 1-\max_{h\in\mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[h(\tilde{\boldsymbol{x}}+\boldsymbol{\epsilon})=1] \\ &\text{Let } h'(\boldsymbol{x}) = 1-h(\boldsymbol{x}) \text{ then} \\ &= 1-\max_{h'\in\mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[1-h'(\tilde{\boldsymbol{x}}+\boldsymbol{\epsilon})=1] = \min_{h'\in\mathcal{H}} \Pr_{\boldsymbol{\epsilon}}[h'(\tilde{\boldsymbol{x}}+\boldsymbol{\epsilon})=1] \end{aligned}$$

The constraint also translates similarly

$$\Pr_{\boldsymbol{\epsilon}}[h(\boldsymbol{x}+\boldsymbol{\epsilon})=1]=\Pr_{\boldsymbol{\epsilon}}[1-h'(\boldsymbol{x}+\boldsymbol{\epsilon})=1]=1-\Pr_{\boldsymbol{\epsilon}}[h'(\boldsymbol{x}+\boldsymbol{\epsilon})=1]=1-p$$

And the new problem is by definition same as $1-\mathrm{c}^{\downarrow}[1-p,\mathcal{B}]$. □

Certified Upper and Lower bounds for all Shapes and Sizes. In Proposition 7.3 we rephrased the Theorem 4.1 from Yang et al. (2020) to return the upper bound probability instead of the robust radius. Here we prove Proposition 7.3, using the original proof from Yang et al. (2020).

Let $g(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{\epsilon}}[f(\boldsymbol{x} + \boldsymbol{\epsilon})]$ for any binary decision function f and any $\boldsymbol{\epsilon} \sim \xi$ where $\xi(\boldsymbol{x}) \propto \exp(-\psi(\boldsymbol{x}))$. If $g(\cdot)$ is continuous in $\mathcal{X}$, for any point $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\delta}$ one can compute the $g(\boldsymbol{x} + \tilde{\boldsymbol{\delta}})$ through line integral as

$$g(\tilde{\boldsymbol{x}}) = g(\boldsymbol{x} + \boldsymbol{\delta}) = g(\boldsymbol{x}) + \int_0^r \frac{\mathrm{d}}{\mathrm{d}t}[g(\boldsymbol{x} + t \cdot \boldsymbol{\delta}')]\mathrm{d}t$$

where $\boldsymbol{\delta}' = \frac{\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|}$ is the unit vector in the same direction as $\boldsymbol{\delta}$ and $r := \|\boldsymbol{\delta}\|$. In other words, we add all the infinitisimal changes on the path from $\boldsymbol{x}$ to $\tilde{\boldsymbol{x}}$ to compute the value of $g(\tilde{\boldsymbol{x}})$ given $g(\boldsymbol{x})$.

With the decision boundary $\mathcal{U} = \{\boldsymbol{x} : f(\boldsymbol{x}) = 1\}$, and $\mathcal{U} - \boldsymbol{z}$ as the decision boundary translated by $-\boldsymbol{z}$, consider the following function

$$\Omega(p) := \sup_{\boldsymbol{\delta}:\|\boldsymbol{\delta}\|=1} \sup_{\mathcal{U}\in\mathbb{R}^d:\xi(\mathcal{U})=p} \lim_{r\to 0^+} \frac{\xi(\mathcal{U} - r\boldsymbol{\delta}) - p}{r}$$

Proposition F.8 from Yang et al. (2020) show that anywhere in $\mathcal{X}$, we have $\frac{\mathrm{d}}{\mathrm{d}t}[g(\boldsymbol{z})] \leq \Omega(g(\boldsymbol{z}))$. This means that one can upper bound the growth of $g(\boldsymbol{x})$ while shifted by $\boldsymbol{\delta}$ by integrating over $\Omega(g(\boldsymbol{z}))$ instead. For easier notation let $h(t) = g(\boldsymbol{x} + t \cdot \boldsymbol{\delta}')$ which implies $h'(t) = \frac{\mathrm{d}}{\mathrm{d}t}h(t) \leq \Omega(h(t))$. Notably, the function $\Omega(p)$ is always non-negative. See the equivalent Definition H.12 from Yang et al. (2020) where the function is denoted as $\Phi(p)$ and it is written as an expectation of a maximum over a value that is always non-negative. For positive (non zero) $\Omega(h(t))$, including but not limited to $h(t) \leq 1/2$ (see the definition of the function in Appendix F from Yang et al. (2020)) It follows:

$$\frac{h'(t)}{\Omega(h(t))} \leq 1 \Rightarrow \int_0^r \frac{h'(t)}{\Omega(h(t))}\mathrm{d}t \leq \int_0^r 1\mathrm{d}t = r$$

We set $u = h(t) \Rightarrow \mathrm{d}u = h'(t)\mathrm{d}t$ which implies

$$\Pi(u) = \int_0^r \frac{h'(t)}{\Omega(h(t))}\mathrm{d}t = \int_{u=h(0)}^{u=h(r)} \frac{1}{\Omega(u)}\mathrm{d}u \leq r$$

Here $h(0) = g(\boldsymbol{x} + 0 \cdot \boldsymbol{\delta}') = g(\boldsymbol{x}) = \beta$, and $h(0) = g(\boldsymbol{x} + r \cdot \boldsymbol{\delta}') = g(\tilde{\boldsymbol{x}}) = \bar{\beta}$. With the reference function $F(\gamma) = \int_\gamma^{1/2} \frac{1}{\Omega(p)}\mathrm{d}p$ we have

$$r \geq \int_\beta^{\bar{\beta}} \frac{1}{\Omega(p)}\mathrm{d}p = \int_\beta^{1/2} \frac{1}{\Omega(p)}\mathrm{d}p - \int_{\bar{\beta}}^{1/2} \frac{1}{\Omega(p)}\mathrm{d}p = F(\beta) - F(\bar{\beta}) \Rightarrow F(\beta) - F(\bar{\beta}) \leq r$$

which is $F(\bar{\beta}) \geq F(\beta) - r$.

The authors already computed $\Omega(p)$ (in their paper it is called as Φ) for following several distributions, including:

- Isotropic Gaussian smoothing against ℓ_2 ball ($\sigma = 1$): $\Omega(u) = \Phi'(\Phi^{-1}(1-u))$ which implies:

$$\Pi_{\text{Gaussian}}(u) = \int_\beta^{\bar{\beta}} \frac{1}{\Omega(u)}\mathrm{d}u = \int_\beta^{\bar{\beta}} \frac{1}{\Phi'(\Phi^{-1}(1-u))}\mathrm{d}u$$

With $c = \Phi^{-1}(1-u)$ we have $\mathrm{d}p = -\Phi'(c)\mathrm{d}c$, and $\Phi'(c) = \Phi'(\Phi^{-1}(1-u))$. Therefore

$$\int_{\beta}^{\overline{\beta}} \frac{1}{\Phi'(\Phi^{-1}(1-u))}\mathrm{d}u = \int_{\Phi^{-1}(1-\beta)}^{\Phi^{-1}(1-\overline{\beta})} -\mathrm{d}u = \Phi^{-1}(1-\beta) - \Phi^{-1}(1-\overline{\beta}) \leq r$$
$$\Rightarrow \Phi^{-1}(1-\overline{\beta}) \geq \Phi^{-1}(1-\beta) - r \Rightarrow 1 - \Phi^{-1}(\overline{\beta}) \geq 1 - \Phi^{-1}(\beta) - r$$
$$\Rightarrow \Phi^{-1}(\overline{\beta}) \leq \Phi^{-1}(\beta) + r \rightarrow \overline{\beta} \leq \Phi(\Phi^{-1}(\beta) + r)$$

Which completely aligns with the aforementioned closed form ℓ_2 certificate.

