Should We Pay This Much for Robustness? Efficient Proxy Certificates with Marginal Guarantees

Soroush H. Zargarbashi[⋆,1], **Mohammad Sadegh Akhondzadeh**[⋆,2], **Aleksandar Bojchevski**[2]
[1]CISPA Helmholtz Center for Information Security, [2]University of Cologne
[zargarbashi, akhondzadeh, bojchevski]@cs.uni-koeln.de

Abstract

Modern classifiers remain sensitive to small input perturbations, while strong model-agnostic robustness certificates such as randomized smoothing are often too expensive for deployment, requiring an extensive number (e.g. 2000) of model forward passes per input. We propose a framework that delegates certification to a cheap proxy function while statistically controlling its deviation from the original sound certificate. Given an exchangeable unlabeled calibration set, we tune the proxy via robust conformal risk control to guarantee a bounded marginal (rather than pointwise) false-approval rate: the probability, over future inputs, that the proxy accepts an input whose prediction is flipped due to perturbation. The resulting proxy certificate can be used either as a standalone marginal certificate or as a fallback mechanism that invokes the expensive certificate only when the proxy rejects. We provide both single-sample and sample-heavy calibration recipes. Empirically, as a standalone certificate our method reduces the test-time certification cost to a single model forward while retaining much of the certified performance of randomized smoothing at small radii, and as a fallback it recovers the certified ratio of randomized smoothing at a more modest speedup. This enables cheap yet effective robustness guarantees for larger architectures such as vision transformers and language models.

1 Introduction

High accuracy alone is not sufficient for reliability in machine learning. Although modern neural networks achieve strong empirical performance, their predictions can be highly sensitive to small, often imperceptible, input perturbations [Goodfellow et al., 2014]. This sensitivity can be exploited by adversaries to easily degrade the model's performance. There are many empirical, and heuristic defenses proposed in the literature which have been circumvented by stronger or adaptive attacks [Athalye et al., 2018]. A principled way to address this issue is to use robustness certificates, which provide formal guarantees that the model prediction remains invariant within a predefined perturbation set. Certificates can be viewed as binary decision rules: when they certify an input at a sufficiently large radius, they signal that the prediction can be trusted with respect to the specified threat model, whether the input is clean or even if adversarially perturbed, the perturbation is not effective.

Existing robustness certificates typically obtain such guarantees through architecture-specific verification methods, such as convex relaxations, or mixed-integer programming [Wong and Kolter, 2018, Katz et al., 2017, Tjeng et al., 2019]. While powerful, these methods require white-box access to the model, and often need to be adapted when new architectures or layers are introduced. Alternatively, randomized smoothing (RS) [Cohen et al., 2019] wraps around any black-box classifier, and convolves it with a noise augmentation. The resulting smoothed classifier changes slowly with local perturbations. Hence, one can derive closed-form model-agnostic bounds on its change. This black-box nature is a major practical advantage, allowing it to easily apply to large modern networks, and even to API-accessible systems [Carlini et al., 2022]. However, randomized smoothing-based

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Results on ImageNet/ResNet, r=0.14, σ=0.5, α=0.05, Gaussian smoothing (2000 MC samples: 50 for the majority-vote label and 1950 for the certificate). The verification entry does not use smoothing (result directly reported from the reference); it gives a pointwise guarantee on a different architecture, so it is not directly comparable to the marginal guarantee of ProxC. Speedup is on forward calls and relative to randomized smoothing. Certification-only cost excludes the forwards for the label; end-to-end cost includes them.

Method	Variant	Guarantee	Cert. Ratio	Cert. Accuracy	Error Rate	Forwards / image		
						Cal	Cert.-only (Speedup)	End-to-end (Speedup)
Rand. Smoothing	—	Pointwise	70.06%	56.08%	—	—	1950	2000
Verification	Hu et al. [2023]	Pointwise	—	35.0%	—	—	1	1
ProxC	Standalone	Marginal	56.11%	48.53%	2.29%	2000	1 ($\mathbf{1950}\times$)	51 ($\mathbf{39.2}\times$)
	Fallback	Marginal	72.35%	57.3%	2.29%	2000	779 ($\mathbf{2.50}\times$)	829 ($\mathbf{2.41}\times$)

certificates require a large number of model forward passes (e.g. 2×10^3) to estimate the smoothed class probabilities with high confidence, which can be prohibitively expensive especially with limited resources, or when the model itself is expensive to evaluate.

This cost is especially wasteful because not every input received by a system is adversarial; indeed, most inputs are expected to be clean. Moreover, in many settings, it is cheap to obtain a relatively small set of unlabeled samples representative of the deployment distribution. These observations motivate a distributional relaxation of robustness certification: rather than requiring a local certificate for every individual input, it may suffice to guarantee that certification failure is controllable below an adjustable low probability over test inputs.

In light of the arguments above, we define Proxy Certificate (ProxC), a framework for delegating expensive certification to cheap proxy functions under calibrated statistical control. Through a variant of conformal risk control (CRC) that we define in § 2.2, we control how often the proxy deviates from the expensive function. Our proxy reduces the cost of the certification step to a single model forward. We define three variants for ProxC, tailored for various compute budgets:ProxC-A, that uses a single forward pass per calibration point, ProxC which computes an expensive backbone certificate on the small calibration set, and ProxC-H which can increase the certified ratio, in return for more Monte-Carlo (MC) sampling. All three variants use a single sample to certify each test point. We bound the risk anchored at the latent clean point, which lets a sound certificate computed only on clean calibration points calibrate a cheap proxy that remains valid under perturbation (see § F).

The resulting proxy can be used either as a standalone certificate or as a hybrid fallback mechanism that only calls the expensive certificate on inputs rejected by the proxy. These are two different operating regimes: the standalone proxy gives the largest saving, while the fallback recovers the certified ratio of the backbone at a smaller speedup. As we show in Table 1, by relaxing the soundness to bounded false approvals, we can reduce the cost of the certification step to a single forward, the same as verification-based approaches (importantly, the choice of label passed into the certificate is excluded from the cost evaluation. This can still follow the inference rule of the backbone). ProxC shows a higher certified accuracy compared to verification (note that our guarantee is marginal, and we use a slightly different architecture), and it is cheaper than randomized smoothing. In other words, in the trade-off between computational efficiency and applicability, we now propose a new dimension which is soundness. We propose smoothing-based guarantees with a realistic computational cost that is suitable for larger models like ViT, or language modeling in realistic time-sensitive scenarios.

2 Background

Problem setup. Let $f : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$ be a pre-trained classifier (can be a single model, a pipeline of models, or any general inference policy, e.g. majority vote over random augmentations) that returns a score per class. For an input $\boldsymbol{x}_i \in \mathcal{X}$, its top prediction is $y_i = \arg\max_y f(\boldsymbol{x}_i)_y$. For brevity, whenever f is compared across two inputs, e.g. $f(\tilde{\boldsymbol{x}}) \neq f(\boldsymbol{x})$, we compare the predicted labels, i.e. $\arg\max_y f(\tilde{\boldsymbol{x}})_y \neq \arg\max_y f(\boldsymbol{x})_y$. For the test point, $\boldsymbol{x}_{n+1}$, an adversary introduces a perturbation $\tilde{\boldsymbol{x}}_{n+1} = \boldsymbol{x}_{n+1} + \boldsymbol{\delta}$ aiming to change the model's prediction $f(\tilde{\boldsymbol{x}}_{n+1}) \neq f(\boldsymbol{x}_{n+1})$.

The adversary is limited to keep the perturbation imperceptible – $\tilde{\boldsymbol{x}}_{n+1}$ has to be very close to $\boldsymbol{x}_{n+1}$. This limitation is formally presented as a perturbation ball $\mathcal{B} : \mathcal{X} \to 2^{\mathcal{X}}$ – the set of all points from a given $\boldsymbol{x}$ that the adversary is allowed to choose. A common example for images is the ℓ_2 ball $\mathcal{B}_r(\boldsymbol{x}) = \{\tilde{\boldsymbol{x}} : \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq r\}$. The adversary has white-box access to the model (its architecture, parameters, etc), and the clean input $\boldsymbol{x}_{n+1}$, but is not in control of the inference, and is unaware of the random seed used by the MC samples evaluated at test time. The test-time noise is freshly sampled for every query and is unknown when the adversary chooses its perturbation; hence we do not require a permanently secret global seed. We study the *worst-case* robustness: for *any* $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}_r(\boldsymbol{x}_{n+1})$, the prediction should either remain the same, or the defense should abstain from certifying it. This is what we call the threat model.

Robustness certificate. A robustness certificate is a binary decision function

$$\mathrm{cert}_{f,r}(\boldsymbol{x}_i) = 1 \qquad \Rightarrow \qquad \forall \tilde{\boldsymbol{x}} \in \mathcal{B}_r(\boldsymbol{x}_i) : f(\tilde{\boldsymbol{x}}) = f(\boldsymbol{x}_i). \tag{1}$$

The certificate is *sound*: $\mathrm{cert}_{f,r}(\boldsymbol{x}_i) = 1$ ensures f is stable entirely within $\mathcal{B}_r(\boldsymbol{x}_i)$, but it is not *complete*: $\mathrm{cert}_{f,r}(\boldsymbol{x}_i) = 0$ does not necessarily imply vulnerability. The standard use case of the certificate is to apply it at test time on each (potentially perturbed) input $\tilde{\boldsymbol{x}}_{n+1}$, and if $\mathrm{cert}_{f,r}(\tilde{\boldsymbol{x}}_{n+1}) = 1$, we can safely use the prediction $f(\tilde{\boldsymbol{x}}_{n+1})$; otherwise, we abstain.

Inverse ball $\mathcal{B}^{-1}$. While we allow the adversary to perturb $\boldsymbol{x}$ within $\mathcal{B}(\boldsymbol{x})$, from the defender's perspective, we should ensure that the received $\tilde{\boldsymbol{x}}$ has the same prediction as $\boldsymbol{x}$ which is the reverse act w.r.t. the adversary. Hence, at the test time we use the certificate over the inverse ball $\mathcal{B}^{-1}(\boldsymbol{x}) = \{\tilde{\boldsymbol{x}} : \boldsymbol{x} \in \mathcal{B}(\tilde{\boldsymbol{x}})\}$ which is the smallest set that from any perturbed point $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$ it includes $\boldsymbol{x}$. This is the ball that covers the clean point from any potential perturbation. In symmetric balls, we have $\mathcal{B}^{-1}(\boldsymbol{x}) = \mathcal{B}(\boldsymbol{x})$. An example for non-symmetric ball is token insertion, or deletion, where the inverse ball is the opposite act; i.e. $\mathcal{B}_r^{\mathrm{ins},-1} = \mathcal{B}_r^{\mathrm{del}}$.

Binary and confidence certificate. We use $\mathrm{c}^{\downarrow}$ and $\mathrm{c}^{\uparrow}$ for certified lower and upper bounds, with a slight overload: formally $\mathrm{c}^{\downarrow}(f, \mathcal{B}(\boldsymbol{x})) \leq \inf_{\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})} f(\tilde{\boldsymbol{x}})$ and $\mathrm{c}^{\uparrow}$ is the corresponding upper bound defined with supremum. In case that the bounds only depend on the value $p = f(\boldsymbol{x})$ (not the function, and the point), we write $\mathrm{c}^{\downarrow}(p, \mathcal{B})$ and $\mathrm{c}^{\uparrow}(p, \mathcal{B})$. Certificates often report $\mathrm{c}^{\downarrow}$, and $\mathrm{c}^{\uparrow}$, from which we obtain the 0/1 decision or the certified radii – the largest radius for which the prediction is provably invariant; i.e. $r^*_{\boldsymbol{x}} = \max\{r : \mathrm{c}^{\downarrow}(f, \mathcal{B}_r(\boldsymbol{x})) \geq 1/2\}$. The latter is a common presentation of certificates; larger radii indicate stronger robustness. However ultimately the certification of a point boils down to comparison with a predefined radius r, which is a binary decision. We say $\boldsymbol{x}$ is certified at radius r either if $\mathrm{c}^{\downarrow}(p_{y^\star}(\boldsymbol{x}), \mathcal{B}_r) > 1/2$ or if $\mathrm{c}^{\downarrow}(p_{y^\star}(\boldsymbol{x}), \mathcal{B}_r) > \max_{y \neq y^\star} \mathrm{c}^{\uparrow}(p_y(\boldsymbol{x}), \mathcal{B}_r)$, for $y^\star$ as the top predicted class. In this work, we operate directly on these certified bounds.

Common certificates are often obtained via exact verification methods [Wong and Kolter, 2018, Katz et al., 2017] and randomized smoothing [Cohen et al., 2019, Yang et al., 2020]. The former is a white-box method that requires access to the model's architecture and parameters, and is typically limited to smaller models of known architectures. The latter is a black-box method that can be applied to any model, and can scale to large and modern architectures, however it requires a large number of model forward passes to estimate the smoothed class probabilities with high confidence.

Objective. We replace the sound but expensive certificate with a tuned proxy function that controls the marginal false-approval rate at level α. The guarantee holds for all adversarial perturbations $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$ for any given clean point $\boldsymbol{x}_{n+1}$, and the expectation in Eq. 2 is taken marginally over the distribution of (latent) clean inputs. We assume a calibration dataset $\mathcal{D}_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ of unlabeled points drawn exchangeably from the same distribution as the clean test input $\boldsymbol{x}_{n+1}$ over which we tune the certificate. Only the latent clean point $\boldsymbol{x}_{n+1}$, and not $\tilde{\boldsymbol{x}}_{n+1}$, needs to be exchangeable with $\mathcal{D}_n$; e.g. for unseen deployment inputs, $\mathcal{D}_n$ must be held out from the training data of f, as training and test points are not exchangeable conditional on the model. The outcome is a proxy that takes significantly lower computational cost only by relaxing the soundness of the original certificate to a marginally bounded false-approval rate; formally with $\hat{\mathrm{cert}}_{f,r}$, we guarantee that

$$\mathbb{E}_{\mathcal{D}}\left[\mathbb{I}\{f(\tilde{\boldsymbol{x}}_{n+1}) \neq f(\boldsymbol{x}_{n+1})\} \cdot \hat{\mathrm{cert}}_{f,r}(\tilde{\boldsymbol{x}}_{n+1}) \mid \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})\right] \leq \alpha. \tag{2}$$

In words, over many future (possibly perturbed) inputs, on average at most an α fraction are approved while their prediction is flipped. The expectation is over the calibration set, the clean test point, and

the internal randomness. Same as conformal prediction itself, the risk is bounded in expectation, and for a fixed calibration set can exceed α (with a variance decreasing in the number of calibration points n); the guarantee is also not conditional on any particular input, class, or subpopulation.

Our proxy certificate (ProxC) is built upon two key components, which we introduce next.