- Laplace smoothing against ℓ_1 ball ($\sigma = \sqrt{2}\lambda$): $\Omega(u) = \frac{u}{\lambda}$ which implies:

$$\Pi_{\text{Laplace}}(u) = \int_{\beta}^{\overline{\beta}} \frac{1}{\Omega(u)}\mathrm{d}u = \int_{\beta}^{\overline{\beta}} \lambda\frac{1}{u}\mathrm{d}u = \lambda \log\frac{\overline{\beta}}{\beta} \leq r$$
$$\Rightarrow \frac{\overline{\beta}}{\beta} \leq 2^{r/\lambda} \Rightarrow \overline{\beta} \leq 2^{r/\lambda} \cdot \beta$$

Note that in both cases, the function $\Omega(p)$ is positive in $(0, 1)$.

E.3 Supplementary Experiments

Compute resources. We ran our experiment using Nividia A-100 and H-100 Tensor Core GPUs. For each experiment only one GPU was used. We use the A-100 GPU for the CIFAR-10 dataset under ResNet setup, and the conformal risk control experiment. The rest of the results use H-100 as the compute resource.

Experimental setup. For the CIFAR-10 datasets we evaluate the results over 2048 test samples for ResNet model and 10000 images for the ViT models. For the ImageNet since the number of classes are 1000, we report our results over 5000 images for ViT models and 50000 images on ResNet models. Ultimately the number of samples does not influence the empirical results. The number of Monte Carlo samples are initially set to 500 for CIFAR and 300 for ImageNet. For each experiment, and for the reported sample rate we cut the precomputed samples, from the reported number.

Our results are reports over 100 runs (except the conformal risk control which is over one run. In each run we sample the 10% of the points as the calibration set. For conformal risk control we report the result on 300 images where 100 random images from it is taken for the calibration. Ultimately the size of the calibration set does not effect the final performance. As the calibration set gets larger the distribution of the coverage probability concentrates around $1 - \alpha$.

The time to compute the logits for the CIFAR-10 dataset is 1:30:56 (ViT with 10000 datapoints and 500 samples), and for the ImageNet dataset it is 13:52:11 (ViT with 5000 datapoints and 300 MC samples). For the ImageNet, and the ResNet model this number is 2:52:11 (for 50000 datapoints and 1000 samples).

Set size experiment. Tables Table E.2 and Table E.3 report the empirical coverage and average prediction set size of RCP1 for different radii r on the CIFAR-10 dataset using a ResNet model under two noise levels, $\sigma = 0.25$ and $\sigma = 0.5$, respectively. We also report the result of the ImageNet dataset (for the ResNet model) in Table E.4. Specifically for this dataset, because of the large number of classes we also reported the

proportion of the sets below specific sizes (1, 3, 5, and 10). As expected, increasing the radius r results in a more conservative setup and hence higher coverage on the clean points. For CIFAR-10 dataset Fig. E.2, and for the ImageNet dataset Fig. E.3 visualize the comparative performance between BinCP and RCP1 across various radii and sampling budgets. These results are on the ResNet model. The heatmaps show the difference in set sizes $|C_{r,\text{BinCP}}| - |C_{r,\text{RCP1}}|$, where positive (green) values indicate that RCP1 provides smaller or more efficient sets. RCP1 generally outperforms BinCP across low sample rates, especially for smaller radii and moderate sampling budgets. The reference tables (Tables Table E.2 and Table E.3, Table E.4) can be used to interpret these differences in absolute terms.

Proportion of small sets. As also discussed in § 7.4 although the proportion of sets with size less than a threshold shows how applicable a CP algorithm is, it can be misleading – a CP framework can return many false prediction sets with very small set size. Therefore alongside the proportion of these sets we should also report their coverage. We show these results in Fig. E.4. Our observation is that all setups result in sets with coverage higher than the determined level. Note that in terms of proportion RCP1 stands somewhere between BinCP with 64 and 128 samples which aligns with our aforementioned intuition.

Regression Experiment For robust conformal regression with RCP1 we use the Udacity and originates from Nvidia's DAVE-2 system(Bojarski et al., 2016). The input of this task is an scene, and the task is to estimate the steering angle of the car. The output range is from -1 (completely steering right) to 1 (left). For this task we finetune a ResNet18 model (He et al., 2016) on images augmented with isotopic Gaussian noise with $\sigma = 0.5$. We run finetuning for 200 epochs. We use the same σ for augmenting the input in RCP1. We set $1 - \alpha = 0.9$, and evalute on $r \in \{0.12, 0.25, 0.5\}$. To the best of our knowledge our result is the first robust conformal prediction with randomized smoothing for regression task. Table E.5 compares the interval length and empirical coverage across various radii.

Table E.2: Empirical coverage and average set size for different radii (r), for CIFAR-10 dataset with ResNet model and $\sigma = 0.25$.

r	Coverage	Avg Set Size
0.06	0.936 ± 0.018	2.156 ± 0.241
0.12	0.961 ± 0.014	2.646 ± 0.306
0.18	0.981 ± 0.010	3.315 ± 0.478
0.25	0.990 ± 0.008	4.178 ± 0.798
0.37	1.000 ± 0.000	10.000 ± 0.000
0.50	1.000 ± 0.000	10.000 ± 0.000
0.75	1.000 ± 0.000	10.000 ± 0.000

Table E.3: Empirical coverage and average set size for different radii (r), for CIFAR-10 dataset with ResNet model and $\sigma = 0.5$.

r	Coverage	Avg Set Size
0.06	0.921 ± 0.020	2.684 ± 0.244
0.12	0.937 ± 0.018	2.937 ± 0.285
0.18	0.951 ± 0.016	3.236 ± 0.356
0.25	0.966 ± 0.014	3.741 ± 0.530
0.37	0.980 ± 0.010	4.500 ± 0.560
0.50	0.990 ± 0.007	5.300 ± 0.712
0.75	1.000 ± 0.000	10.000 ± 0.000

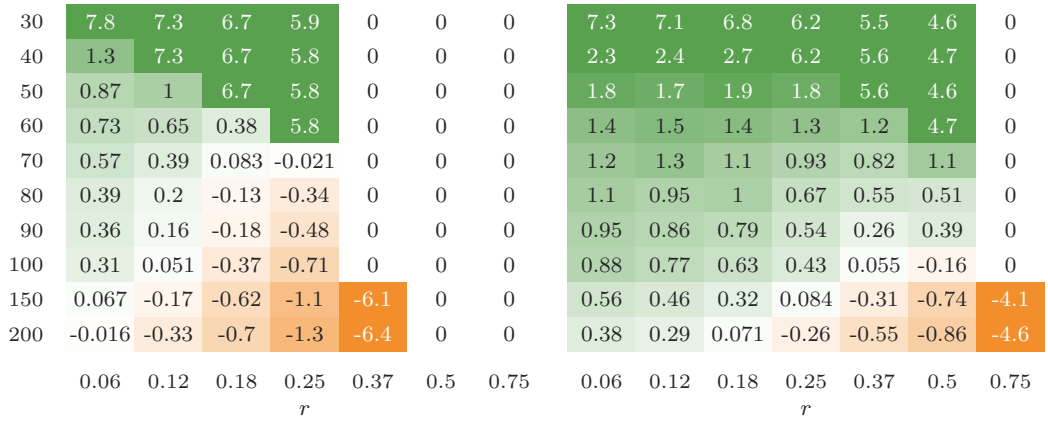

Figure E.2: Comparison of BinCP and RCP1 in terms of $|C_{r,\text{BinCP}}| - |C_{r,\text{RCP1}}|$ (higher (green) shows better performance for RCP1) across various radii and sample rates. Results are on the ResNet model and for the CIFAR-10 dataset. [Left] $\sigma = 0.25$, and [Right] $\sigma = 0.5$. Note that the numbers are in terms of difference and to compute the absolute number Table E.2, and Table E.3 can be used as reference.

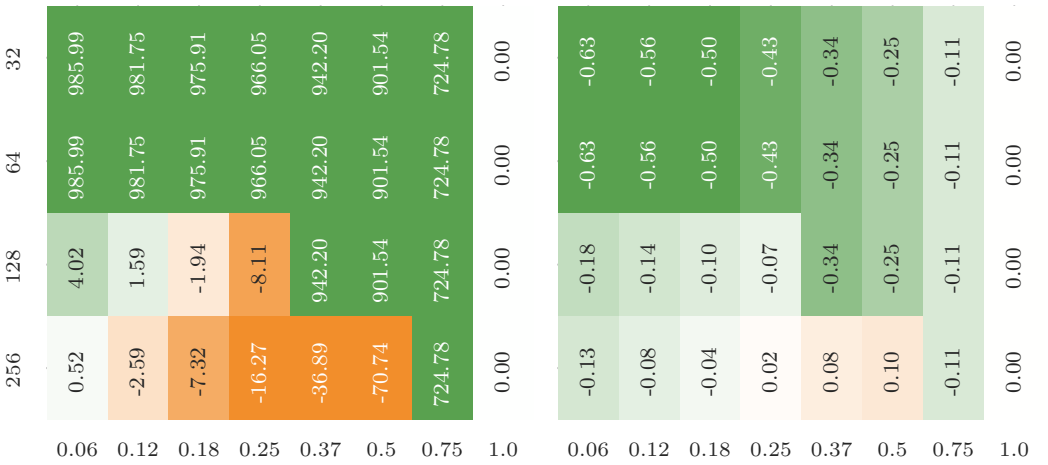

Figure E.3: [Left] Comparison of the average set size $|C_{r,\text{BinCP}}| - |C_{r,\text{RCP1}}|$ and [Right] the proportion of the sets with size ≤ 10 expressed in BinCP - RCP1. In both plots green shows that RCP1 is performing better. To convert the relative difference to absolute number Table E.4 can be used as the reference.

Table E.4: Statistics from RCP1 across various radii. The results are for ImageNet dataset and the ResNet model.

r	**Avg Set Size**	**Emp. Coverage**	$C \leq 1$	$C \leq 3$	$C \leq 5$	$C \leq 10$
0.06	14.013 ± 1.787	0.921 ± 0.008	0.139 ± 0.010	0.311 ± 0.018	0.430 ± 0.025	0.626 ± 0.035
0.12	18.246 ± 2.606	0.936 ± 0.009	0.120 ± 0.010	0.274 ± 0.019	0.383 ± 0.024	0.560 ± 0.032
0.18	24.095 ± 3.645	0.951 ± 0.008	0.101 ± 0.010	0.239 ± 0.018	0.336 ± 0.023	0.501 ± 0.031
0.25	33.953 ± 5.744	0.964 ± 0.007	0.082 ± 0.008	0.201 ± 0.017	0.288 ± 0.022	0.432 ± 0.033
0.37	57.802 ± 10.753	0.979 ± 0.005	0.058 ± 0.008	0.151 ± 0.017	0.219 ± 0.022	0.337 ± 0.031
0.50	98.464 ± 19.136	0.989 ± 0.003	0.036 ± 0.006	0.104 ± 0.016	0.160 ± 0.021	0.252 ± 0.029
0.75	275.222 ± 88.968	0.998 ± 0.002	0.012 ± 0.006	0.036 ± 0.016	0.063 ± 0.025	0.115 ± 0.039
1.00	1000.000 ± 0.000	1.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000

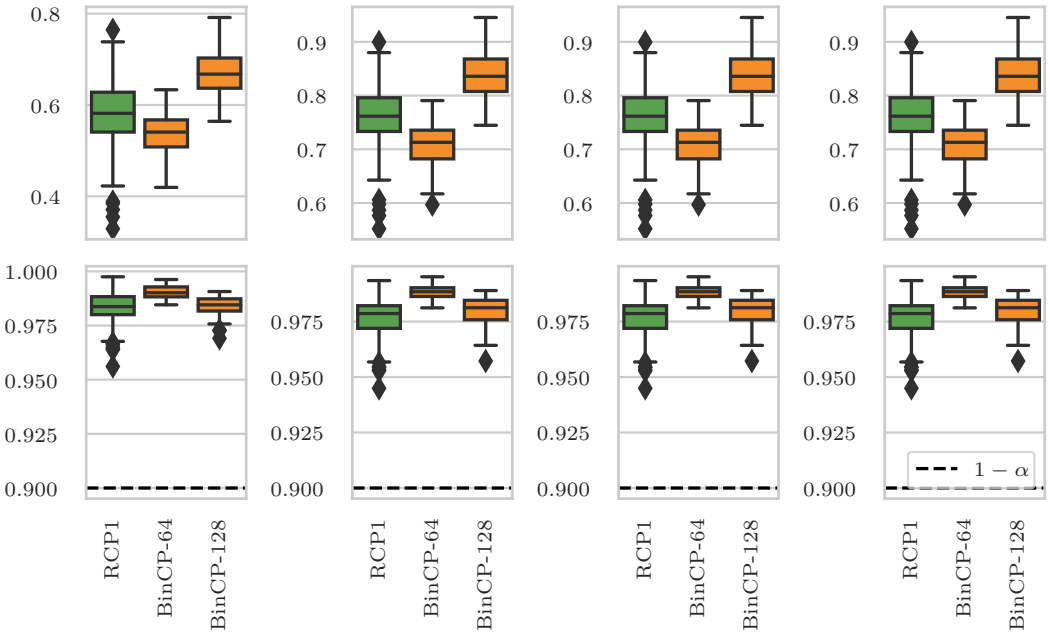

Figure E.4: [Up] The proportion and [Bottom] the coverage of the prediction sets with size [From left to right] $|C| \leq 1$, $|C| \leq 3$, $|C| \leq 5$, $|C| \leq 10$.

Table E.5: RCP1 for conformal regression. We report the empirical coverage and the interval length across varius radii.

r	**Empirical Coverage**	**Interval Width**
0.00	0.900 ± 0.005	0.371 ± 0.012
0.12	0.920 ± 0.005	0.426 ± 0.014
0.25	0.938 ± 0.004	0.494 ± 0.018
0.50	0.963 ± 0.003	0.667 ± 0.026

F Supplementary Material: Front-Loaded Robust Conformal Prediction: Heavy Calibration, Minimal Test-Time Cost

F.1 Algorithm and time complexity

Here we provide the algorithm for Flor$^{(1)}$. The prediction set is defined in the same way as RCP1: $C_{\mathcal{B}}(\boldsymbol{x}_{n+1}) := \{y \in \mathcal{Y} : s(\boldsymbol{x}_{n+1}+\boldsymbol{\epsilon}, y) \geq 1-\lambda\}$. Notably for an easier notation, in Algorithm 7 we use λ instead of $1-\lambda$. Additionally, while the theory is expressed in risk which is set to the miscoverage probability, here we define everything with the coverage probability. Notably this decision only changes few steps and does not affect the validity of conformal risk control; these changes are only for better readability, and by a change in sign the setup is equivalent as proposed in Proposition 8.2. We define the algorithm in each block separately, and finally Algorithm Algorithm 7 combines them all together. Usually with a small δ_0 the "while" loop in Algorithm Algorithm 7 works only once, and we do not need to resample again.

Tuning to find λ (Algorithm Algorithm 8) In this step, we first find a candidate λ that potentially results in robust $1-\alpha$ coverage guarantee. Since the validation evaluates this quantity using a finite Monte Carlo budget, we simulate the behavior of a substantially larger sampling budget – we estimate each β with m_{tune} samples and pretend this number is estimated with m_{cert} samples. This mismatch does not affect the validity of the conformal risk control guarantee, since the certification phase is later performed using an independent set of m_{cert} and confidence intervals here are just a simulation to prevent failure in the next step.

Validation (Algorithm Algorithm 9) Given a candidate λ, this step ensures that under the worst case noise this threshold still results in $1-\alpha$ coverage probability. Notably this step is similar to one iteration of the binary search, only this time with the actual m_{cert} samples.