2.1 Randomized Smoothing

We recall a high level understanding of randomized smoothing, and defer the details to § B. A smoothing function $\psi : \mathcal{X} \to \mathcal{X}$ augments the input with a random noise from a predefined distribution; $\psi(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{\epsilon}$. Wrapping any base classifier f, we define the smoothed class probability $g(\boldsymbol{x})_y = \Pr_{\boldsymbol{\epsilon}}[f(\boldsymbol{x} + \boldsymbol{\epsilon}) = y]$. Because the push-forward measures $f_{\#}\psi(\boldsymbol{x})$ and $f_{\#}\psi(\tilde{\boldsymbol{x}})$ overlap heavily for nearby inputs, we can derive model-agnostic lower and upper bounds on $g(\tilde{\boldsymbol{x}})_y$ that depend only on $p = g(\boldsymbol{x})_y$ and the perturbation radius. On images a well-known smoothing scheme is the isotropic Gaussian noise of magnitude σ, i.e. $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. For ℓ_1, and ℓ_2 ball, isotropic Gaussian smoothing has closed form lower and upper bounds ($\mathrm{c}^{\downarrow}$, and $\mathrm{c}^{\uparrow}$) as $\Phi_\sigma(\Phi_\sigma^{-1}(p) \pm r)$ regardless of the model's architecture, or the actual value of $\boldsymbol{x}$. Here Φ_σ is the CDF of $\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. For other smoothing schemes, similar bounds can be computed through the recipe from Yang et al. [2020].

2.2 Conformal Risk Control

CRC provides distribution-free, finite-sample guarantees on the expected risk of a system using an exchangeable subsample of its inputs [Angelopoulos et al., 2022]. Let $L : \mathcal{X} \times [0, 1] \to \mathbb{R}$ be a risk function with conservativeness parameter λ, so that $L(\boldsymbol{x}, \lambda)$ is non-increasing in λ, right continuous, upper bounded by $l_{\max}$, and $\min_\lambda L(\cdot, \lambda) \leq \alpha$ for some user-specified α. CRC guarantees that we can find $\lambda^\star$ such that for future test points $\boldsymbol{x}_{n+1}$ drawn from the same distribution, $\mathbb{E}[L(\boldsymbol{x}_{n+1}, \lambda^\star)] \leq \alpha$. Over the holdout calibration points $\mathcal{D}_n = \{\boldsymbol{x}_i\}_{i=1}^n$, and $\bar{L}_n(\lambda) = \frac{1}{n}\sum_{i=1}^n L(\boldsymbol{x}_i, \lambda)$, under exchangeability of $\mathcal{D} = \mathcal{D}_n \cup \{\boldsymbol{x}_{n+1}\}$ CRC sets

$$\lambda^\star = \inf\left\{\lambda : \frac{n}{n+1}\bar{L}_n(\lambda) + \frac{l_{\max}}{n+1} \leq \alpha\right\}, \quad \text{ensuring} \quad \mathbb{E}[L(\boldsymbol{x}_{n+1}, \lambda^\star)] \leq \alpha$$

3 ProxC: Proxy Certificate

Our proxy certificate is based on an internal binary proxy function $h : \mathcal{X} \times [0, 1] \to \{0, 1\}$; the second parameter of this function, λ, controls its conservativeness. Through conformal risk control we compute the smallest valid parameter λ^* that guarantees to control the rate of false-approval made by $h(\cdot, \lambda^*)$. A direct application of CRC would treat false approval as an ordinary risk, guaranteeing validity of $h(\cdot, \lambda)$ for clean test points that are in-distribution with $\mathcal{D}_n$. However, we aim to maintain the guarantee for any $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$, not just the clean point. Thus, the risk observed at test time is not exchangeable with the risks used for calibration. We address this mismatch in § 3.1. After obtaining a valid robustly calibrated parameter $\lambda^\star$, we define $\hat{\mathrm{cert}}_{f,r}(\tilde{\boldsymbol{x}}_{n+1}) = h(\tilde{\boldsymbol{x}}_{n+1}, \lambda^\star)$, and guarantee $\mathbb{E}_{\mathcal{D}}\left[\mathbb{I}\left\{f(\tilde{\boldsymbol{x}}_{n+1}) \neq f(\boldsymbol{x}_{n+1})\right\} \cdot h(\tilde{\boldsymbol{x}}_{n+1}, \lambda^\star)\right] \leq \alpha$ for all $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$.

3.1 Robust Conformal Risk Control Through Randomized Smoothing

Angelopoulos et al. [2022] propose robust risk control by replacing the pointwise risk $L(\boldsymbol{x}_{n+1}, \lambda)$ with its worst-case local value $\widetilde{L}(\boldsymbol{x}, \lambda) = \sup_{\tilde{\boldsymbol{x}} \in \mathcal{B}_r(\boldsymbol{x})} L(\tilde{\boldsymbol{x}}, \lambda)$. Calibrating over $\widetilde{L}$ ensures

$$\mathbb{E}_{\mathcal{D}}[L(\tilde{\boldsymbol{x}}_{n+1}, \lambda) \mid \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x})] \leq \mathbb{E}_{\mathcal{D}}\left[\sup_{\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}_r(\boldsymbol{x}_{n+1})} L(\tilde{\boldsymbol{x}}_{n+1}, \lambda)\right] \leq \alpha.$$

In practice, however, evaluating $\widetilde{L}$ is itself a certification problem. While it can be upper bounded via $\mathrm{c}^{\uparrow}(L(\cdot, \lambda), \mathcal{B}(\boldsymbol{x}))$, this reintroduces the computational cost we seek to avoid, as we use the certificate again. We instead rewrite the approach of Zargarbashi et al. [2025] for general risk control, and generalize it to sample-heavy calibration. We apply randomized smoothing at the level of the expected risk. We define a noise-augmented set $\mathcal{D}^+ = \{\boldsymbol{x} + \boldsymbol{\epsilon} : \boldsymbol{x} \in \mathcal{D}\}$ that preserves exchangeability, yielding a conformal guarantee that is marginal over both the data distribution and

the smoothing noises. Here, randomized smoothing bounds transfer this guarantee (conditional on each individual $\boldsymbol{x}_{n+1}$) to all perturbations in $\mathcal{B}_r(\boldsymbol{x}_{n+1})$, enabling robust single-sample inference without Monte Carlo estimation at deployment.

Proposition 1. *Let $\mathcal{D}_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be exchangeable together with the (potentially unseen) clean test point $\boldsymbol{x}_{n+1}$. Let $L : \mathcal{X} \times [0,1] \to \{0,1\}$ be such that, for every $\boldsymbol{x} \in \mathcal{X}$, the map $\lambda \mapsto L(\boldsymbol{x}, \lambda)$ is non-increasing, right-continuous, and we have $L(\cdot, 1) = 0$. Let $\psi : \mathcal{X} \to \mathcal{X}$ be a smoothing scheme that is independent of the data, with model-agnostic certified bounds $\mathrm{c}^{\downarrow}(\cdot, \mathcal{B}^{-1})$ and $\mathrm{c}^{\uparrow}(\cdot, \mathcal{B})$. Given any user-specified tolerance $\alpha > 0$, let $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_{n+1}$ be iid draws from ψ. With λ^* defined as either*

$$\lambda^* = \inf\{\lambda : \frac{1}{n+1}\left[1 + \sum_{i=1}^{n} L(\boldsymbol{x}_i + \boldsymbol{\epsilon}, \lambda)\right] \leq \mathrm{c}^{\downarrow}(\alpha, \mathcal{B}^{-1})\} \tag{3}$$

$$\textit{or} \quad \lambda^* = \inf\{\lambda : \frac{1}{n+1}\left[1 + \sum_{i=1}^{n} \mathrm{c}_e^{\uparrow}(\Pr_{\boldsymbol{\epsilon}}[L(\boldsymbol{x}_i + \boldsymbol{\epsilon}, \lambda) = 1], \mathcal{B})\right] \leq \alpha - \delta\} \tag{4}$$

where the latter λ^* corresponds to the calibration-heavy variant, we have

$$\mathbb{E}_{\mathcal{D}, \boldsymbol{\epsilon}}[L(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, \lambda^*)] \leq \alpha, \quad \forall \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1}).$$

where $\delta \in [0,1]$ is the confidence of the finite sample estimation, and $\mathrm{c}_e^{\uparrow}(\Pr_{\boldsymbol{\epsilon}}[L(\boldsymbol{x} + \boldsymbol{\epsilon}, \lambda) = 1], \mathcal{B})$ is an upper confidence bound on $\mathrm{c}^{\uparrow}(\Pr_{\boldsymbol{\epsilon}}[L(\boldsymbol{x} + \boldsymbol{\epsilon}, \lambda) = 1], \mathcal{B})$ with confidence $1 - \frac{\delta}{n}$.

Proofs are in § C. For the calibration-heavy approach, we estimate the risk at each point $\boldsymbol{x}_i$ by averaging over m noises $\{\boldsymbol{\epsilon}_{i,j}\}_{j=1}^{m}$; i.e. $\hat{p}_i(\lambda) = \frac{1}{m}\sum_{j=1}^{m} L(\boldsymbol{x}_i + \boldsymbol{\epsilon}_{i,j}, \lambda)$. We then construct the Clopper-Pearson upper confidence bound $\overline{p}_i(\lambda)$ with confidence $1 - \delta/n$, and set $\mathrm{c}_e^{\uparrow}\big(\Pr_{\boldsymbol{\epsilon}}[L(\boldsymbol{x}_i + \boldsymbol{\epsilon}, \lambda) = 1], \mathcal{B}\big) \coloneqq \mathrm{c}^{\uparrow}(\overline{p}_i(\lambda), \mathcal{B})$. Since $\mathrm{c}^{\uparrow}$ is monotone and $\overline{p}_i(\lambda) \geq \Pr_{\boldsymbol{\epsilon}}[L(\boldsymbol{x}_i + \boldsymbol{\epsilon}, \lambda) = 1]$ with the stated confidence, this construction is a valid upper bound on the true smoothed risk (see § A.1).

3.2 ProxC: Conformal Proxy Certificate

Let $\hat{h}(\cdot, \lambda)$ be a binary proxy function that is controlled by a conservativeness parameter λ – for any $\boldsymbol{x} \in \mathcal{X}$ and $\lambda_1 < \lambda_2$, we have $\hat{h}(\boldsymbol{x}, \lambda_1) \geq \hat{h}(\boldsymbol{x}, \lambda_2)$; a trivial example for proxy function is $\hat{h}(\boldsymbol{x}, \lambda) = \mathbb{I}[\max_y f(\boldsymbol{x}) > \lambda]$. We expect the certificate, and the proxy function to ensure that $f(\tilde{\boldsymbol{x}}_{n+1}) = f(\boldsymbol{x}_{n+1})$; hence, we directly define the false approval risk for $\hat{h}$ as

$$\Pr_{\mathcal{D}}\Big[\mathbb{I}\{f(\tilde{\boldsymbol{x}}_{n+1}) \neq f(\boldsymbol{x}_{n+1})\} \cdot \hat{h}(\tilde{\boldsymbol{x}}_{n+1}, \lambda^*)\Big] \leq \alpha \quad \forall \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1}).$$

From Prop. 1, we concluded that augmenting $\boldsymbol{x}$ with a random noise results in black-box extension of the risk control guarantee to the whole $\mathcal{B}(\boldsymbol{x}_{n+1})$. The risk defined above does not allow the proxy function for any inherent randomness. Hence, we replace the deterministic risk with the following probabilistic generalization

$$\Pr_{\mathcal{D}, \boldsymbol{\epsilon}}\Big[\mathbb{I}\{f(\tilde{\boldsymbol{x}}_{n+1}) \neq f(\boldsymbol{x}_{n+1})\} \cdot \hat{h}(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, \lambda^*)\Big] \leq \alpha \quad \forall \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$$

$$\text{equivalently} \quad \mathbb{E}_{\mathcal{D}}\left[\max_{\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})} \mathbb{I}\{f(\tilde{\boldsymbol{x}}_{n+1}) \neq f(\boldsymbol{x}_{n+1})\} \cdot \Pr_{\boldsymbol{\epsilon}}\Big[\hat{h}(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, \lambda^*) = 1\Big]\right] \leq \alpha. \tag{5}$$

This also generalizes the threat model. In a deterministic setup, the adversary's goal is to change the model's prediction on an input. In case the certificate was not sound, a more realistic goal was to change the prediction, and fool the certificate to certify the point. Here, with the probabilistic notion of certificate, the goal of the adversary is to increase the probability of the proxy to approve a point while the label is also flipped. We guarantee the above risk to be bounded through the following:

Proposition 2. *Let the proxy function $\hat{h}$ be non-increasing and right continuous in λ, and let $\hat{h}(\cdot, 1) = 0$. Let $\mathrm{cert}_{f,r}$ be a sound certificate with respect to $\mathcal{B}_r(\cdot)$, and let ψ be a smoothing scheme with model-agnostic upper, and lower bounds $\mathrm{c}^{\uparrow}$, and $\mathrm{c}^{\downarrow}$. Then the proxy function $\hat{h}(\cdot + \boldsymbol{\epsilon}, \lambda^*)$ with*

either

$$\lambda^* = \inf\left\{\lambda : \frac{1}{n+1}\left[1 + \sum_{i=1}^{n} \hat{h}(\boldsymbol{x}_i + \boldsymbol{\epsilon}, \lambda) \cdot (1 - \mathrm{cert}_{f,r}(\boldsymbol{x}_i))\right] \leq \mathrm{c}^{\downarrow}(\alpha, \mathcal{B}^{-1})\right\} \tag{6}$$

or $$\lambda^* = \inf\left\{\lambda : \frac{1}{n+1}\left[1 + \sum_{i=1}^{n} \mathrm{c}^{\uparrow}(\hat{h}_\lambda, \mathcal{B}(\boldsymbol{x}_i)) \cdot (1 - \mathrm{cert}_{f,r}(\boldsymbol{x}_i))\right] \leq \alpha\right\} \tag{7}$$

guarantees Eq. 5 to hold.