Computational complexity. Table F.1 shows the computational complexity of each method in both calibration and test phases. Notably since calibration is a pre-processing step, and the calibration set is small, high computational cost at this stage can be considered negligible. In almost all setups, the test set is assumed to have unlimited size, and in real-life scenarios, test-time requires a quick response where Flor$^{(1)}$ and RCP1 align with that need. As shown in Fig. 8.2, as the number of test points increases, the computational costs of Flor$^{(1)}$ and RCP1 eventually converge. Notably, this evaluation is conducted over 10^8 test points, while in reality this number can be significantly higher.

F.2 Related Works

Conformal prediction, introduced by Vovk et al. (2005), is a framework for constructing prediction sets that contain the true outcome with a user-specified probability of $1-\alpha$. The method relies on a score function $s : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$, which measures the compatibility between inputs and candidate outputs, together with an exchangeable set of held-out calibration samples. In § 8.2 we discuss the calibration and inference procedure, and a comprehensive tutorial can also be found in (Angelopoulos et al., 2024). Robust conformal prediction - as originally introduced by Gendler et al. (2021) - aims to extend the coverage guarantee the inputs from the same distribution perturbed

Algorithm 7 Flor$^{(1)}$

Input: Calibration set $\mathcal{D}_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^{n}$, score function $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, smoothing distribution ψ, threat model $\mathcal{B}$, nominal miscoverage α, confidence parameter δ, total MC budget $m = m_{\text{tune}} + m_{\text{cert}}$, offset δ_0
Output: Robust prediction set mapping $\tilde{x}_{n+1} \mapsto C_{\mathcal{B}}(\tilde{x}_{n+1})$ with expected coverage $1 - \alpha$
$\lambda \leftarrow +\infty$
while Validation($\mathcal{D}_{\text{cal}}, s, \psi, \mathcal{B}, \alpha, \delta, m_{\text{cert}}, \lambda$) from Algorithm 9 returns False **do**
  $\lambda \leftarrow$ Tuning($\mathcal{D}_{\text{cal}}, s, \psi, \mathcal{B}, \alpha, \delta, m_{\text{tune}}, m_{\text{cert}}, \delta_0$) from Algorithm 8
end while
For a new input $\tilde{x}_{n+1}$, draw $\epsilon \sim \psi$
Set $C_{\mathcal{B}}(\tilde{x}_{n+1}) := \{y \in \mathcal{Y} : s(\tilde{x}_{n+1} + \epsilon, y) \geq 1 - \lambda\}$
Return $C_{\mathcal{B}}(\cdot)$

Table F.1: Computational complexity of smoothing-based RCPs.

Method	Calibration	Test (per point)
BinCP	$O(\lvert\mathcal{D}_{\text{cal}}\rvert \cdot n_{\text{MC}})$	$O(n_{\text{MC}})$
RCP1	$O(1)$	$O(1)$
Flor$^{(1)}$	$O(\lvert\mathcal{D}_{\text{cal}}\rvert \cdot n_{\text{MC}})$	$O(1)$

with worst case noise. Besides the worst case (a.k.a. adversarial) robustness, there are orthogonal robustness frameworks proposed for CP: robustness to average noise or average adversarial examples, known as probabilistically-robust CP (Ghosh et al., 2023), and robustness to covariate Tibshirani et al. (2019b), and general distribution shifts Barber et al. (2022). For adversarial robustness, a group of works are using Lipschitz boundedness of the networks or verifier (Jeary et al., 2024; Massena et al., 2025). These methods (i) require white-box access to the model, and (ii) they provide robustness with useful sets up to a very small radius (magnitude of perturbation). Alternatively, randomized smoothing provides black-box robustness to a very larger radii and it can apply to any model. Originally smoothing-based robustness was introduced to CP by Gendler et al. (2021), proposing to use $\hat{s}(x, y) = \mathbb{E}_{\epsilon}[s(x + \epsilon, y)]$ as a new score. Here the noise ϵ comes from a standard distribution, and the smooth score $\hat{s}$ changes slowly around the input. Therefore, the mean score at the clean input can be bounded by the score at the received (potentially perturbed) input. While computing $\hat{s}$ is in general intractable, we can estimate it through Monte-Carlo (MC) sampling, followed by finite sample correction. The original work did not account for finite sample error proposing a guarantee that would be only asymptotically valid. A follow-up by Yan et al. (2024) solves the issue by applying finite sample correction. Further Zargarbashi et al. (2024) proposed to use the tighter CDF-base bound and restated calibration-time robustness to derive smaller prediction sets with the same guarantee. Furthermore the authors propose robustness to label, and feature poisoning which is out of the scope of this work. All three methods are using confidence certificates (since the score function is continuous) which requires more MC samples compared to binary certificates. Zargarbashi and Bojchevski (2025) proposes a robust CP through a quantile of quantiles method (called BinCP), and show that the robustness can be attained using a single binary certificate. This allows them to reduce the needed sample-rate by an order of magnitude. By the time, BinCP is the state of the art sample-heavy robust CP.

In practice, robust CP methods must support fast inference, yet existing approaches

Algorithm 8 Flor$^{(1)}$: Pre-computation (tuning samples)

Input: Calibration set $\mathcal{D}_{\text{cal}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, score function $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, smoothing distribution ψ, threat model $\mathcal{B}$, nominal miscoverage α, confidence parameter δ, sample budgets $m_{\text{tune}}, m_{\text{cert}}$, offset δ_0
Output: Candidate threshold λ
Sample $S_{i,j} \leftarrow s(\boldsymbol{x}_i + \boldsymbol{\epsilon}_{i,j}, y_i)$ for all $(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}$, $j \in [m_{\text{tune}}]$, and $\boldsymbol{\epsilon}_{i,j} \sim \psi$
$\lambda_{\min} \leftarrow \min_{i,j} S_{i,j}$, $\lambda_{\max} \leftarrow \max_{i,j} S_{i,j}$
while $\lambda_{\max} - \lambda_{\min} >$ tol **do**
  $\lambda' \leftarrow (\lambda_{\min} + \lambda_{\max})/2$
  for each $(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}$ **do**
    $\beta_i(\lambda') \leftarrow \frac{1}{m_{\text{tune}}} \sum_{j=1}^{m_{\text{tune}}} \mathbb{I}[s(\boldsymbol{x}_i + \boldsymbol{\epsilon}_{i,j}, y_i) \geq \lambda']$
    $\underline{\beta}_i \leftarrow \text{ClPr}(m_{\text{cert}}\beta_i(\lambda'), m_{\text{cert}}, 1 - \delta/|\mathcal{D}_{\text{cal}}|)$
    $\tilde{\beta}_i \leftarrow \text{c}^{\downarrow}[\underline{\beta}_i, \mathcal{B}]$
  end for
  $R(\lambda') \leftarrow \frac{1}{n+1} \sum_{i=1}^{n} \tilde{\beta}_i$
  if $R(\lambda') \geq 1 - \alpha + \delta + \delta_0$ **then**
    $\lambda_{\max} \leftarrow \lambda'$
  else
    $\lambda_{\min} \leftarrow \lambda'$
  end if
end while
$\lambda \leftarrow \lambda_{\min}$
Return λ

incur substantial computational overhead at test time. To address this limitation, Zargarbashi et al. (2025) provides smoothing-based robustness that works witha single forward pass on a noise-augmented input, thereby eliminating the need for Monte Carlo estimation during inference. The resulting work is a RCP, that is equal to BinCP with modest sampling budget (e.g. in their image classification between 70 to 150 samples). While working with the same running time, RCP1 outperforms the verification-based robustness. Clearly by increasing the sampling budget in BinCP, RCP1 falls behind in usability.