Proof. Given that the certificate is sound, for any $\boldsymbol{x}$, and $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$, we have $\mathbb{I}\{f(\tilde{\boldsymbol{x}}) \neq f(\boldsymbol{x})\} \leq 1 - \mathrm{cert}_{f,r}(\boldsymbol{x})$, and therefore

$$\begin{aligned}
&\max_{\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})} \mathbb{I}\{f(\tilde{\boldsymbol{x}}_{n+1}) \neq f(\boldsymbol{x}_{n+1})\} \cdot \Pr_{\boldsymbol{\epsilon}}\left[\hat{h}(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, \lambda^*) = 1\right] \\
\leq &\max_{\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})} \mathbb{I}\{f(\tilde{\boldsymbol{x}}_{n+1}) \neq f(\boldsymbol{x}_{n+1})\} \cdot \max_{\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})} \Pr_{\boldsymbol{\epsilon}}\left[\hat{h}(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, \lambda^*) = 1\right] \\
&\leq (1 - \mathrm{cert}_{f,r}(\boldsymbol{x}_{n+1})) \cdot \max_{\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})} \Pr_{\boldsymbol{\epsilon}}\left[\hat{h}(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, \lambda^*) = 1\right] \\
&\leq (1 - \mathrm{cert}_{f,r}(\boldsymbol{x}_{n+1})) \cdot \mathrm{c}^{\uparrow}(\hat{h}_\lambda, \mathcal{B}(\boldsymbol{x}_{n+1}))
\end{aligned}$$

The risk function has two inputs besides λ – the received point and its ground truth clean point, as an anchor. The anchor variable $\boldsymbol{x}_{n+1}$ should be kept explicit. We define the risk for the exchangeable paired sample $\{(\boldsymbol{x}_i, \boldsymbol{x}_i)\}_{i=1}^n \cup \{(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+1})\}$, and only allow the first coordinate to be perturbed. For the single-sample rule define the binary anchored risk

$$L_{\mathrm{anc}}((\boldsymbol{z}, \boldsymbol{x}), \lambda) = (1 - \mathrm{cert}_{f,r}(\boldsymbol{x}))\, \hat{h}(\boldsymbol{z}, \lambda).$$

This risk is non-increasing, right-continuous, and bounded in λ as it inherits properties from $\hat{h}$. Note that the other term is only 0, or 1, and only a function of the second element $\boldsymbol{x}$. Applying Prop. 1 in single-sample form, i.e. Eq. 3, tunes to λ^* as defined in Eq. 6, and guarantees that

$$\mathbb{E}_{\mathcal{D},\boldsymbol{\epsilon}}\left[(1 - \mathrm{cert}_{f,r}(\boldsymbol{x}_{n+1}))\hat{h}(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, \lambda^*)\right] \leq \alpha, \qquad \forall \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1}).$$

Note that here, the risk control is applied to the tuple, and the certificate only to the first argument (the anchor). The soundness inequality $\mathbb{I}\{f(\tilde{\boldsymbol{x}}_{n+1}) \neq f(\boldsymbol{x}_{n+1})\} \leq 1 - \mathrm{cert}_{f,r}(\boldsymbol{x}_{n+1})$ then implies

$$\mathbb{E}_{\mathcal{D},\boldsymbol{\epsilon}}\left[\mathbb{I}\{f(\tilde{\boldsymbol{x}}_{n+1}) \neq f(\boldsymbol{x}_{n+1})\}\hat{h}(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, \lambda^*)\right] \leq \alpha \quad \forall \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1}).$$

For the calibration-heavy rule, we compute the worst-case smoothed risk of the anchored loss. Since the certificate term $(1 - \mathrm{cert}_{f,r}(\boldsymbol{x}))$ depends only on the anchor $\boldsymbol{x}$ and not on the perturbed first argument, $\mathrm{c}^{\uparrow}(L_{\mathrm{anc}}((\cdot, \boldsymbol{x}), \lambda), \mathcal{B}(\boldsymbol{x})) = (1 - \mathrm{cert}_{f,r}(\boldsymbol{x}))\, \mathrm{c}^{\uparrow}(\hat{h}_\lambda, \mathcal{B}(\boldsymbol{x}))$. Applying the calibration-heavy form of Prop. 1 (i.e. Eq. 4) with this upper bound yields Eq. 7 and the same final guarantee Eq. 5. The core intuition throughout is to define the risk as a function of *both* the perturbed input and the corresponding clean anchor. During calibration, both coordinates coincide and are observable; at test time, the guarantee holds with respect to the certificate evaluated at the (latent) clean point. □

3.3 Augmented Inference

Prop. 2 requires an existing sound certificate at calibration time, which is itself computationally expensive. An alternative is to *incorporate* the noise augmentation directly into the inference rule, defining a probabilistic classifier which already exhibits robustness. In this case, the smoothed inference simultaneously produces a prediction and serves as an abstention rule: the proxy approves when the noise-augmented model agrees with the clean prediction, and the guarantee covers the event that the prediction at $\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}$ toggles from that of $\boldsymbol{x}_{n+1}$, and the proxy nevertheless approves. This is a guarantee on a randomized joint prediction-and-approval event: both the label and the approval are obtained from the same noise draw. The guarantee (for bounded false approval ratio) is written as:

$$\mathbb{E}_{\mathcal{D}}\left[\max_{\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})} \Pr_{\boldsymbol{\epsilon}}\left[\mathbb{I}\{f(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}) \neq f(\boldsymbol{x}_{n+1})\} \cdot \hat{h}(\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}, \lambda^*)\right]\right] \leq \alpha \tag{8}$$

Proposition 3. *Given the proxy function $\hat{h}$, and the smoothing distribution ψ with the same properties as in Prop. 2, the proxy function $h(\boldsymbol{z}) = \hat{h}(\boldsymbol{z} + \boldsymbol{\epsilon}, \lambda^*)$ with*

$$\lambda^* = \inf \left\{ \lambda : \frac{1}{n+1} \left[1 + \sum_{i=1}^{n} \eta(\boldsymbol{x}_i + \boldsymbol{\epsilon}, \lambda, \boldsymbol{x}_i) \right] \leq \mathrm{c}^{\downarrow}(\alpha, \mathcal{B}^{-1}) \right\}$$

and $\eta(\boldsymbol{z}, \lambda, \boldsymbol{x}) = \hat{h}(\boldsymbol{z}, \lambda) \cdot \mathbb{I}\{f(\boldsymbol{z}) \neq f(\boldsymbol{x})\}$ guarantees Eq. 8 to hold.

3.4 ProxC: The framework

While our framework is general, we focus on a trivial example which is a thresholded proxy function $\hat{h}(\boldsymbol{z}, \lambda) = \mathbb{I}[f(\boldsymbol{z})_y > \lambda]$. Before deployment, we calibrate the proxy on $\mathcal{D}_n$ using noise-augmented inputs, following either Prop. 2 or Prop. 3 depending on the desired guarantee, and with either a single noise sample per point or a larger MC budget. Intuitively, for ProxC we run the expensive certificate once per calibration point; only the points it does not certify can be falsely approved, and we pick the least conservative λ for which the proxy approves such (noise-augmented) points rarely enough, at the corrected level $\mathrm{c}^{\downarrow}(\alpha, \mathcal{B}^{-1})$. At test time only the proxy is evaluated. The calibration produces a scalar λ^* for the given radius r. At test time, for the input $\tilde{\boldsymbol{x}}_{n+1}$, one noise sample from ψ is drawn, added to $\tilde{\boldsymbol{x}}_{n+1}$, and the result is passed through the proxy. Based on the desired guarantee our inference can be in either of the following directions: 1) If we target Eq. 5, calibration uses a sound certificate as backbone to identify non-robust points; at deployment the same inference rule produces the label y and the proxy certifies robustness – therefore, we follow the inference rule of the certificate (in our experiments a majority vote over 50 noisy samples); 2) If we target Eq. 8 a single noise augmentation from ψ is used both to obtain the predicted label and to evaluate the proxy. In all cases, the output is a (label, certification decision) pair. Pseudo-code for all three variants is in § A. Based on the budget for calibration, ProxC-A (Prop. 3) is the cheapest option with single sample both in calibration and test, ProxC (Prop. 2) stays in between only sampling to compute the backbone certificate, and ProxC-H uses high sample rate both for the backbone certificate and for estimation of the worst case risk. All three methods use a single sample at test time; see Table 5 in § A.

Fundamental limitations. Our guarantee relaxes the soundness to bounded error – our proxy certifies by mistake with a rate of at most α for either clean or perturbed inputs. So to apply our framework we assume that such bounded error might be tolerable; e.g. large-scale dataset auditing, robustness monitoring, high-throughput content pipelines, or APIs where the proxy decides whether to escalate to the expensive certificate. Additionally, the backbone certified ratio is a performance ceiling; the *sound* certified ratio of ProxC (the fraction of certifications that are correct) is bounded by the backbone certificate's certified ratio; however, the total certified ratio (because of the at-most-α fraction of permitted false approvals) can get slightly higher (see Table 2 ViT at r=0). Our framework relies on a holdout calibration set $\mathcal{D}_n = \{\boldsymbol{x}_i\}_{i=1}^n$ sampled exchangeably from the distribution of clean future test inputs; in contrast to conformal prediction, this assumption is relatively mild because the calibration points are unlabeled. Nevertheless, our approach is similarly sensitive to shifts in the distribution of unobserved clean points. Under distribution shift, the proxy can be recalibrated on a few recent unlabeled inputs by running the expensive certificate only on them. ProxC offers a marginal guarantee rather than conditional. This mirrors the usual limitation of conformal prediction, and risk control. This means that the proxy may be more conservative on some parts of the distribution and less reliable on others. Our main assumption is that the clean inputs follow the deployment distribution. This means that the adversary is not in control of data distribution (which is a shift in subpopulation). In local regions, and in low chance classes the risk can exceed α (we report class-specific risks in Table 11, and Table 12).

These limitations motivate the future work: (i) designing better proxies to increase coverage over the backbone's certified ratio; (ii) defining localized and adaptive forms of CRC, approaching for close-to-conditional guarantees; and (iii) handling distribution shift in the latent clean space.

4 Experiments

Our evaluation focuses on three points, also summarized in Table 1. We show that (i) as guaranteed, the marginal false-approval rate remains below the tolerance α on both clean and adversarial inputs;

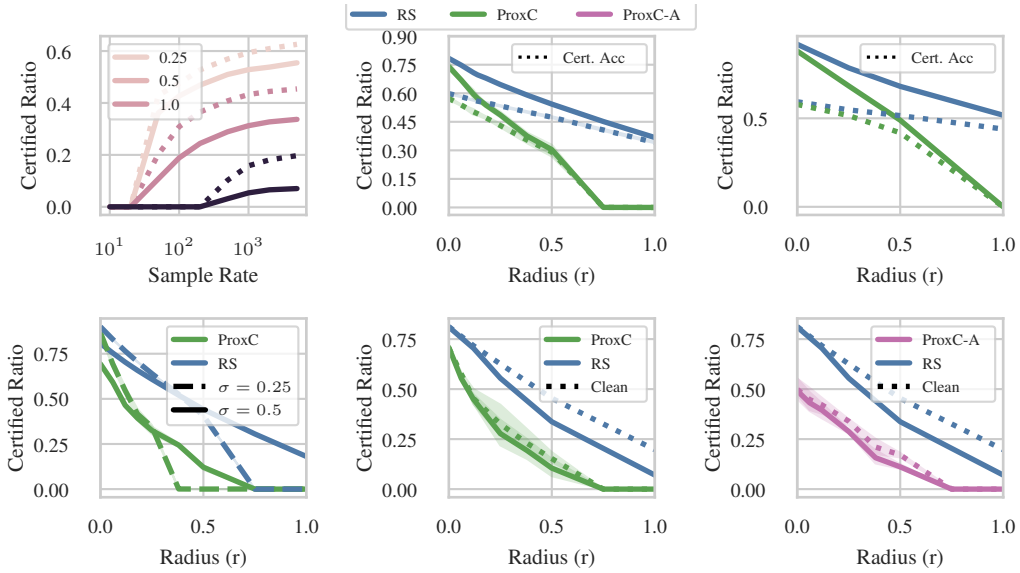


Figure 1: *Top row (ImageNet).* Left: certified ratio of randomized smoothing (RS) as a function of the Monte Carlo sample budget, across perturbation radii r (dotted lines show the performance on perturbed points). Middle: certified ratio (solid) and certified accuracy (dotted) as functions of r for ResNet. Right: the same comparison on the Vision Transformer. *Bottom row (CIFAR-10).* Left: Effect of σ on certified ratio. Middle: certified ratio of ProxC, and Right ProxC-A across various radii. All results are with $\sigma = 0.5$ unless specified. All curves report mean over 5 runs.

(ii) ProxC recovers a large fraction of the certified ratio and accuracy of randomized smoothing at a substantially lower computational cost; and (iii) the computational gains scale with the reduction in Monte Carlo samples required at inference time We further compare ProxC with state-of-the-art robustness certificates beyond randomized smoothing. Despite relaxing soundness to bounded risk, randomized smoothing scales more effectively to larger models such as vision transformers. We evaluate both text and image settings. The code is available at `https://github.com/soroushzargar/ProxC`.

Expected behaviour. The objective of ProxC is not to outperform the backbone certificate on which it is calibrated; by design, the backbone is the ceiling for performance. Instead, our aim is to approximate the certificate with a cheaper function (without the need for MC-sampling), while controlling its false signals statistically. In this sense, ProxC is successful when it remains valid, and can cover a higher portion of backbone certificate's approval ratio; by design it is also supposed to reduce the cost by thousands of forward passes. Therefore we compare the risk as a sanity check, and report the certified ratio and accuracy as the analog to coverage in conformal prediction.

Setup of image experiments. We evaluate ProxC on CIFAR-10 [Krizhevsky et al., 2009] and ImageNet [Deng et al., 2009] datasets under ℓ_2 perturbations. Our results are applicable to any binary certificate, and any smoothing scheme (same as Zargarbashi et al. [2025]). We use two main model configurations. (i) ResNet-110 (CIFAR-10) and ResNet-50 (ImageNet) from Cohen et al. [2019], trained with Gaussian noise augmentation at the same σ as smoothing (default is $\sigma = 0.5$). (ii) Following Salman et al. [2020], and Carlini et al. [2022], the augmented input ($\tilde{\boldsymbol{x}}_{n+1} + \boldsymbol{\epsilon}$) is passed through class-unconditional diffusion followed by a BEiT-L model from Bao et al. [2021]. Models are frozen; only the proxy threshold λ is calibrated. Our default tolerance level is $\alpha = 0.05$ and for calibration-heavy experiment we use the Clopper-Pearson intervals with union confidence of $1 - \delta = 0.999$ — both for randomized smoothing and our own proxy certificate.[1] We report the average over 5 runs with calibration set resampling (unless specified). $|\mathcal{D}_n|$ is set variably between

[1]We assume the certificates are 100% correct, meaning that while computing the risk, at most $\delta = 0.1\%$ of the calibration points mistakenly report zero as error. Replacing with the upper bound risk for them, the overall risk will increase to at most $\leq \alpha + \frac{\delta \cdot n}{n+1} \leq \alpha + \delta$ which here is 5.1%.

Table 2: Certified ratio (CR, %) and certified accuracy (CA, %) on ImageNet with $\sigma = 0.5$. RS uses $n = 2{,}000$ (ResNet-50) and $n = 10{,}000$ (ViT) Monte Carlo samples (single calibration run each). Entries marked † use deterministic (non-smoothing) certificates reported directly from the reference. Note the comparison is between pointwise and marginal guarantees.

		RS (Pointwise)		**ProxC** (Marginal 5% Risk)			**Ref. CA**
Model	r	CR	CA	CR	CA	Risk	(†)
ResNet-50	0.00	78.79	60.00	72.88	56.78	4.17	—
	0.14	70.06	56.08	56.11	48.53	2.29	—
	0.25	64.91	53.61	46.29	41.70	1.25	—
	0.50	54.65	47.62	29.01	28.04	0.26	—
	0.75	45.54	41.12		—		—
ViT	0.00	91.81	59.42	95.00	59.62	5.79	—
	0.25	78.96	55.23	58.56	46.96	1.83	—
	0.50	68.44	51.60	37.54	34.04	0.31	—
	0.75	59.77	48.19		—		—
Verifier-based certificates (†)*:* $\epsilon = 36/255 \approx 0.14$*, CA only, pointwise guarantee*							
LipNeXt-2B†	0.14	— Hu et al. [2026] —					41.2
LipNeXt-1B†	0.14	— Hu et al. [2026] —					40.3
LiResNet+EMMA†	0.14	— Hu et al. [2023] —					35.0
CertViT (ViT-T/16)†	0.14	— Gupta and Verma [2023] —					21.7
CertViT (DeiT-T)†	0.14	— Gupta and Verma [2023] —					17.3

Table 3: Calibration sample budget n_{mc} for ProxC-H on ImageNet (ViT + denoising, $\sigma = 0.5$, $\alpha = 0.05$, averaged over 3 runs). Each cell reports CR (%) with FA (%) in parentheses. All FA values remain $\leq \alpha = 5\%$.

	Base (Cohen)		**ProxC-H** Proxy CR (FA)				
r	CR	CA	$K = 1$k	$K = 2.5$k	$K = 5$k	$K = 10$k	$K = 20$k
0.25	79.0	55.2	62.2 (2.6)	63.9 (2.9)	64.6 (3.0)	63.2 (2.4)	63.6 (2.6)
0.38	73.4	53.3	51.8 (1.5)	54.4 (2.0)	55.7 (2.3)	55.5 (1.5)	56.0 (1.6)
0.50	68.4	51.6	39.2 (0.4)	46.1 (0.9)	48.1 (1.1)	49.1 (1.0)	49.8 (1.0)
0.75	59.8	48.2	—	—	21.5 (0.0)	35.2 (0.2)	38.9 (0.5)

200 and 500 points. We examine our method on 4096 images from CIFAR-10 and 2048 images from ImageNet. By default we run the backbone certificate with 2000 MC samples (unless specified), out of which 50 are used for the majority-vote prediction and 1950 for the certificate. Note that ProxC-A does not require such sampling. For calibration heavy methods, we use 20,000 MC samples at calibration-time which is a one time pre-processing.

Cost accounting. The single forward pass refers only to the certification decision: ProxC and ProxC-H still obtain the label with the inference rule of the backbone (50 forwards, i.e. 51 in total), ProxC-A uses the same noisy forward for the label and the approval (1 in total). The full accounting, the fallback cost across radii, and the break-even point of ProxC-H are in Table 9, Table 10, and § E.

Results on image datasets. Table 1 reports results on ImageNet at $r = 0.14$. There are two use cases: using ProxC as standalone reduces the certification cost from the MC sampling budget to 1, recovering acceptably high portion of the certified accuracy (in our case it is a $1950\times$ reduction of the certification-only cost, and from 2000 to 51 forwards, i.e. $39\times$, end-to-end including the label). In fallback setup, the full certified ratio of the backbone certificate is recovered, at the cost of the speed up. Still in this setup we have a speed up of $2.50\times$ in certification-only cost ($2.41\times$ end-to-end), which decreases with r (Table 10). ProxC-H improves the certified ratio of the ViT pipeline at $r = 0.5$ at the cost of a heavier calibration phase (Table 7). It uses the heavy computation only once, and in the test time it is under the same single-sample regime. The empirical false-approval rate remains well below $\alpha = 5\%$ in almost all experiments, confirming the validity of the theoretical guarantee (the guarantee is in expectation also over calibration sets, meaning that for a single split exceeding α by a minor margin can happen, e.g. ViT at $r = 0$ in Table 2; this variance shrinks as $|\mathcal{D}_n|$ grows, see Table 13). Figure 1 compares ProxC, and ProxC-A with the backbone randomized smoothing certificate in more detail: It shows that with RS we can not reduce the sample rate to acceptable budgets (top left). It also shows the effect of the smoothing magnitude σ (bottom left).

Our main goal can be empirically validated in Table 2. Randomized smoothing, and exact certificates operate at two ends of a trade-off between computational efficiency (from sample-rate), and other

Table 4: NLP task: Certified ratio (CR) and certified accuracy (CA) of the backbone certificate and ProxC and risk. The certification is over r-insertions.

		r					
Method	Metric	0	1	2	3	4	5
Backbone	CR	98.5%	89.3%	77.9%	61.8%	38.8%	14.6%
	CA	91.0%	84.7%	75.2%	60.3%	38.2%	14.5%
ProxC	CR	100.0%	70.7%	45.2%	23.9%	10.3%	4.5%
	CA	92.0%	66.3%	43.6%	23.5%	10.3%	4.5%
Risk	–	1.5%	4.5%	4.1%	3.7%	3.7%	3.3%

benefits like larger robust radii, and black-box access. The certification step of ProxC operates at the same computational cost as verification (a single forward), while its guarantee is from randomized smoothing. Setting aside the relaxation from a sound certificate to marginal bounded error, it highlights the potentials of randomized smoothing once its sampling bottleneck is removed by the proxy — redefining the trade-off. This relaxation emphasizes that this comparison should be interpreted with care – the results are not directly comparable; but it shows the trade-off between soundness, cost, and ratio. In applications where this relaxation is acceptable, the savings can be decisive. As noted before, the Jensen's inequality is the cost of using a single sample in calibration. ProxC-H instead uses sampling to estimate the expected risk. As shown in Table 3, by increasing the calibration time sample rate even new values of r become reachable. Note that this sampling is only done once, during calibration. Later in Table 6 we show the validity under PGD-RS attack which aims to break randomized smoothing. Still under attack the risk remains below the tolerance level.

NLP experiments. We validate the framework on text classification, using CERT-ED [Huang et al., 2024]. The corresponding threat model is the edit (Levenshtein) distance at the token level The backbone smoothed classifier is RoBERTa-base fine-tuned on AG-News (4-class topic classification, 7,600 test inputs) with the matching noise distribution applied at every training forward pass. Table 4 reports the performance of ProxC, and ProxC-H alongside the base certificate. Here, the backbone certificate is over edit distance, and the proxy certificate is over r insertions. We further discuss the theoretical background behind the NLP-experiments in § E.

5 Conclusion

We introduced the proxy certificate (ProxC), a framework that significantly saves computational cost of robustness, by trading the pointwise soundness of a robustness certificate for a bounded marginal false-approval rate. We combine randomized smoothing with conformal risk control, allowing for a robust risk control setup, that works with a single augmented sample for the certification decision at test time (in contrast with heavy MC sampling). Depending on the compute budget, the calibration can be done with heavy MC sampling or again with a single sample. As a standalone certificate, ProxC certifies with one forward pass (51 end-to-end including the label), while as a fallback it recovers the backbone's certified ratio with a more modest speedup (1.3–$3.8\times$ end-to-end on ImageNet). Under the bounded-risk relaxation, ProxC achieves higher certified accuracy than state-of-the-art deterministic verifiers at matching certification cost, although the two provide qualitatively different guarantees (marginal vs. pointwise) on different architectures, while its black-box nature enables application to large architectures such as vision transformers and language models that are out of reach for white-box verification. This highlights the potential in randomized smoothing, as a powerful defense. Our guarantee is marginal and does not extend to individual inputs, classes, or shifted distributions; addressing these, together with a cheaper label prediction and early-stopping backbones, is left for future work.

Acknowledgments and Disclosure of Funding

This work was supported only by the authors' affiliations, the CISPA Helmholtz Center for Information Security and the University of Cologne, and received no third-party funding. The authors

declare no competing interests. We furthermore thank the ITCC (IT Center University of Cologne) for providing compute resources on the DFG-funded HPC (High Performance Computing) system RAMSES (Research Accelerator for Modeling and Simulation with Enhanced Security) as well as support (DFG funding number: INST 216/512-1 FUGG).

References

Anastasios N Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. *arXiv preprint arXiv:2208.02814*, 2022.

Anastasios N Angelopoulos, Rina Foygel Barber, and Stephen Bates. Theoretical foundations of conformal prediction. *arXiv preprint arXiv:2411.11824*, 2024.

Anastasios N Angelopoulos, Stephen Bates, Emmanuel J Candès, Michael I Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025.

Anastasios Nikolas Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *ArXiv*, abs/2107.07511, 2021. URL `https://api.semanticscholar.org/CorpusID:235899036`.

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

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 2022. URL `https://api.semanticscholar.org/CorpusID:247158820`.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021.

Peter Blohm, Patrick Indri, Thomas Gärtner, and Sagar Malhotra. Probably approximately global robustness certification. *arXiv preprint arXiv:2511.06495*, 2025.

Nicholas Carlini, Florian Tramer, Krishnamurthy Dj Dvijotham, Leslie Rice, Mingjie Sun, and J Zico Kolter. (certified!!) adversarial robustness for free! *arXiv preprint arXiv:2206.10550*, 2022.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Marc Fischer, Maximilian Baader, and Martin Vechev. Scalable certified segmentation via randomized smoothing. In *International Conference on Machine Learning*, pages 3340–3351. PMLR, 2021.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, pages 4885–4894, 2017.

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2151–2159. PMLR, 2019.

Asaf Gendler, Tsui-Wei Weng, Luca Daniel, and Yaniv Romano. Adversarially robust conformal prediction. In *International Conference on Learning Representations*, 2021.

Subhankar Ghosh, Yuanjie Shi, Taha Belkhouja, Yan Yan, Janardhan Rao Doppa, and Brian Jones. Probabilistically robust conformal prediction. In *Conference on Uncertainty in Artificial Intelligence*, 2023. URL `https://api.semanticscholar.org/CorpusID:260334753`.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kavya Gupta and Sagar Verma. CertViT: Certified robustness of pre-trained vision transformers. *arXiv preprint arXiv:2302.10287*, 2023. URL https://arxiv.org/abs/2302.10287.

Kai Hu, Andy Zou, Zifan Wang, Klas Leino, and Matt Fredrikson. Unlocking deterministic robustness certification on imagenet. *Advances in Neural Information Processing Systems*, 36:42993–43011, 2023.

Kai Hu, Haoqi Hu, and Matt Fredrikson. LipNeXt: Scaling up Lipschitz-based certified robustness to billion-parameter models. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=2fgzEJPH02.

Zhuoqun Huang, Neil G Marchant, Olga Ohrimenko, and Benjamin IP Rubinstein. Cert-ed: Certifiably robust text classification for edit distance. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10813–10835, 2024.

Linus Jeary, Tom Kuipers, Mehran Hosseini, and Nicola Paoletti. Verifiably robust conformal prediction. 2024.

Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *Computer Aided Verification*, pages 97–117. Springer, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Aounon Kumar, Alexander Levine, Soheil Feizi, and Tom Goldstein. Certifying confidence via randomized smoothing. *Advances in Neural Information Processing Systems*, 33:5165–5177, 2020.

Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi Jaakkola. Tight certificates of adversarial robustness for randomly smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019.

Thomas Massena, Léo Andéol, Thibaut Boissin, Corentin Friedrich, Franck Mamalet, Mathieu Serrurier, and Sébastien Gerchinovitz. Efficient robust conformal prediction via lipschitz-bounded networks. 2025.

Leslie Rice, Anna Bair, Huan Zhang, and J Zico Kolter. Robustness between the worst and average case. *Advances in Neural Information Processing Systems*, 34:27840–27851, 2021.

Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. Classification with valid and adaptive coverage. *arXiv: Methodology*, 2020.

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

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in neural information processing systems*, 32, 2019.

Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J Zico Kolter. Denoised smoothing: A provable defense for pretrained classifiers. *Advances in Neural Information Processing Systems*, 33:21945–21957, 2020.

Vincent Tjeng, Kai Y. Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*, 2019.

Václav Voráček. Treatment of statistical estimation problems in randomized smoothing for adversarial robustness. In *Advances in Neural Information Processing Systems*, volume 37, pages 133464–133486, 2024.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. Algorithmic learning in a random world. 2005.

Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for relu networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 5276–5285. PMLR, 2018.

Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 5286–5295. PMLR, 2018.

Ge Yan, Yaniv Romano, and Tsui-Wei Weng. Provably robust conformal prediction with improved efficiency. *arXiv preprint arXiv:2404.19651*, 2024.

Greg Yang, Tony Duan, J. Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10693–10705. PMLR, 2020.

Soroush H Zargarbashi and Aleksandar Bojchevski. Robust conformal prediction with a single binary certificate. 2025.

Soroush H Zargarbashi, Mohammad Sadegh Akhondzadeh, and Aleksandar Bojchevski. Robust yet efficient conformal prediction sets. In *Forty-first International Conference on Machine Learning*, 2024.

Soroush H. Zargarbashi, Mohammad Sadegh Akhondzadeh, and Aleksandar Bojchevski. One sample is enough to make conformal prediction robust. *ArXiv*, abs/2506.16553, 2025. URL `https://api.semanticscholar.org/CorpusID:279465051`.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

A Algorithms

For simplicity, in the algorithms below we write $\hat{h}(\boldsymbol{z}, \lambda; y)$ for the proxy approval of class y at the received point $\boldsymbol{z}$. The class y can be determined by any arbitrary inference rule. For ProxC, and ProxC-H, we use the conventional classification approach in randomized smoothing, which is to take the majority vote class over a low sample rate (e.g. 50). We also write

$$\bar{R}_n(\lambda) = \frac{1 + \sum_{i=1}^{n} R_i(\lambda)}{n+1},$$

which is the conformal empirical risk with the standard one-point correction. For ProxC, and ProxC-H we use $\mathrm{c}^{\downarrow}(\alpha, \mathcal{B}^{-1})$, and $\alpha - \delta$ as the risk threshold, respectively, following the setup in Prop. 1.

Alg 1 is the pseudo-code for ProxC. It uses the backbone certificate only on $\mathcal{D}_n$ in order to identify the points for which false approval is possible – the points that the backbone certificate approves directly have a risk of 0 (i.e. $c_i = 1$) since for any value of the proxy they are not a false approval. On non-certified points, the proxy is allowed to approve only often enough that the smoothed, robustly corrected CRC risk stays below α. At test time the certificate is no longer evaluated: the calibrated proxy itself is the certificate.