F.3 Supplementary to Theory

F.3.1 Proof

Proposition 8.2. *Given clean calibration points $Z_1, \ldots, Z_n$ ($Z_i = (X_i, Y_i)$), and a potentially perturbed $\tilde{Z}_{n+1} = (\tilde{X}_{n+1}, Y_{n+1})$, for $\tilde{X}_{n+1} \in \mathcal{B}(X_{n+1})$, let $E_i : i \in [n+1] \sim \psi$ from a predefined smoothing scheme ψ, we have*

$$\Pr_{E_{n+1}, \mathcal{D}_{n+1}} [Y_{n+1} \in C(\tilde{X}_{n+1})] \geq 1 - \alpha$$

For $C(\tilde{X}_{n+1}) = \{y : s(\tilde{X}_{n+1} + E_{n+1}, y) \geq 1 - \lambda\}$ where $1 - \lambda$ is defined as following: Let $\beta_i(\lambda) = \Pr_{E_i}[s(X_i + E_i, Y_i) \geq -\lambda]$ and $\underline{\beta_i} \leq \beta_i$ with $1 - \delta/|\mathcal{D}_{\text{cal}}|$ probability. Then

$$\lambda = 1 - \sup\{\lambda' : \frac{1}{n+1} \sum_{i=1}^{n} \text{c}^{\downarrow}(\underline{\beta}_i(\lambda'), \mathcal{B}) \geq 1 - \alpha + \delta\}$$

Proof for Proposition 8.2. We prove through conformal risk control: As discussed, define

Algorithm 9 Flor$^{(1)}$: Validating λ

Input: Calibration set $\mathcal{D}_{\text{cal}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, score function $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, smoothing distribution ψ, threat model $\mathcal{B}$, nominal miscoverage α, confidence parameter δ, sample budget m_{cert}, candidate λ
Output: Whether λ attains robust $1-\alpha$ coverage
Sample $S_{i,j}^{\text{cert}} \leftarrow s(\boldsymbol{x}_i + \boldsymbol{\epsilon}_{i,j}, y_i)$ for all $(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}$, $j \in [m_{\text{cert}}]$, and $\boldsymbol{\epsilon}_{i,j} \sim \psi$
for each $(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}$ **do**
  $\beta_i(\lambda) \leftarrow \frac{1}{m_{\text{cert}}} \sum_{j=1}^{m_{\text{cert}}} \mathbb{I}[S_{i,j}^{\text{cert}} \geq \lambda]$
  $\underline{\beta}_i \leftarrow \text{ClPr}(m_{\text{cert}}\beta_i(\lambda), m_{\text{cert}}, 1 - \delta/|\mathcal{D}_{\text{cal}}|)$
  $\tilde{\beta}_i \leftarrow \text{c}^{\downarrow}[\underline{\beta}_i, \mathcal{B}]$
end for
$R(\lambda) \leftarrow \frac{1}{n+1} \sum_{i=1}^{n} \tilde{\beta}_i$
if $R(\lambda) \geq 1 - \alpha + \delta$ **then**
  Return Yes
else
  Return No
end if

the risk as the miscoverage probability; i.e.

$$L(X_i; \lambda) = \max_{\tilde{X}_i \in \mathcal{B}(X_i)} Pr_{E_i}[s(\tilde{X}_i + E_i, Y_i) < -\lambda]$$

which by definition is equivalent to $1 - \tilde{\beta}_i(\lambda)$. This risk upper bounded by 1, continuous, and decreasing w.r.t. λ as by increasing it, (as $-\lambda$ decreases) the probability of miscoverage decreases. It shows that the risk function complies with all conditions to be used in risk control framework.

Notably replacing $L_i(\lambda)$ with any upper bound can only increase λ, making the final expected risk even lower. From Eq. 70

$$\begin{aligned} \beta_i(\lambda) = Pr_{E_i}[s(X_i + E_i, Y_i) < -\lambda] \quad \text{implies} \\ \text{c}^{\downarrow}(\beta_i(\lambda), \mathcal{B}) \leq \min_{\tilde{X}_i \in \mathcal{B}(X_i)} \Pr_{E_i}[s(\tilde{X}_i + E_i, Y_i) < -\lambda] \end{aligned}$$

Meaning that $L(X_i; \lambda) \leq 1 - \text{c}^{\downarrow}(\beta_i(q), \mathcal{B})$ is a valid replacement. Through conformal risk control we have

$$\begin{aligned} \mathbb{E}[L(X_{n+1}, \lambda^*)] \leq \alpha - \delta \quad \text{for} \\ \lambda^* = \inf\{\lambda : \frac{1}{n+1}\Big(\sum_{i=1}^{n} L(X_i; \lambda) + 1\Big) \leq \alpha - \delta\} \end{aligned}$$

Replacing the risk with a valid upper bound we have

$$\begin{aligned} \lambda_{\text{alt}} \geq \lambda \quad \text{for} = \lambda_{\text{alt}} \\ \inf\{\lambda : \frac{1}{n+1}\Big(\sum_{i=1}^{n} 1 - \text{c}^{\downarrow}[\beta(-\lambda), \mathcal{B}] + 1\Big) \leq \alpha - \delta\} \\ = -\sup\{q : \frac{1}{n+1} \sum_{i=1}^{n} \text{c}^{\downarrow}[\beta(q), \mathcal{B}] \geq 1 - \alpha + \delta\} \end{aligned}$$

For the test point we have

$$\mathbb{E}[L(X_{n+1}, \lambda^{\text{alt}})] \leq \mathbb{E}[L(X_{n+1}, \lambda^*)] \leq \alpha - \delta \quad \text{i.e.}$$
$$\Pr\left[s(\tilde{X}_{n+1} + E_i, Y_{n+1} | \tilde{X}_{n+1} \in \mathcal{B}(X_{n+1})\right] \geq 1 - \alpha + \delta$$

Estimating $\beta_i(\lambda)$ through MC sampling and computing Clopper-Pearson intervals with $1 - \delta/|\mathcal{D}_{\text{cal}}|$ confidence result in δ failure probability of the bound through union bound. The failure probability of CP itself is $\alpha - \delta$, hence the failure of the whole setup is bounded by $(\alpha - \delta) + \delta = \alpha$. □

F.4 Supplementary to Experiments

Experimental setup. We follow Zargarbashi et al. (2025) in evaluation pipeline: For classification task we evaluate our approach on CIFAR-10, and ImageNet datasets. For CIFAR-10, we evaluate performance over 2,048 test samples, and for the ResNet and 10,000 images for the ViT models. For ImageNet, we report results for 5,000 images for ViT models and 50,000 images for ResNet models. The total number of evaluated samples does not affect the empirical conclusions. The subsets are taken exchangeably from the actual dataset. Unless specified otherwise, Flor$^{(1)}$ uses 10,000 Monte Carlo samples during calibration, and a single MC sample during test. Same sample rate is used for BinCP (the ideal setup) both at calibration sample and test. The size of the calibration set is random between 100 to 250 and sampled exchangeably. The only effect calibration set size is on the concentration of the empirical result around the expectation – the coverage comes from a Beta distribution $\Pr[S_{n+1} \geq q \mid \mathcal{D}_{\text{cal}}] \sim \text{Beta}((1-\alpha)\cdot(n+1), \alpha(n+1)))$ and it concentrates around the expected coverage with a rate increasing by the calibration set size. The initial number of Monte Carlo samples is set to 10,000 both datasets; we subsample from these precomputed runs for other sample rates. There is one exception in Fig. 8.6-right where we increase the sample rate up to 60,000 for the CIFAR-10 dataset. Our results are averaged over 100 runs. In each run, 10% of the points are randomly selected as the calibration set.

Computationally demanding setup. For CIFAR-10, we pair a 50M-parameter diffusion model from Dhariwal and Nichol (2021) with a `ViT-B/16` model from Dosovitskiy et al. (2020), pretrained on ImageNet at 224×224 resolution and finetuned on CIFAR-10, achieving 97.9% accuracy in the HuggingFace implementation. For ImageNet, we employ a 552M-parameter class-unconditional diffusion model followed by a BEiT-L model (305M parameters) from Bao et al. (2021), which attains 88.6% top-1 validation accuracy. Implementations are taken from the `timm` library (Wightman, 2019).