Algorithm 1 PROXC: Proxy Certificate Inference

Require: Calibration set $\mathcal{D}_n = \{\boldsymbol{x}_i\}_{i=1}^n$, classifier f, sound certificate $\mathrm{cert}_{f,r}$, smoothing scheme ψ, proxy $\hat{h}$, target risk α, radius r

Ensure: Calibrated threshold λ^* and proxy certificate $\hat{\mathrm{cert}}_{f,r}$

Calibration:
for $i = 1, \ldots, n$ **do**
  Compute the reference prediction y_i using the inference rule associated with $\mathrm{cert}_{f,r}$
  Compute the certificate decision $c_i \leftarrow \mathrm{cert}_{f,r}(\boldsymbol{x}_i)$
  Draw one independent smoothing noise $\boldsymbol{\epsilon}_i \sim \psi$ and set $\boldsymbol{z}_i \leftarrow \boldsymbol{x}_i + \boldsymbol{\epsilon}_i$
end for
Define $R_i(\lambda) \leftarrow (1 - c_i)\hat{h}(\boldsymbol{z}_i, \lambda; y_i)$
Via binary search, set

$$\lambda^* \leftarrow \inf\left\{\lambda : \bar{R}_n(\lambda) \le \mathrm{c}^{\downarrow}(\alpha, \mathcal{B}^{-1})\right\}.$$

Deployment:
Given $\tilde{\boldsymbol{x}}$, compute $y \leftarrow$ the reference prediction at $\tilde{\boldsymbol{x}}$ (e.g. majority vote over 50 noisy samples)
Draw $\boldsymbol{\epsilon} \sim \psi$ and return y if $\hat{h}(\tilde{\boldsymbol{x}} + \boldsymbol{\epsilon}, \lambda^*; y) = 1$; otherwise abstain
return $\hat{\mathrm{cert}}_{f,r}(\tilde{\boldsymbol{x}}) = \hat{h}(\tilde{\boldsymbol{x}} + \boldsymbol{\epsilon}, \lambda^*; y)$

Alg 2 removes the explicit certificate completely. Instead of asking whether $\mathrm{cert}_{f,r}$ can prove local stability, calibration directly measures whether the randomized inference rule changes the clean prediction. The proxy is then calibrated as an abstention rule for the augmented probabilistic classifier itself. This is the cheapest version of the framework: calibration and deployment both use the same one-sample augmented inference mechanism, and the guarantee is exactly the joint event of toggled prediction and false-approval in Eq. 8.

A.1 Front-Loaded Calibration for ProxC-H

Alg 3 is useful when a single-time high computational cost is available; given higher compute power, it allows for less conservative proxy. Calibration proceeds in two independent phases (detailed below): a tuning phase (Alg 4) that uses m_{tune} noise samples per point to select a candidate threshold λ_{cand} via a simulated CRC condition, and a validation phase (Alg 5) that uses m_{cert} fresh, independent samples to confirm λ_{cand} at the fixed threshold. The two-stage split eliminates the selection effect that arises when the same samples are used to both scan for λ^* and bound the risk at λ^*. Deployment remains identical to PROXC: one noisy proxy evaluation decides whether the prediction is accepted.

Taken together, the three algorithms give a simple recipe. Use PROXC when a trusted certificate can be afforded once on the calibration set and deployment must be cheap. Use PROXC-A when even calibration-time certification is undesirable and one is willing to certify the randomized inference rule directly. Use PROXC-H when calibration is allowed to be sample-heavy; the additional calibration

Algorithm 2 PROXC-A: Proxy-Augmented Inference

Require: Calibration set $\mathcal{D}_n = \{\boldsymbol{x}_i\}_{i=1}^n$, classifier f, smoothing scheme ψ, proxy $\hat{h}$, target risk α, radius r

Ensure: Calibrated threshold λ^* and augmented proxy certificate $\hat{\mathrm{cert}}_{f,r}^{\mathrm{A}}$

Calibration:
for $i = 1, \ldots, n$ **do**
  Compute the clean anchor prediction $y_i \leftarrow f(\boldsymbol{x}_i)$
  Draw one independent smoothing noise $\boldsymbol{\epsilon}_i \sim \psi$ and set $\boldsymbol{z}_i \leftarrow \boldsymbol{x}_i + \boldsymbol{\epsilon}_i$
  Compute the augmented prediction $\tilde{y}_i \leftarrow f(\boldsymbol{z}_i)$
end for
Define $R_i(\lambda) \leftarrow \mathbb{I}\{\tilde{y}_i \neq y_i\}\hat{h}(\boldsymbol{z}_i, \lambda; \tilde{y}_i)$
Set

$$\lambda^* \leftarrow \inf\left\{\lambda : \bar{R}_n(\lambda) \leq \mathrm{c}^{\downarrow}(\alpha, \mathcal{B}^{-1})\right\}.$$

Deployment:
Given $\tilde{\boldsymbol{x}}$, draw $\boldsymbol{\epsilon} \sim \psi$, set $\boldsymbol{z} \leftarrow \tilde{\boldsymbol{x}} + \boldsymbol{\epsilon}$, and compute $y \leftarrow f(\boldsymbol{z})$
Return y if $\hat{h}(\boldsymbol{z}, \lambda^*; y) = 1$; otherwise abstain
return $\hat{\mathrm{cert}}_{f,r}^{\mathrm{A}}(\tilde{\boldsymbol{x}}) = \hat{h}(\tilde{\boldsymbol{x}} + \boldsymbol{\epsilon}, \lambda^*; f(\tilde{\boldsymbol{x}} + \boldsymbol{\epsilon}))$

Algorithm 3 PROXC-H: Calibration-Heavy Proxy Certificate

Require: Calibration set $\mathcal{D}_n = \{\boldsymbol{x}_i\}_{i=1}^n$, classifier f, sound certificate $\mathrm{cert}_{f,r}$, smoothing scheme ψ, proxy $\hat{h}$, target risk α, radius r, tuning sample-rate m_{tune}, certification sample-rate m_{cert}, confidence parameter δ, tuning slack $\delta_0 > 0$

Ensure: Calibrated threshold λ^* and proxy certificate $\hat{\mathrm{cert}}_{f,r}^{\mathrm{H}}$

Precomputation:
for $i = 1, \ldots, n$ **do**
  Compute the reference prediction y_i and certificate decision $c_i \leftarrow \mathrm{cert}_{f,r}(\boldsymbol{x}_i)$
end for
$\lambda_{\mathrm{cand}} \leftarrow \textsc{Tune}(\mathcal{D}_n, y_{1:n}, c_{1:n}, m_{\mathrm{tune}}, m_{\mathrm{cert}}, \alpha, \delta, \delta_0)$
if $\textsc{Validate}(\mathcal{D}_n, y_{1:n}, c_{1:n}, \lambda_{\mathrm{cand}}, m_{\mathrm{cert}}, \alpha, \delta)$ **then**
  $\lambda^* \leftarrow \lambda_{\mathrm{cand}}$
else
  $\lambda^* \leftarrow 1$ ▷ $\hat{h}(\cdot, 1) = 0$: no certifications
end if
Deployment:
Given $\tilde{\boldsymbol{x}}$, compute $y \leftarrow$ the reference prediction at $\tilde{\boldsymbol{x}}$ (e.g. vote over 50 noisy samples)
Draw one smoothing noise $\boldsymbol{\epsilon} \sim \psi$ and return y if $\hat{h}(\tilde{\boldsymbol{x}} + \boldsymbol{\epsilon}, \lambda^*; y) = 1$; otherwise abstain
return $\hat{\mathrm{cert}}_{f,r}^{\mathrm{H}}(\tilde{\boldsymbol{x}}) = \hat{h}(\tilde{\boldsymbol{x}} + \boldsymbol{\epsilon}, \lambda^*; y)$

samples usually reduce the conservativeness of the single-sample proxy while preserving the same marginal false-approval guarantee.

The multiple-testing issue in single-phase calibration. The probabilities used in Eq. 7 need to be estimated via MC sampling. However, using an algorithm like binary search over the same set of samples brings statistical flaw to the estimation. On the other hand, resampling and re-estimating the values at each step of the binary search is also expensive. As an alternative, we divide the computation of λ^* into two phases: tuning (with low sample rate), and validation with high sample rate. The theoretical guarantee is the result of the validation step, therefore we can reuse the samples while tuning.

Two-stage fix. We decouple threshold selection from threshold validation by splitting the noise budget as $m_{\mathrm{tune}} + m_{\mathrm{cert}}$. The *tuning phase* (Alg 4) uses m_{tune} samples per calibration point to select a candidate λ_{cand} by simulating the certification budget: the empirical count $\hat{p}_i^{\mathrm{tune}}(\lambda) \cdot m_{\mathrm{cert}}$ is projected to scale m_{cert}, a Clopper-Pearson bound is computed at that simulated count, and the smoothing upper bound is applied, all with an extra slack δ_0. The *validation phase* (Alg 5) then draws

Table 5: What is guaranteed by randomized smoothing (RS) and by the ProxC variants. In all cases the adversary knows the model weights, the clean input, and the procedure, and may choose any $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$; it does not control or observe the internal randomness (e.g. $\boldsymbol{\epsilon}$).

Method	Prediction rule	Randomness	Averaged over	Guarantee
RS	Majority vote, small n_0	MC-corrected	— (pointwise)	Pointwise soundness
ProxC	Same as RS	One fresh smoothing noise at proxy evaluation	Clean input, calibration set, internal noise	Marginal false-approval control (not pointwise)
ProxC-H	Same as ProxC	One fresh noise draw at test	Same as ProxC, plus calibration MC	Same marginal guarantee, usually less conservative
ProxC-A	One-sample noise-augmented inference; approves with the same draw	One fresh draw affects both prediction and approval	Clean input, calibration set, noises	Controls the joint event {randomized prediction toggles} $\wedge$ {proxy approves}; not a certificate for a fixed deterministic prediction

Algorithm 4 PROXC-H-TUNE: Tuning Phase for PROXC-H

Require: Precomputed $(y_i, c_i)_{i=1}^n$, calibration set $\mathcal{D}_n$, proxy $\hat{h}$, smoothing ψ, m_{tune}, m_{cert}, α, δ, tuning slack δ_0
Ensure: Candidate threshold λ_{cand}

for $i = 1, \dots, n$ **do**
  for $j = 1, \dots, m_{\text{tune}}$ **do**
    Draw $\boldsymbol{\epsilon}_{i,j} \sim \psi$; set $a_{i,j}(\lambda) \leftarrow \hat{h}(\boldsymbol{x}_i + \boldsymbol{\epsilon}_{i,j}, \lambda;\, y_i)$
  end for
  $\hat{p}_i^{\text{tune}}(\lambda) \leftarrow m_{\text{tune}}^{-1} \sum_{j=1}^{m_{\text{tune}}} a_{i,j}(\lambda)$
  $k_i^{\text{sim}}(\lambda) \leftarrow \lfloor \hat{p}_i^{\text{tune}}(\lambda) \cdot m_{\text{cert}} \rfloor$
  $\overline{p}_i^{\text{sim}}(\lambda) \leftarrow \mathrm{CP}^{\uparrow}\big(k_i^{\text{sim}}(\lambda),\, m_{\text{cert}},\, 1 - \delta/n\big)$
  $R_i^{\text{sim}}(\lambda) \leftarrow (1 - c_i)\, \mathrm{c}^{\uparrow}\big(\overline{p}_i^{\text{sim}}(\lambda),\, \mathcal{B}(\boldsymbol{x}_i)\big)$
end for
$\bar{R}_n^{\text{sim}}(\lambda) \leftarrow \frac{1 + \sum_{i=1}^n R_i^{\text{sim}}(\lambda)}{n+1}$
return $\lambda_{\text{cand}} \leftarrow \inf\big\{\lambda : \bar{R}_n^{\text{sim}}(\lambda) \le \alpha - \delta - \delta_0\big\}$

m_{cert} independent noise samples at the *fixed* threshold λ_{cand}. Because λ_{cand} is determined before the validation samples are drawn, the CP bounds in the validation phase are valid for the specific threshold λ_{cand} without any selection correction. Note that both phases reuse the same calibration inputs $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$. This is not a source of selection bias; only reusing the same Monte Carlo noise to both select and validate λ_{cand} would be. Splitting the calibration points into tuning and validation sets is therefore unnecessary, and would only reduce the effective calibration size.

Formal validity.

Proposition 4. *Let λ^* be the output of Alg 3, i.e. $\lambda^* = \lambda_{\text{cand}}$ if the validation check in Alg 5 passes and $\lambda^* = 1$ otherwise, and suppose the validation noise is drawn independently of the tuning noise. Then*

$$\mathbb{E}\big[(1 - c(\boldsymbol{x}_{n+1}))\, \mathrm{c}^{\uparrow}\big(p(\boldsymbol{x}_{n+1}, \lambda^*),\, \mathcal{B}\big)\big] \le \alpha,$$

where $p(\boldsymbol{x}, \lambda) = \Pr_{\boldsymbol{\epsilon}}[\hat{h}(\boldsymbol{x} + \boldsymbol{\epsilon}, \lambda) = 1]$, $c(\boldsymbol{x}) = \mathrm{cert}_{f,r}(\boldsymbol{x})$, and the expectation is over the calibration set, the test point, and the tuning and validation noise.

Proof. Let $R^{\circ}(\boldsymbol{x}, \lambda) = (1 - c(\boldsymbol{x}))\, \mathrm{c}^{\uparrow}(p(\boldsymbol{x}, \lambda), \mathcal{B})$ be the exact (infinite-MC) risk. It is bounded in $[0, 1]$, non-increasing in λ (since p is non-increasing in λ and $\mathrm{c}^{\uparrow}$ is non-decreasing in its first argument), and $R^{\circ}(\cdot, 1) = 0$. The backbone decisions c_i (computed with their own independent MC noise per point) are treated as part of the exchangeable calibration data.

Oracle step. Consider first an oracle procedure that knows p exactly and targets $\alpha - \delta$:

$$\lambda_{\text{or}} = \inf\left\{\lambda : \frac{1 + \sum_{i=1}^n R^{\circ}(\boldsymbol{x}_i, \lambda)}{n+1} \le \alpha - \delta\right\},$$

Algorithm 5 PROXC-H-VALIDATE: Validation Phase for PROXC-H

Require: Precomputed $(y_i, c_i)_{i=1}^n$, calibration set $\mathcal{D}_n$, proxy $\hat{h}$, smoothing ψ, fixed threshold λ_{cand}, m_{cert}, α, δ
Ensure: Boolean: True iff the CRC condition holds at λ_{cand}

for $i = 1, \dots, n$ **do**
  for $j = 1, \dots, m_{\text{cert}}$ **do**
    Draw $\boldsymbol{\epsilon}_{i,j}^{\text{cert}} \sim \psi$; set $a_{i,j}^{\text{cert}} \leftarrow \hat{h}(\boldsymbol{x}_i + \boldsymbol{\epsilon}_{i,j}^{\text{cert}}, \lambda_{\text{cand}};\, y_i)$
  end for
  $k_i^{\text{cert}} \leftarrow \sum_{j=1}^{m_{\text{cert}}} a_{i,j}^{\text{cert}}$
  $\overline{p}_i^{\text{cert}} \leftarrow \mathrm{CP}^{\uparrow}\big(k_i^{\text{cert}},\, m_{\text{cert}},\, 1 - \delta/n\big)$
  $R_i^{\text{cert}} \leftarrow (1 - c_i)\, \mathrm{c}^{\uparrow}\big(\overline{p}_i^{\text{cert}},\, \mathcal{B}(\boldsymbol{x}_i)\big)$
end for
$\bar{R}_n^{\text{cert}}(\lambda_{\text{cand}}) \leftarrow \frac{1 + \sum_{i=1}^n R_i^{\text{cert}}}{n+1}$
return $\big[\bar{R}_n^{\text{cert}}(\lambda_{\text{cand}}) \le \alpha - \delta\big]$

with the convention $\lambda_{\text{or}} = 1$ if the set is empty. Since R° is bounded and non-increasing in λ, the standard CRC argument [Angelopoulos et al., 2022] gives

$$\mathbb{E}[R^\circ(\boldsymbol{x}_{n+1}, \lambda_{\text{or}})] \le \alpha - \delta.$$

Importantly, CRC is applied only to this canonical oracle threshold. In the next steps we show that, for any calibration set, the implemented threshold is at least as conservative as λ_{or} except on an event of probability at most δ.

Tuning and validation. Let $\lambda_{\text{cand}} = g(\boldsymbol{x}_{1:n}, \xi_0)$ be the output of the tuning phase, where ξ_0 denotes the tuning noise. During validation we draw fresh noise ξ_1, independent of ξ_0. Conditional on the calibration inputs and on ξ_0, the threshold λ_{cand} is fixed relative to ξ_1, and for each i the count k_i^{cert} is $\mathrm{Binomial}(m_{\text{cert}},\, p(\boldsymbol{x}_i, \lambda_{\text{cand}}))$. Therefore the Clopper-Pearson bounds are valid binomial bounds at one fixed threshold, i.e. $\overline{p}_i^{\text{cert}} \ge p(\boldsymbol{x}_i, \lambda_{\text{cand}})$ with probability at least $1 - \delta/n$, and no simultaneous confidence band over all λ is needed.

Union bound. Let $\mathcal{E} = \{\overline{p}_i^{\text{cert}} \ge p(\boldsymbol{x}_i, \lambda_{\text{cand}}) \text{ for every } i\}$. By the union bound over the n calibration points, $\Pr(\mathcal{E}^c) \le \delta$. Since $\mathrm{c}^{\uparrow}$ is non-decreasing in its first argument, on $\mathcal{E}$ the validation risks upper-bound the corresponding oracle risks, $R_i^{\text{cert}} = (1 - c_i)\, \mathrm{c}^{\uparrow}(\overline{p}_i^{\text{cert}}, \mathcal{B}) \ge R^\circ(\boldsymbol{x}_i, \lambda_{\text{cand}})$ for every i. Consequently, on $\mathcal{E}$, if the validation passes,

$$\frac{1 + \sum_{i=1}^n R^\circ(\boldsymbol{x}_i, \lambda_{\text{cand}})}{n+1} \le \frac{1 + \sum_{i=1}^n R_i^{\text{cert}}}{n+1} \le \alpha - \delta,$$

so λ_{cand} is feasible for the oracle constraint and, by the definition of λ_{or}, $\lambda^* = \lambda_{\text{cand}} \ge \lambda_{\text{or}}$. If the validation fails, $\lambda^* = 1 \ge \lambda_{\text{or}}$. In both cases, by monotonicity, $R^\circ(\boldsymbol{x}_{n+1}, \lambda^*) \le R^\circ(\boldsymbol{x}_{n+1}, \lambda_{\text{or}})$ on $\mathcal{E}$.

Conclusion. Since $R^\circ \le 1$, we have $R^\circ(\boldsymbol{x}_{n+1}, \lambda^*) \le R^\circ(\boldsymbol{x}_{n+1}, \lambda_{\text{or}}) + \mathbb{I}\{\mathcal{E}^c\}$, and therefore

$$\mathbb{E}[R^\circ(\boldsymbol{x}_{n+1}, \lambda^*)] \le \mathbb{E}[R^\circ(\boldsymbol{x}_{n+1}, \lambda_{\text{or}})] + \Pr(\mathcal{E}^c) \le (\alpha - \delta) + \delta = \alpha.$$

□

Combined with the randomized-smoothing bound and the soundness of the backbone certificate (as in the proof of Prop. 2), this yields the false-approval guarantee of Eq. 5 for ProxC-H. Note that the Monte Carlo validation is not used to apply CRC directly to an adaptively selected empirical risk; it certifies that the selected threshold is at least as conservative as the exact oracle CRC threshold, except on the validation-failure event whose probability is budgeted by δ (the same δ by which the CRC target is lowered to $\alpha - \delta$). The slack δ_0 only affects how often the validation passes, not the validity. As for all our results, the guarantee is in expectation over the calibration set, the test point, and the method's randomness: if $R(\mathcal{D}')$ denotes the expected test risk conditional on a fixed calibration set $\mathcal{D}'$, we have $\mathbb{E}_{\mathcal{D}'}[R(\mathcal{D}')] \le \alpha$, while $R(\mathcal{D}')$ itself can vary across calibration sets. When the validation check fails, $\lambda^* = 1$ and no certifications are issued; this branch is already accounted for in the proof above. Repeating the validation with fresh noise after a failure would require splitting δ

across the attempts. To avoid such failure we use the slack variable δ_0 to stay on the safe side. Note that the $\inf$ in Eq. 4 is to find the most efficient parameter, however for any $\lambda > \lambda^*$ still the risk remains below the desired α. This is due to the monotonicity of the risk. Therefore, finding in any form and then validating it via Alg 5 does not introduce any flaw in the framework.

B Introduction to Randomized Smoothing

Some of our theoretical results rely on the structure of randomized smoothing. For completeness, and to fix notation, we therefore present a self-contained formulation of randomized smoothing.

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the input space and let $\mathcal{Y} = \{1, \dots, K\}$ be the label space. For a classifier $f : \mathcal{X} \to \mathcal{Y}$ and a perturbation set $\mathcal{B}(x)$, pointwise adversarial robustness requires $f(\tilde{x}) = f(x)$ for all $\tilde{x} \in \mathcal{B}(x)$. It is convenient to reduce the K-class problem to a binary one. Fix a predicted class $y^\star$ via some inference rule (e.g. majority vote over m noisy evaluations of f), and define the binary acceptance indicator $f^\top(x) = \mathbf{1}[f(x) = y^\star]$. Let $\bar{f}^\top(x) = \Pr_{\boldsymbol{\epsilon}}[f(x + \boldsymbol{\epsilon}) = y^\star]$ be its smoothed version. Robustness of the prediction $y^\star$ at radius r is then equivalent to $\min_{\tilde{x} \in \mathcal{B}_r(x)} \bar{f}^\top(\tilde{x}) \geq 1/2$, a condition the smoothing certificate can lower-bound in closed form.

More formally, if $\ell : \mathcal{Y} \times \mathcal{Y} \to \{0, 1\}$ denotes the zero-one loss, the worst-case robust error of f at (x, y) is $\sup_{\tilde{x} \in \mathcal{B}(x)} \ell(f(\tilde{x}), y)$. The ideal robust classifier minimizes the expected worst-case error:

$$f^\star_{\mathrm{rob}} \in \operatorname*{arg\,min}_{f : \mathcal{X} \to \mathcal{Y}} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\sup_{\tilde{x} \in \mathcal{B}(x)} \ell(f(\tilde{x}), y) \right].$$

This objective is the natural formalization of robustness, but it is computationally intractable for modern neural networks: the inner maximization ranges over a continuous, high-dimensional perturbation set, while the outer minimization ranges over a non-convex class of classifiers.

Randomized smoothing replaces the deterministic classifier with a noise-averaged one. Let ψ be a smoothing distribution over perturbations $\boldsymbol{\epsilon} \in \mathbb{R}^d$, and define the smoothed class probabilities

$$p_y(x) \coloneqq \Pr_{\boldsymbol{\epsilon} \sim \psi} [f(x + \boldsymbol{\epsilon}) = y], \qquad y \in \mathcal{Y}.$$

The associated smoothed classifier is $g(x) \coloneqq \arg\max_{y \in \mathcal{Y}} p_y(x)$; it predicts the class most frequently returned by the base classifier under random perturbations of the input. For Gaussian smoothing $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I)$, the smoothed classifier admits a closed-form ℓ_2 robustness certificate [Cohen et al., 2019]. Let

$$y_A = \operatorname*{arg\,max}_{y \in \mathcal{Y}} p_y(x), \qquad p_A = p_{y_A}(x), \qquad p_B = \max_{y \neq y_A} p_y(x).$$

If $p_A > p_B$, then the prediction of g is invariant inside the ℓ_2 ball of radius

$$r^\star = \frac{\sigma}{2} \left(\Phi^{-1}(p_A) - \Phi^{-1}(p_B) \right),$$

where Φ denotes the standard Gaussian CDF. Hence randomized smoothing converts probabilistic class separation under Gaussian noise into a deterministic robustness guarantee over an ℓ_2 ball.

More generally, randomized smoothing certifies expectations of bounded functions under the distribution shift induced by adversarial perturbations. For any measurable function $h : \mathcal{X} \to [0, 1]$, define its smoothed value

$$\bar{h}(x) \coloneqq \mathbb{E}_{\boldsymbol{\epsilon} \sim \psi}[h(x + \boldsymbol{\epsilon})].$$

A smoothing certificate provides deterministic functions $\mathrm{c}^\downarrow(\cdot, \mathcal{B}^{-1})$ and $\mathrm{c}^\uparrow(\cdot, \mathcal{B})$ such that, for all $\tilde{x} \in \mathcal{B}(x)$,

$$\bar{h}(\tilde{x}) \leq \mathrm{c}^\uparrow(\bar{h}(x), \mathcal{B}), \qquad \bar{h}(x) \leq \mathrm{c}^\downarrow(\bar{h}(\tilde{x}), \mathcal{B}^{-1}).$$

These inequalities say that the smoothed value at any perturbed point is bounded above by a function of the smoothed value at the clean point, and vice versa.

Optimization formulation. The bound $\mathrm{c}^{\uparrow}(p, \mathcal{B})$ is defined as the solution of a functional optimization problem: given that the smoothed value at the clean point equals p, what is the largest possible smoothed value the same function can achieve at any perturbed point?

$$\mathrm{c}^{\uparrow}(p, \mathcal{B}) \;=\; \sup_{\substack{h\,:\,\mathcal{X}\to[0,1]\\ \tilde{x}\,\in\,\mathcal{B}(x)}} \bar{h}(\tilde{x}) \quad \text{subject to} \quad \bar{h}(x) = p. \tag{9}$$

This is an infinite-dimensional linear program in h. By the Neyman–Pearson lemma, the optimizer is a threshold function on the likelihood ratio of the two noise-shifted distributions:

$$h^{\star}(\mathbf{z}) = \mathbf{1}\left[\frac{q_{\tilde{x}}(\mathbf{z})}{q_x(\mathbf{z})} \geq \tau\right],$$

where q_x and $q_{\tilde{x}}$ are the densities of $x + \boldsymbol{\epsilon}$ and $\tilde{x} + \boldsymbol{\epsilon}$ respectively, and $\tau \geq 0$ is chosen so that $\bar{h}^{\star}(x) = p$. Geometrically, $h^{\star}$ accepts precisely the half-space where the noise-shifted distribution is more likely to have originated from $\tilde{x}$ than from x.

Function independence. A consequence of the Neyman–Pearson characterization is that $\mathrm{c}^{\uparrow}(p, \mathcal{B})$ depends on h *only through* $p = \bar{h}(x)$: any two functions with the same smoothed value at x share the same worst-case upper bound. In particular, the bound does not depend on the architecture of f, the specific input x, or any internal structure of the model — only on the scalar p, the noise level σ, and the perturbation radius r. This is the model-agnostic ("black-box") property that makes randomized smoothing applicable without modification to any classifier, and that justifies writing $\mathrm{c}^{\uparrow}(p, \mathcal{B})$ as a function of p alone.

For Gaussian smoothing $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I)$, the likelihood ratio between $\tilde{x} + \boldsymbol{\epsilon}$ and $x + \boldsymbol{\epsilon}$ is a function of the scalar projection $(\mathbf{z} - x)^{\top}(\tilde{x} - x)/\|\tilde{x} - x\|_2$, which is univariate Gaussian under both distributions. The half-space threshold then integrates to a Gaussian CDF:

$$\mathrm{c}^{\uparrow}(p, \mathcal{B}) = \Phi\Big(\Phi^{-1}(p) + \frac{r}{\sigma}\Big), \qquad \mathrm{c}^{\downarrow}(p, \mathcal{B}^{-1}) = \Phi\Big(\Phi^{-1}(p) - \frac{r}{\sigma}\Big),$$

where $r = \|\tilde{x} - x\|_2$ and Φ is the standard Gaussian CDF. The worst-case perturbation direction is irrelevant: only the radius r enters. Note that $\mathrm{c}^{\uparrow}(\mathrm{c}^{\downarrow}(p, \mathcal{B}^{-1}), \mathcal{B}) = p$, a property used crucially in the proof of Prop. 1.

In practice, the probabilities $p_y(x)$ and smoothed expectations $\bar{h}(x)$ are not available in closed form. They are estimated by Monte Carlo:

$$\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_m \overset{\text{iid}}{\sim} \psi, \qquad \hat{p}_y(x) = \frac{1}{m}\sum_{j=1}^{m} \mathbf{1}\{f(x + \boldsymbol{\epsilon}_j) = y\}.$$

Finite-sample certificates replace the exact probabilities with high-confidence lower bounds (e.g. Clopper–Pearson intervals). This statistical correction is essential: without it, the certificate is only asymptotically valid and may overestimate the certified radius at finite sample budgets. This sampling requirement—thousands of forward passes per input—is the primary computational bottleneck that motivates the proxy framework of the present paper.