Supplementary reports. We report the numerical value of the result for ImageNet dataset in Table F.2 (ResNet model) and Table F.3 (diffusion + ViT model). In the same order we report the results for CIFAR-10 dataset in Table F.4, and Table F.5.

Flor$^{(k)}$. Recall that Flor$^{(k)}$ explicitly calibrates the to account for the downstream majority-vote aggregation, by enforcing a margin-above-half constraint on the smoothed coverage probabilities. This effectively shifts calibration from controlling single-sample coverage (which calibrates with the objective over expected β_{n+1}) to directly enabling majority vote success (ensuring that in $1 - \alpha$ cases β_{n+1} remains above the margin). Fig. F.1 empirically examines how this mechanism interacts with the aggregation rate k

Table F.2: Average prediction set size (mean ± std) as a function of perturbation radius r. The results are for the ImageNet dataset, ResNet model

r	BinCP	RCP1	Flor$^{(1)}$
0.00	10.770 ± 7.584	8.605 ± 5.110	12.451 ± 8.123
0.06	11.243 ± 7.912	10.396 ± 6.896	13.128 ± 8.597
0.12	11.732 ± 8.240	19.595 ± 14.091	13.786 ± 8.960
0.18	12.265 ± 8.592	23.810 ± 18.245	14.673 ± 9.195
0.25	12.911 ± 9.041	45.019 ± 28.125	15.905 ± 9.931
0.37	14.133 ± 9.869	69.886 ± 50.844	18.693 ± 12.025
0.50	15.570 ± 10.795	144.226 ± 105.848	24.076 ± 16.621

Table F.3: Average prediction set size (mean ± std) as a function of perturbation radius r. The results are for the ImageNet dataset, Diffusion + ViT model

r	BinCP	RCP1	Flor$^{(1)}$
0.00	3.658 ± 0.624	3.989 ± 0.990	6.084 ± 1.912
0.06	3.877 ± 0.685	6.464 ± 2.093	6.735 ± 2.198
0.12	4.088 ± 0.738	8.414 ± 2.390	7.507 ± 2.535
0.18	4.345 ± 0.802	10.347 ± 2.341	8.401 ± 2.833
0.25	4.672 ± 0.899	26.859 ± 15.663	9.802 ± 3.335
0.37	5.383 ± 1.099	79.228 ± 108.014	13.367 ± 4.348
0.50	6.304 ± 1.361	197.633 ± 150.188	20.765 ± 7.514

Table F.4: Average prediction set size (mean ± std) as a function of perturbation radius r. Results for CIFAR-10 dataset and ResNet model.

r	BinCP	RCP1	Flor$^{(1)}$
0.00	2.143 ± 0.187	2.470 ± 0.257	2.680 ± 0.165
0.06	2.238 ± 0.212	2.717 ± 0.269	2.791 ± 0.198
0.12	2.336 ± 0.201	2.912 ± 0.301	2.921 ± 0.180
0.18	2.475 ± 0.185	3.228 ± 0.328	3.097 ± 0.162
0.25	2.642 ± 0.207	3.743 ± 0.578	3.291 ± 0.176
0.37	2.889 ± 0.207	4.537 ± 0.583	3.647 ± 0.192
0.50	3.266 ± 0.220	5.413 ± 0.754	4.165 ± 0.214

Table F.5: Average prediction set size (mean ± std) as a function of perturbation radius r. Results for CIFAR-10 dataset and Diffusion + ViT model.

r	BinCP	RCP1	Flor$^{(1)}$
0.00	2.335 ± 0.652	3.166 ± 0.483	3.319 ± 0.448
0.06	2.507 ± 0.690	3.877 ± 0.314	3.683 ± 0.495
0.12	2.716 ± 0.750	4.284 ± 0.245	4.054 ± 0.544
0.18	2.939 ± 0.789	4.657 ± 0.267	4.471 ± 0.598
0.25	3.193 ± 0.828	6.856 ± 1.758	4.944 ± 0.598
0.37	3.667 ± 0.861	8.327 ± 1.060	5.729 ± 0.532
0.50	4.223 ± 0.911	8.935 ± 0.825	6.453 ± 0.556

and the offset parameter α_0 across three representative robustness radii. Formally for a fixed value of α_0, and k, the value β_0 has a closed of:

$$\beta_0 = \inf\Big\{ p \in [1/2, 1] \ : \ (1 - \alpha + \alpha_0) \cdot \Big(1 - \Phi_{\text{bin}}\Big(\tfrac{k-1}{2}, k, \tfrac{1}{2} + p\Big)\Big) \ > \ 1 - \alpha \Big\}.$$

Picking any value above this closed form only increases the set size.

The top row reports relative improvements over the single-sample baseline Flor$^{(1)}$. We observe a broad region in which Flor$^{(k)}$ consistently yields smaller prediction sets, with the largest gains attained for moderate test-time sample rates and small but nonzero offsets α_0. This behavior reflects the intended role of α_0: introducing a mild safety margin over $1 - \alpha$ is necessary as the majority vote has a probability which by definition is lower than one. Increasing α_0 clearly decreases the needed β_0 but enforces larger set size through calibrating over a higher coverage. What we observe is that it is better not to increase α_0 more than a very small safety margin. As k increases, the improvement region becomes more structured and gradually saturates, indicating diminishing returns once the variance of the majority-vote estimator is sufficiently reduced.

Fig. F.2 visualizes stochasticity, in form of *input-conditional* set size probability. To compute this probability, we first sample the calibration set, and calibrate each method (notably for BinCP, Flor$^{(1)}$, and Flor$^{(k)}$ we use 10^4 samples per each point), then for 10 different runs, we run each method with its predefined test-time sample rate, which applies to BinCP, and Flor$^{(k)}$, as for RCP1, and Flor$^{(1)}$ the test-time sample rate is always 1. We compute the probability the frequency of the discrete events (set size values). Although this is a MC-sampling estimation with very low sample, we do not need the probabilities to be precise as we only want to compare the stochasticity of each method. Each column corresponds to a fixed test input $\boldsymbol{x}_{n+1}$, and the intensity profile along the y-axis depicts the distribution of $|C_r(\boldsymbol{x}_{n+1})|$ induced solely by the sampling randomness (with calibration performed using 10^4 samples). As expected, both RCP1 and Flor$^{(1)}$, stochasticity in the prediction sets while BinCP remains more determinisitic for all sample rates. Interestingly in all empirical evaluations, Flor$^{(k)}$ is more deterministic compared to BinCP. Notably by increasing the test-time sample rate (from SR= 21 to SR= 101) progressively BinCP and Flor$^{(k)}$ sharpens their conditional distribution and reduces both the frequency of large outliers and the column-wise dispersion.

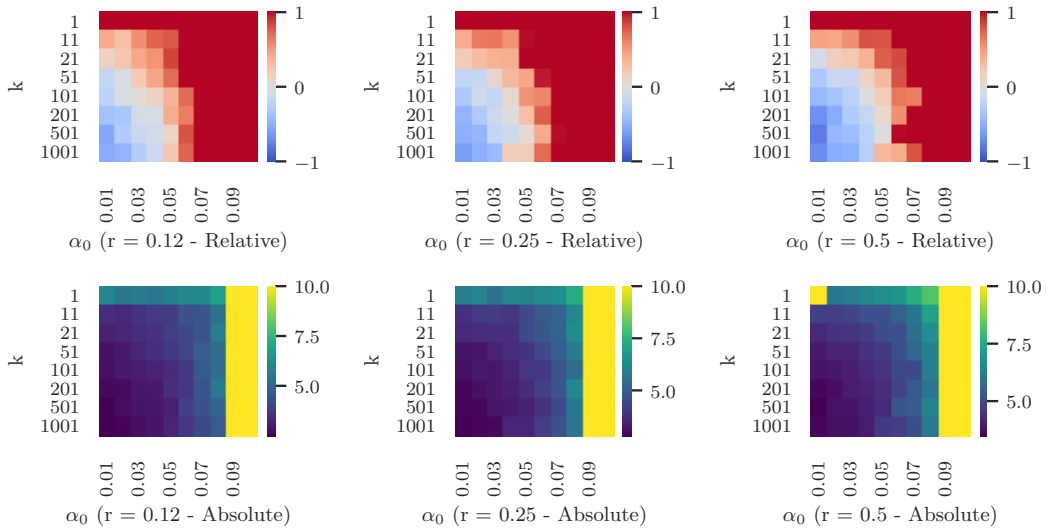

Figure F.1: Ablation study on hyperparameters α_0 and k for Flor$^{(k)}$: We evaluate the average set size of Flor$^{(k)}$ across different values of the initial coverage offset α_0 and the test-time sample rate k at three representative radii ($r \in \{0.12, 0.25, 0.5\}$). [Top row] Relative improvement over Flor$^{(1)}$ (negative values indicate smaller sets - note that the colorbar is cutted at -1 to 1 meaning that areas of same extreme color might project difference more than 1, or less than -1). [Bottom row] Absolute average set sizes. All results are for CIFAR-10 with $\sigma = 0.5$ and ResNet model.

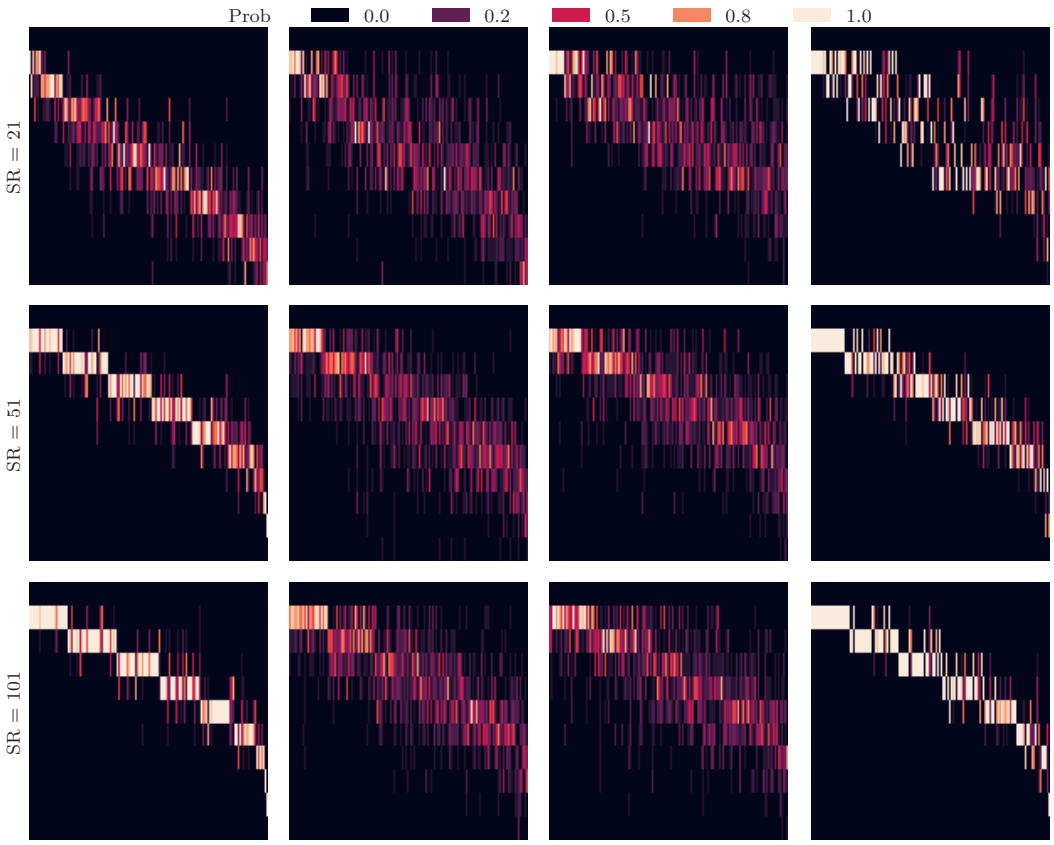

Figure F.2: The distribution of prediction set sizes, conditional to test inputs. Each column of the plot refers to a single x_{n+1}, and the y-axis shows the set size. The brightness of each cell i, j shows the probability $|C(x_{n+i})| = j$. The methods are [from left to right] BinCP, RCP1, Flor$^{(1)}$, and Flor$^{(k)}$. The results are for CIFAR-10 dataset, ResNet model with $\sigma = 0.5$, and $r = 0.12$, with 10^4 calibration time samples. The test time sample rate is shown in the beginning of each row.

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P. W., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., and Nahavandi, S. (2020). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *ArXiv preprint*.

Aldous, D. J. (1981). Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598.

Angelopoulos, A., Bates, S., Malik, J., and Jordan, M. I. (2020). Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*.

Angelopoulos, A. N., Barber, R. F., and Bates, S. (2024). Theoretical foundations of conformal prediction. *arXiv preprint arXiv:2411.11824*.

Angelopoulos, A. N. and Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *ArXiv*, abs/2107.07511.

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

Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., and Schuster, T. (2022). Conformal risk control. *arXiv preprint arXiv:2208.02814*.

Angelopoulos, A. N., Bates, S., Jordan, M. I., and Malik, J. (2021). Uncertainty sets for image classifiers using conformal prediction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Anonymous (2024a). COLEP: Certifiably robust learning-reasoning conformal prediction via probabilistic circuits. In *The Twelfth International Conference on Learning Representations*.

Anonymous (2024b). Conformal inductive graph neural networks. In *The Twelfth International Conference on Learning Representations*.

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

Bahri, D. and Jiang, H. (2021). Locally adaptive label smoothing improves predictive churn. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research.

Bao, H., Dong, L., Piao, S., and Wei, F. (2021). Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*.

Barber, R. F. (2020). Is distribution-free inference possible for binary regression? *arXiv: Statistics Theory*.

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

Barber, R. F., Candès, E. J., Ramdas, A., and Tibshirani, R. J. (2019b). Predictive inference with the jackknife+. *arXiv: Methodology*.

Barber, R. F., Candès, E. J., Ramdas, A., and Tibshirani, R. J. (2022). Conformal prediction beyond exchangeability. *The Annals of Statistics*.

Bazhenov, G., Ivanov, S., Panov, M., Zaytsev, A., and Burnaev, E. (2022). Towards ood detection in graph classification from uncertainty estimation perspective. *ArXiv*.

Berti, P. and Rigo, P. (1997). A glivenko-cantelli theorem for exchangeable random variables. *Statistics & probability letters*, 32(4):385–391.

Bhatia, K., Dahiya, K., Jain, H., Kar, P., Mittal, A., Prabhu, Y., and Varma, M. (2016). The extreme classification repository: Multi-label datasets and code.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

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

Bojchevski, A. and Günnemann, S. (2018). Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *International Conference on Learning Representations*.

Cai, D., Campbell, T., and Broderick, T. (2016a). Edge-exchangeable graphs and sparsity. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*.

Cai, D., Campbell, T., and Broderick, T. (2016b). Edge-exchangeable graphs and sparsity. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Carlini, N., Tramer, F., Dvijotham, K. D., Rice, L., Sun, M., and Kolter, J. Z. (2022). (certified!!) adversarial robustness for free! *arXiv preprint arXiv:2206.10550*.

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

Clarkson, J. (2022). Distribution free prediction sets for node classification. In *Learning on Graphs Conference*.

Clarkson, J. (2023). Distribution free prediction sets for node classification. In *International Conference on Machine Learning*, pages 6268–6278. PMLR.

Clopper, C. J. and Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413.