C Proofs

Proposition 1. *Let $\mathcal{D}_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be exchangeable together with the (potentially unseen) clean test point $\boldsymbol{x}_{n+1}$. Let $L : \mathcal{X} \times [0, 1] \to \{0, 1\}$ be such that, for every $\boldsymbol{x} \in \mathcal{X}$, the map $\lambda \mapsto L(\boldsymbol{x}, \lambda)$ is non-increasing, right-continuous, and we have $L(\cdot, 1) = 0$. Let $\psi : \mathcal{X} \to \mathcal{X}$ be a smoothing scheme that is independent of the data, with model-agnostic certified bounds $\mathrm{c}^{\downarrow}(\cdot, \mathcal{B}^{-1})$ and $\mathrm{c}^{\uparrow}(\cdot, \mathcal{B})$. Given any user-specified tolerance $\alpha > 0$, let $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_{n+1}$ be iid draws from ψ. With λ^* defined as either*

$$\lambda^* = \inf\{\lambda : \frac{1}{n+1}\left[1 + \sum_{i=1}^{n} L(\boldsymbol{x}_i + \boldsymbol{\epsilon}, \lambda)\right] \leq \mathrm{c}^{\downarrow}(\alpha, \mathcal{B}^{-1})\} \tag{3}$$

$$\textit{or} \quad \lambda^* = \inf\{\lambda : \frac{1}{n+1}\left[1 + \sum_{i=1}^{n} \mathrm{c}_e^{\uparrow}(\Pr_{\boldsymbol{\epsilon}}[L(\boldsymbol{x}_i + \boldsymbol{\epsilon}, \lambda) = 1], \mathcal{B})\right] \leq \alpha - \delta\} \tag{4}$$

where the latter λ^* corresponds to the calibration-heavy variant, we have

$$\mathbb{E}_{\mathcal{D},\boldsymbol{\epsilon}}[L(\tilde{\boldsymbol{x}}_{n+1}+\boldsymbol{\epsilon},\lambda^*)] \leq \alpha, \quad \forall \tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1}).$$

where $\delta \in [0,1]$ is the confidence of the finite sample estimation, and $\mathrm{c}_e^{\uparrow}(\Pr_{\boldsymbol{\epsilon}}[L(\boldsymbol{x}+\boldsymbol{\epsilon},\lambda)=1],\mathcal{B})$ is an upper confidence bound on $\mathrm{c}^{\uparrow}(\Pr_{\boldsymbol{\epsilon}}[L(\boldsymbol{x}+\boldsymbol{\epsilon},\lambda)=1],\mathcal{B})$ with confidence $1-\frac{\delta}{n}$.

Proof for Prop. 1. First, we prove for the single-sample calibration case. Since the noise augmenting function ψ is independent of the data, the set $\mathcal{D}^+ = \{\boldsymbol{x}+\boldsymbol{\epsilon} : \boldsymbol{x} \in \mathcal{D}\}$ is also exchangeable. The risk function evaluated at the new points, i.e. $L(\cdot+\boldsymbol{\epsilon},\lambda)$, maintains all mentioned properties; this is since the non-increasing, right-continuous, and bounded behaviour of L is over the entire domain $\mathcal{X}$. Therefore, we can apply the conformal risk control result of Angelopoulos et al. [2022] to the augmented dataset, meaning that for the arbitrary tolerance α, we first compute $\mathrm{c}^{\downarrow}(\alpha,\mathcal{B}^{-1})$, and for

$$\lambda^* = \inf\{\lambda : \frac{1}{n+1}[1+\sum_{i=1}^{n} L(\boldsymbol{x}_i+\boldsymbol{\epsilon}_i,\lambda)] \leq \mathrm{c}^{\downarrow}(\alpha,\mathcal{B}^{-1})\}$$

CRC guarantees that

$$\mathbb{E}_{\mathcal{D},\boldsymbol{\epsilon}_{[1,n+1]}}[L(\boldsymbol{x}_{n+1}+\boldsymbol{\epsilon}_{n+1},\lambda^*)] \leq \mathrm{c}^{\downarrow}(\alpha,\mathcal{B}^{-1}).$$

with $\boldsymbol{\epsilon}_{[1,n+1]}$ as the noises in $\mathcal{D}^+$. Defining $\beta_{n+1} = \Pr_{\boldsymbol{\epsilon}_{n+1}}\Big[L(\boldsymbol{x}_{n+1}+\boldsymbol{\epsilon}_{n+1},\hat{\lambda})=1\Big]$ we have

$$\mathbb{E}_{\mathcal{D},\boldsymbol{\epsilon}_{[1,n]}}\left[\Pr_{\boldsymbol{\epsilon}_{n+1}}\Big[L(\boldsymbol{x}_{n+1}+\boldsymbol{\epsilon}_{n+1},\hat{\lambda})=1\Big]\right] \leq \mathrm{c}^{\downarrow}(\alpha,\mathcal{B}^{-1}) \quad \Rightarrow \quad \mathbb{E}_{\mathcal{D},\boldsymbol{\epsilon}_{[1,n]}}[\beta_{n+1}] \leq \mathrm{c}^{\downarrow}(\alpha,\mathcal{B}^{-1}),$$

with $\beta_{n+1} = \Pr_{\boldsymbol{\epsilon}_{n+1}}\Big[L(\boldsymbol{x}_{n+1}+\boldsymbol{\epsilon}_{n+1},\hat{\lambda})=1\Big]$. Through randomized smoothing we have

$$\max_{\tilde{\boldsymbol{x}}\in\mathcal{B}(\boldsymbol{x}_{n+1})} \Pr_{\boldsymbol{\epsilon}_{n+1}}\Big[L(\tilde{\boldsymbol{x}}+\boldsymbol{\epsilon}_{n+1},\hat{\lambda})=1\Big] \leq \mathrm{c}^{\uparrow}\Big(\Pr_{\boldsymbol{\epsilon}_{n+1}}\Big[L(\boldsymbol{x}_{n+1}+\boldsymbol{\epsilon}_{n+1},\hat{\lambda})=1\Big],\mathcal{B}\Big) = \mathrm{c}^{\uparrow}\big(\beta_{n+1},\mathcal{B}\big).$$

Zargarbashi et al. [2025] proves that the lower bound $\mathrm{c}^{\downarrow}(\alpha,\mathcal{B}^{-1})$ is a convex function. Similarly, Lemma 1 shows that $\mathrm{c}^{\uparrow}(\beta_{n+1},\mathcal{B})$ is a concave function of β_{n+1}. Since $\mathrm{c}^{\uparrow}(\cdot,\mathcal{B})$ is monotone in its first argument, we can use Jensen's inequality. Let $\alpha_r := \mathrm{c}^{\downarrow}(\alpha,\mathcal{B}^{-1})$, we have

$$\mathbb{E}_{\mathcal{D},\boldsymbol{\epsilon}_{[1,n]}}\left[\mathrm{c}^{\uparrow}\big(\beta_{n+1},\mathcal{B}\big)\right] \leq \mathrm{c}^{\uparrow}\big(\mathbb{E}_{\mathcal{D},\boldsymbol{\epsilon}_{[1,n]}}[\beta_{n+1}],\mathcal{B}\big) \leq \mathrm{c}^{\uparrow}(\alpha_r,\mathcal{B}) = \mathrm{c}^{\uparrow}(\mathrm{c}^{\downarrow}(\alpha,\mathcal{B}^{-1}),\mathcal{B}) = \alpha.$$

The last equality follows from $\mathrm{c}^{\uparrow}(\mathrm{c}^{\downarrow}(\alpha,\mathcal{B}^{-1}),\mathcal{B}) = \alpha$, which is a property of the randomized-smoothing certificate. Therefore, for all $\tilde{\boldsymbol{x}}_{n+1} \in \mathcal{B}(\boldsymbol{x}_{n+1})$

$$\mathbb{E}_{\mathcal{D},\boldsymbol{\epsilon}_{[1,n]}}\left[\Pr_{\boldsymbol{\epsilon}_{n+1}}\Big[L(\tilde{\boldsymbol{x}}_{n+1}+\boldsymbol{\epsilon}_{n+1},\hat{\lambda})=1\Big]\right] \leq \mathbb{E}_{\mathcal{D},\boldsymbol{\epsilon}_{[1,n]}}\left[\max_{\tilde{\boldsymbol{x}}\in\mathcal{B}(\boldsymbol{x}_{n+1})} \Pr_{\boldsymbol{\epsilon}_{n+1}}\Big[L(\tilde{\boldsymbol{x}}+\boldsymbol{\epsilon}_{n+1},\hat{\lambda})=1\Big]\right] \leq \alpha.$$

For the calibration-heavy version, first consider the asymptotically valid setup – estimation with infinite samples; define

$$R(\boldsymbol{x}_i,\lambda) = \mathrm{c}^{\uparrow}\Big(\Pr_{\boldsymbol{\epsilon}}[L(\boldsymbol{x}_i+\boldsymbol{\epsilon},\lambda)=1],\mathcal{B}\Big).$$

By the randomized-smoothing certificate,

$$R(\boldsymbol{x}_i,\lambda) \geq \sup_{\tilde{\boldsymbol{x}}_i\in\mathcal{B}(\boldsymbol{x}_i)} \Pr_{\boldsymbol{\epsilon}}[L(\tilde{\boldsymbol{x}}_i+\boldsymbol{\epsilon},\lambda)=1].$$

Since $L(\cdot,\lambda)$ is non-increasing and right-continuous in λ, upper bounded by 1, and $L(\cdot,1)=0$, the same properties are preserved for $\sup_{\tilde{\boldsymbol{x}}_i\in\mathcal{B}(\boldsymbol{x}_i)} \Pr_{\boldsymbol{\epsilon}}[L(\tilde{\boldsymbol{x}}_i+\boldsymbol{\epsilon},\lambda)=1]$. Replacing this loss with an upper bound (e.g. R) regardless of whether it preserves the same properties, results in a higher parameter λ and therefore it satisfies the guarantee over the original risk. Therefore, we can directly apply CRC with the risk R and tune λ resulting in the guarantee over $\sup_{\tilde{\boldsymbol{x}}_{n+1}\in\mathcal{B}(\boldsymbol{x}_{n+1})} \Pr_{\boldsymbol{\epsilon}}[L(\tilde{\boldsymbol{x}}_{n+1}+\boldsymbol{\epsilon},\lambda)=1]$

$$\lambda^* = \inf\left\{\lambda : \frac{1+\sum_{i=1}^{n} R(\boldsymbol{x}_i,\lambda)}{n+1} \leq \alpha\right\} \quad \Rightarrow \quad \mathbb{E}_{\mathcal{D}}[R(\boldsymbol{x}_{n+1},\lambda^*)] \leq \alpha.$$

$$\mathbb{E}_{\mathcal{D},\boldsymbol{\epsilon}}[L(\tilde{\boldsymbol{x}}_{n+1}+\boldsymbol{\epsilon},\lambda^*)] \leq \mathbb{E}_{\mathcal{D}}\left[\sup_{\tilde{\boldsymbol{x}}\in\mathcal{B}(\boldsymbol{x}_{n+1})} \Pr_{\boldsymbol{\epsilon}}[L(\tilde{\boldsymbol{x}}+\boldsymbol{\epsilon},\lambda^*)=1]\right] \leq \mathbb{E}_{\mathcal{D}}[R(\boldsymbol{x}_{n+1},\lambda^*)] \leq \alpha.$$

For the finite-sample version, the confidence event must hold simultaneously over all calibration points. Using the union bound, the failure probability over the n calibration points, each with confidence $1-\frac{\delta}{n}$, accumulates to δ. Notably if we calibrate CRC on $\alpha-\delta$ instead of α, then the total failure probability is at most $\alpha-\delta+\delta=\alpha$. This requires the bounds to hold at the selected λ^*; when λ^* is searched over the same Monte Carlo samples that are used for the bounds, pointwise bounds do not guarantee this. This is why in practice we use the two-phase procedure of § A.1, whose complete failure-probability is given in the proof of Prop. 4.

□

An important building block in the above proof is the following lemma.

Lemma 1. *$\mathrm{c}^{\uparrow}(\beta,\mathcal{B})$ is concave and monotonically increasing w.r.t. β.*

Proof. The Lagrangian form of the upper-bound problem is:

$$\begin{aligned}
\mathcal{L}(\beta,\lambda) &= \max_{h\in\{0,1\}^{|\mathcal{X}|}} \Pr_{\boldsymbol{\epsilon}}[h(\tilde{\boldsymbol{t}}+\boldsymbol{\epsilon})=1] - \lambda\left(\Pr_{\boldsymbol{\epsilon}}[h(\boldsymbol{t}+\boldsymbol{\epsilon})=1]-\beta\right)\\
&= \max_{h\in\{0,1\}^{|\mathcal{X}|}} E_{\boldsymbol{z}\sim q}[h(\boldsymbol{z})] - \lambda\cdot E_{\boldsymbol{z}\sim p}[h(\boldsymbol{z})-\beta]\\
&= \lambda\cdot\beta + \max_{h\in\{0,1\}^{|\mathcal{X}|}} E_{\boldsymbol{z}\sim q}[h(\boldsymbol{z})] - \lambda\cdot E_{\boldsymbol{z}\sim p}[h(\boldsymbol{z})]\\
&= \lambda\cdot\beta + \max_{h\in\{0,1\}^{|\mathcal{X}|}} \int_{\mathcal{X}} (q(\boldsymbol{z})-\lambda\cdot p(\boldsymbol{z}))\cdot h(\boldsymbol{z})\mathrm{d}\boldsymbol{z}.
\end{aligned}$$

The optimal classifier is:

$$h(\boldsymbol{z}) = \begin{cases} 1 & \text{if} \quad q(\boldsymbol{z})-\lambda\cdot p(\boldsymbol{z})\geq 0\\ 0 & \text{otherwise.}\end{cases}$$

Intuitively, to maximize the term $\int_{\mathcal{X}}(q(\boldsymbol{z})-\lambda\cdot p(\boldsymbol{z}))\cdot h(\boldsymbol{z})\mathrm{d}\boldsymbol{z}$, we select $h(\boldsymbol{z})=1$ whenever the integrand is positive.

The resulting dual with $\lambda\geq 0$ is:

$$\mathcal{L}(\beta,\lambda) = \lambda\cdot\beta + \int \max\{0, q(\boldsymbol{z})-\lambda\cdot p(\boldsymbol{z})\}\mathrm{d}\boldsymbol{z} = \lambda\cdot\beta + l(\lambda),$$

where $l(\lambda) := \int \max\{0, q(\boldsymbol{z})-\lambda\cdot p(\boldsymbol{z})\}\mathrm{d}\boldsymbol{z}$ depends only on λ.

Minimizing over λ gives:

$$\mathrm{c}^{\uparrow}(\beta,\mathcal{B}) = \min_{\lambda} \lambda\cdot\beta + l(\lambda),$$

which is a pointwise minimum of affine functions in β, and therefore concave in β.

The monotonicity w.r.t. β directly follows from the definition. Increasing β enlarges the feasible set, and therefore the optimal value cannot decrease.