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

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cresswell, J. C., Sui, Y., Kumar, B., and Vouitsis, N. (2024). Conformal prediction sets improve human decision making. *ArXiv*, abs/2401.13744.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dvijotham, K., Hayes, J., Balle, B., Kolter, Z., Qin, C., György, A., Xiao, K. Y., Gowal, S., and Kohli, P. (2020). A framework for robustness certification of smoothed classifiers using f-divergences. In *International Conference on Learning Representations*.

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

Einbinder, B.-S., Bates, S., Angelopoulos, A. N., Gendler, A., and Romano, Y. (2022). Conformal prediction is robust to label noise. *ArXiv*, abs/2209.14295.

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

Fisch, A., Schuster, T., Jaakkola, T., and Barzilay, R. (2022). Conformal prediction sets with limited false positives. *ArXiv*, abs/2202.07650.

Fischer, M., Baader, M., and Vechev, M. (2021). Scalable certified segmentation via randomized smoothing. In *International Conference on Machine Learning*, pages 3340–3351. PMLR.

Gendler, A., Weng, T., Daniel, L., and Romano, Y. (2022). Adversarially robust conformal prediction. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

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

Ghosh, S., Shi, Y., Belkhouja, T., Yan, Y., Doppa, J., and Jones, B. (2023). Probabilistically robust conformal prediction. In Evans, R. J. and Shpitser, I., editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 681–690. PMLR.

Gosch, L., Geisler, S., Sturm, D., Charpentier, B., Zügner, D., and Günnemann, S. (2024). Adversarial training for graph neural networks: Shchur2018pitfallsog, solutions, and new directions. *Advances in Neural Information Processing Systems*, 36.

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

Hamilton, W. L., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hein, M., Andriushchenko, M., and Bitterwolf, J. (2019). Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*, (2).

Hsu, H. H.-H., Shen, Y., and Cremers, D. (2022). A graph is more than its nodes: Towards structured uncertainty-aware learning on graphs. In *NeurIPS 2022 Workshop: New Frontiers in Graph Learning*.

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

Huang, K., Jin, Y., Candès, E. J., and Leskovec, J. (2023b). Uncertainty quantification over graph with conformalized graph neural networks. *ArXiv*, abs/2305.14535.

Huang, T., Wang, D., Fang, Y., and Chen, Z. (2022). End-to-end open-set semi-supervised node classification with out-of-distribution detection. In *International Joint Conference on Artificial Intelligence*.

Hüllermeier, E. and Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Mach. Learn.*

Jeary, L., Kuipers, T., Hosseini, M., and Paoletti, N. (2024). Verifiably robust conformal prediction.

Kang, J., Zhou, Q., and Tong, H. (2022). Jurygcn: Quantifying jackknife uncertainty on graph convolutional networks. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Kath, C. and Ziel, F. (2021). Conformal prediction interval estimation and applications to day-ahead and intraday power markets. *International Journal of Forecasting*, (2).

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

Kiyani, S., Pappas, G. J., Roth, A., and Hassani, H. (2025). Decision theoretic foundations for conformal prediction: Optimal uncertainty quantification for risk-averse agents. In *Proceedings of Machine Learning Research*, volume 267.

Klicpera, J., Bojchevski, A., and Günnemann, S. (2019). Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

Kuchibhotla, A. K. (2020). Exchangeability, conformal prediction, and rank tests. *arXiv: Methodology*.

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

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

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

Lei, J. and Wasserman, L. A. (2014). Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76.

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

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

Liu, Y., Ding, K., Liu, H., and Pan, S. (2023). Good-d: On unsupervised graph out-of-distribution detection. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, WSDM '23.

Lloyd, J. R., Orbanz, P., Ghahramani, Z., and Roy, D. M. (2012). Random function priors for exchangeable arrays with applications to graphs and relational data. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*.

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

Luo, R., Zhao, S., Kuck, J., Ivanovic, B., Savarese, S., Schmerling, E., and Pavone, M. (2023). Sample-efficient safety assurances using conformal prediction. In *Algorithmic Foundations of Robotics XV*.

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

Massena, T., Andéol, L., Boissin, T., Friedrich, C., Mamalet, F., Serrurier, M., and Gerchinovitz, S. (2025). Efficient robust conformal prediction via lipschitz-bounded networks.

McAuley, J., Targett, C., Shi, Q., and van den Hengel, A. (2015). Image-based recommendations on styles and substitutes. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

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

Mujkanovic, F., Geisler, S., Günnemann, S., and Bojchevski, A. (2023). Are defenses for graph neural networks robust?

Namata, G., London, B., Getoor, L., and Huang, B. (2012). Query-driven active surveying for collective classification.

Orbanz, P. and Roy, D. M. (2014). Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):437–461.

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. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

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

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

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

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine*.

Shafer, G. and Vovk, V. (2008). A tutorial on conformal prediction. *ArXiv*, abs/0706.3188.

Shchur, O., Mumme, M., Bojchevski, A., and Günnemann, S. (2018). Shchur2018pitfallsog of graph neural network evaluation. *ArXiv*, abs/1811.05868.

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

Stadler, M., Charpentier, B., Geisler, S., Zügner, D., and Günnemann, S. (2021). Graph posterior network: Bayesian predictive uncertainty for node classification. *Advances in Neural Information Processing Systems*, 34:18033–18048.

Stutz, D., Cemgil, A. T., Doucet, A., et al. (2021). Learning optimal conformal classifiers. *arXiv preprint arXiv:2110.09192*.

Teng, J., Wen, C., Zhang, D., Bengio, Y., Gao, Y., and Yuan, Y. (2023). Predictive inference with feature conformal prediction. In *The Eleventh International Conference on Learning Representations*.

Tibshirani, R. J., Barber, R. F., Candès, E. J., and Ramdas, A. (2019a). Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*.

Tibshirani, R. J., Foygel Barber, R., Candes, E., and Ramdas, A. (2019b). Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32.

Vazquez, J. and Facelli, J. C. (2022). Conformal prediction in clinical medical sciences. *Journal of Healthcare Informatics Research*.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations*.

Vovk, V. (2012). Conditional validity of inductive conformal predictors. *Machine Learning*.

Vovk, V. (2015). Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, (1).

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

Wang, K., Shen, Z., Huang, C., Wu, C.-H., Dong, Y., and Kanakia, A. (2020). Microsoft Academic Graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413.

Wightman, R. (2019). Pytorch image models. `https://github.com/rwightman/pytorch-image-models`.

Wijegunawardana, P., Gera, R., and Soundarajan, S. (2020). Node classification with bounded error rates.

Yan, G., Romano, Y., and Weng, T.-W. (2024). Provably robust conformal prediction with improved efficiency. *The Twelfth International Conference on Learning Representations*.

Yang, G., Duan, T., Hu, J. E., Salman, H., Razenshteyn, I., and Li, J. (2020). Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pages 10693–10705. PMLR.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

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

Zargarbashi, S. H., Akhondzadeh, M. S., and Bojchevski, A. (2024). Robust yet efficient conformal prediction sets. In *Forty-first International Conference on Machine Learning*.

Zargarbashi, S. H., Akhondzadeh, M. S., and Bojchevski, A. (2025). One sample is enough to make conformal prediction robust. *ArXiv*, abs/2506.16553.

Zargarbashi, S. H., Antonelli, S., and Bojchevski, A. (2023a). Conformal prediction sets for graph neural networks. In *Proceedings of the 40th International Conference on Machine Learning*.

Zargarbashi, S. H., Antonelli, S., and Bojchevski, A. (2023b). Conformal prediction sets for graph neural networks. In *International Conference on Machine Learning*.

Zargarbashi, S. H. and Bojchevski, A. (2025). Robust conformal prediction with a single binary certificate.

Zeng, H., Zhou, H., Srivastava, A., Kannan, R., and Prasanna, V. (2019). Graphsaint: Graph sampling based inductive learning method. *arXiv preprint arXiv:1907.04931*.

Zhang, D., Ye, M., Gong, C., Zhu, Z., and Liu, Q. (2020). Black-box certification with randomized smoothing: A functional optimization based framework. *Advances in Neural Information Processing Systems*, 33:2316–2326.