□

Proof for Prop. 3. This is a direct implication of Prop. 1 with the risk function defined as $\eta(\boldsymbol{z},\lambda,\boldsymbol{x}) = \hat{h}(\boldsymbol{z},\lambda)\cdot\mathbb{I}\{f(\boldsymbol{z})\neq f(\boldsymbol{x})\}$. The risk function inherits the properties from $\hat{h}$, as $\mathbb{I}\{f(\boldsymbol{z})\neq f(\boldsymbol{x})\}$ ranges in $\{0,1\}$ and is independent of λ. The upper bound is defined as $\mathrm{c}^{\uparrow}(\eta(\cdot,\lambda,\boldsymbol{x}),\mathcal{B}(\boldsymbol{x}))$. The

final guarantee is $\Pr_{\mathcal{D},\boldsymbol{\epsilon}}\Big[\hat{h}(\tilde{\boldsymbol{x}}_{n+1}+\boldsymbol{\epsilon},\lambda^*)\cdot\mathbb{I}\{f(\tilde{\boldsymbol{x}}_{n+1}+\boldsymbol{\epsilon})\neq f(\boldsymbol{x}_{n+1})\}=1\Big]\leq\alpha$ for all $\tilde{\boldsymbol{x}}_{n+1}\in\mathcal{B}(\boldsymbol{x}_{n+1})$.

Alternatively we can define a calibration-heavy version for which we have

$$\lambda^*=\inf\left\{\lambda:\frac{1}{n+1}\left[1+\sum_{i=1}^{n}\mathrm{c}^{\uparrow}(\eta(\cdot,\lambda,\boldsymbol{x}_i),\mathcal{B}(\boldsymbol{x}_i))\right]\leq\alpha\right\}$$

however there is no use-case for it that can not be better accomplished by ProxC, or ProxC-H. □

D Technical Details

False-approval rate vs. false positive rate. A more intuitive risk to tune ProxC with, instead of Eq. 2, would be to control the *false positive rate* (FPR), defined as the conditional probability of a proxy approval given that the point is non-robust:

$$\begin{aligned}\mathrm{FPR}\;&=\;\Pr\big[\hat{\mathrm{cert}}_{f,r}(\tilde{\boldsymbol{x}}_{n+1})=1\;\big|\;f(\tilde{\boldsymbol{x}}_{n+1})\neq f(\boldsymbol{x}_{n+1})\big]\\&=\;\frac{\Pr\big[\hat{\mathrm{cert}}_{f,r}(\tilde{\boldsymbol{x}}_{n+1})=1,\;f(\tilde{\boldsymbol{x}}_{n+1})\neq f(\boldsymbol{x}_{n+1})\big]}{\Pr[f(\tilde{\boldsymbol{x}}_{n+1})\neq f(\boldsymbol{x}_{n+1})]}.\end{aligned}$$

Controlling $\mathrm{FPR}\leq\alpha$ is equivalent to controlling our marginal false-approval rate at the adjusted level $\alpha\cdot\rho$, where $\rho=\Pr[f(\tilde{\boldsymbol{x}}_{n+1})\neq f(\boldsymbol{x}_{n+1})]$ is the marginal non-robustness rate of the classifier. This coupling introduces an undesirable dependence on the quality of the underlying classifier. For a poorly robust classifier—one for which the sound certificate denies a large fraction of inputs—ρ is large, and the same absolute number of false approvals is normalized by a large denominator, making $\mathrm{FPR}\leq\alpha$ easy to satisfy. Conversely, for a near-fully robust classifier where ρ is small, the denominator shrinks, the FPR estimate becomes a high-variance ratio, and calibration grows sensitive to individual false approvals in the calibration set. The marginal false-approval rate avoids this instability by treating the risk as a joint probability: the quantity $(1-\mathrm{cert}_{f,r}(\boldsymbol{x}))\hat{h}(\boldsymbol{x}+\boldsymbol{\epsilon},\lambda)\in[0,1]$ is a bounded random variable whose sample mean is directly controlled by CRC with finite-sample guarantees, regardless of ρ. In this sense, our choice is not merely a notational preference: it ensures that the difficulty of proxy calibration is invariant to the non-robustness rate of the backbone classifier.

E Supplementary on Experiments

Perturbed-data evaluation. We additionally evaluate whether the calibrated proxy certificates remain reliable when the test inputs are explicitly perturbed. For each clean CIFAR-10 test point, we generate an adversarial perturbation using the PGD-RS attack within the ℓ_2 ball of radius r, and then evaluate the proxy certificate on the perturbed input. This setting directly tests the failure mode controlled by our risk: a proxy approval on an input whose prediction can change under perturbation. In Table 6, we report the certified ratio (CR), certified accuracy (CA), and the empirical false-approval risk under attack. As expected, increasing r makes the certification problem harder and decreases both CR and CA (both in the backbone certificate, and in our own ProxC, and ProxC-A). At the same time, the average empirical risk remains below the target level $\alpha=0.05$ across all runs, showing that the proxy remains conservative even when evaluated on attacked inputs. Calibration fails for both ProxC and ProxC-A at $r\geq0.75$, where the perturbed problem becomes too severe for either proxy to satisfy the prescribed risk constraint. This results in ProxC, and ProxC-A rejecting all inputs for that radius.

Detailed results. Table 7 reports results on ImageNet with the denoising + ViT pipeline. The ProxC-H approach is calibrated over 20,000 calibration MC samples and 400 calibration points.

Table 8 presents results on CIFAR-10 (σ=0.5) for ProxC under three risk budgets $\alpha\in\{2\%,5\%,10\%\}$. As expected, increasing α grants the proxy a looser budget, allowing it to certify a larger fraction of samples. At r=0.00, ProxC (α=2%) certifies only 48.8% of samples—well below RS (81.0%)—while remaining highly conservative with an empirical risk of 1.6%. With α=5%, the proxy closes much of this gap (70.9% CR), and at α=10% it slightly exceeds RS (82.3%), certifying more samples than the standard certificate by accepting a higher false-certificate rate

Table 6: Certified ratio (**CR**, %) and certified accuracy (**CA**, %) on *perturbed* CIFAR-10 ($\sigma = 0.5$, calibration fraction 0.2), reported as mean $\pm$ std over two independent calibration runs. Test datapoints are attacked with PGD-RS attack on ℓ_2 ball. **Risk** is the empirical false-approval rate of the proxy certificate under attack. Calibration fails at $r \geq 0.75$ for both methods.

	ProxC			**ProxC-A**		
r	CR	CA	Risk	CR	CA	Risk
0.00	$70.83_{\pm 1.56}$	$56.25_{\pm 1.45}$	$1.91_{\pm 0.80}$	$49.18_{\pm 5.46}$	$42.31_{\pm 3.69}$	$4.86_{\pm 1.99}$
0.06	$55.47_{\pm 0.78}$	$45.31_{\pm 1.19}$	$0.69_{\pm 0.15}$	$43.17_{\pm 4.80}$	$37.82_{\pm 3.48}$	$3.53_{\pm 1.16}$
0.12	$44.88_{\pm 2.36}$	$38.45_{\pm 1.48}$	$0.69_{\pm 0.30}$	$39.49_{\pm 3.81}$	$34.28_{\pm 2.43}$	$2.72_{\pm 0.58}$
0.25	$27.78_{\pm 6.43}$	$23.78_{\pm 4.56}$	$0.26_{\pm 0.26}$	$28.81_{\pm 1.05}$	$25.24_{\pm 0.68}$	$1.60_{\pm 0.31}$
0.38	$19.27_{\pm 5.13}$	$16.32_{\pm 4.05}$	$0.09_{\pm 0.15}$	$15.70_{\pm 3.39}$	$14.15_{\pm 2.69}$	$0.51_{\pm 0.22}$
0.50	$10.50_{\pm 4.33}$	$9.46_{\pm 3.30}$	$0.09_{\pm 0.15}$	$11.13_{\pm 1.18}$	$9.98_{\pm 1.25}$	$0.35_{\pm 0.02}$
≥ 0.75		—			—	

Table 7: Certified Ratio (CR, %) and Certified Accuracy (CA, %) on ImageNet with ViT + Denoising pipeline (σ=0.5, 400 calibration samples, 3 independent runs, 99.9% confidence). $\alpha = 5\%$. All values: mean $\pm$ std over runs. "—" denotes calibration failure.

	RS (backbone)		**ProxC-H**			**ProxC**		
r	CR	CA	CR	CA	Risk	CR	CA	Risk
0.00	91.8 ± 0.4	59.4 ± 0.6	93.2 ± 2.1	58.5 ± 0.9	4.4 ± 1.1	95.0 ± 3.2	59.6 ± 0.6	5.8 ± 1.6
0.06	88.5 ± 0.4	58.5 ± 0.5	82.0 ± 2.1	55.6 ± 1.1	3.7 ± 0.7	84.1 ± 2.8	57.2 ± 0.7	4.1 ± 1.0
0.12	85.6 ± 0.3	57.7 ± 0.5	75.2 ± 2.5	54.0 ± 1.2	3.2 ± 0.8	76.7 ± 4.0	55.0 ± 1.4	3.5 ± 1.1
0.18	82.2 ± 0.3	56.5 ± 0.5	68.5 ± 2.0	51.8 ± 1.1	2.5 ± 0.6	67.1 ± 4.4	51.1 ± 1.8	2.7 ± 1.0
0.25	79.0 ± 0.4	55.2 ± 0.5	63.6 ± 1.6	49.9 ± 0.9	2.6 ± 0.4	58.6 ± 2.4	47.0 ± 1.3	1.8 ± 0.3
0.38	73.4 ± 0.3	53.3 ± 0.4	56.0 ± 2.0	45.9 ± 1.1	1.7 ± 0.3	47.6 ± 2.8	40.9 ± 1.8	0.9 ± 0.2
0.50	68.4 ± 0.1	51.6 ± 0.4	49.8 ± 1.1	42.1 ± 0.8	1.0 ± 0.0	37.5 ± 6.0	34.0 ± 4.6	0.3 ± 0.2
0.75	59.8 ± 0.2	48.2 ± 0.3	38.9 ± 0.9	35.1 ± 0.5	0.5 ± 0.0		—	

(9.6%). Across all radii and risk levels, the empirical risk stays within the target budget (except 5.6% for α=5% at r=0, which is within the split-to-split variation of a single run, see Table 13), demonstrating the validity of the conformal calibration procedure.

Fallback cost across radii. In the fallback mode, the backbone certificate is called exactly when the proxy rejects, so the reject rate is $1 - \mathrm{CR}$ and the cost follows the formula in Table 9. Table 10 reports the reject rate and the resulting cost on ImageNet with ResNet (2000 samples in total: 50 for the label and 1950 for randomized smoothing). We report both the certification-only cost (compared to 1950 forwards) and the end-to-end cost including the 50 forwards of the label (compared to 2000 forwards). As the radius grows, the proxy rejects more inputs, and the speedup of the fallback decreases. ProxC-H always costs more than ProxC, but this gap is amortized over the queries, and it allows for a higher certified ratio, especially at larger radii (Table 3).

Class-stratified risk. The guarantee of ProxC is marginal: it says nothing about any particular class, subpopulation, or input. To assess how concentrated the false-approval risk is, Table 11 reports the risk on ImageNet stratified to the best and the worst 100 classes, and shows that the risk is heterogeneous across classes; however, aggregating over groups of 100 classes is only a partial characterization, and the marginal guarantee does not establish class-conditional or worst-class control. Table 12 reports the per-class risk on CIFAR-10. Four of the ten classes exceed α_r at each radius (up to 3.46% at $r = 0.12$ and 2.63% at $r = 0.25$, both for truck), and the risk conditional on approval is highest for classes with a low certified ratio (e.g. 12% for cat and deer at $r = 0.12$). Binned by the proxy score, the false approvals concentrate just above the threshold: at $r = 0.12$ the test points with a score between 0.66 and 0.88 have a risk of about 10–11%, while the top decile has 0.1%. In order to avoid such imbalance, one can use the techniques like Mondrian conformal prediction to compute the λ threshold separately for each class, but this requires a larger calibration set to maintain the same statistical confidence.

Calibration-set size. The guarantee of ProxC holds in expectation over the calibration set, therefore for a single split the risk can exceed the tolerance. Table 13 studies the effect of $n = |\mathcal{D}_n|$ on CIFAR-

Table 8: Certified Ratio (CR, %) and Certified Accuracy (CA, %) on CIFAR-10 (σ=0.5, ResNet backbone, single run). Results are for three risk levels 2%, 5%, and 10%.

	RS (backbone)		**ProxC** ($\alpha = 2\%$)			**ProxC** ($\alpha = 5\%$)			**ProxC** ($\alpha = 10\%$)		
r	CR	CA	CR	CA	Risk	CR	CA	Risk	CR	CA	Risk
0.00	81.0	63.7	48.8	42.4	1.6	70.9	56.1	5.6	82.3	61.9	9.6
0.06	75.8	61.1	41.7	37.6	1.4	59.3	49.2	4.4	71.7	56.6	8.2
0.12	70.7	58.6	29.6	27.7	0.5	47.3	42.1	2.2	59.8	49.9	5.6
0.18	66.1	55.9	27.9	26.3	0.6	35.4	32.7	1.6	49.6	42.4	4.4
0.25	61.2	53.0	20.8	19.9	0.4	29.6	27.7	1.2	38.5	34.9	2.8
0.38	52.2	46.7	18.3	18.1	0.3	22.6	21.8	0.8	29.1	27.3	1.7
0.50	44.4	40.7		—		17.0	16.5	0.6	19.4	18.7	0.9
0.75	30.8	29.6		—			—		6.6	6.5	0.2

Table 9: Forward passes per test input on ImageNet (2000 MC samples for randomized smoothing: 50 for the label and $N_{\mathrm{MC}} = 1950$ for the certificate). CR is the certified ratio of the proxy; the fallback calls the backbone certificate only when the proxy rejects.

Variant	Label	Certification-only	End-to-end
Randomized smoothing (backbone)	50	1950	2000
ProxC/ ProxC-H (standalone)	50	1	51
ProxC-A	shared	1	1
ProxC/ ProxC-H (fallback)	50	$\mathrm{CR} \cdot 1 + (1 - \mathrm{CR})(1 + N_{\mathrm{MC}})$	50 + certification-only

10 with the ResNet-110 of Cohen et al. [2019] ($\sigma = 0.5$, $\alpha = 5\%$). We compute the backbone certificate (2000 MC samples) once on all 10,000 test images, and for each n draw 500 random calibration sets; the remaining images are used for testing. Since the proxy is calibrated on clean points at the corrected level $\alpha_r = \mathrm{c}^{\downarrow}(\alpha, \mathcal{B}^{-1})$, we compare the empirical risk with α_r. For every r and n, the mean risk does not exceed α_r by more than 0.2 standard errors, and it matches the CRC value $\lfloor \alpha_r(n+1) \rfloor / (n+1)$. The CRC correction requires $n \geq 1/\alpha_r - 1$ (e.g. 62 points at $r = 0.25$, and 244 at $r = 0.5$), below which the proxy approves nothing, and the CR levels off at $n \approx 500$. At $n = 500$ the results are within one standard deviation of the single-run values in Table 8.

Compute resources. All experiments were run on NVIDIA H100 and A100 GPUs. For efficiency, we precomputed the noisy logits required by the randomized-smoothing certificate and by the proxy variants, and reused these cached logits across calibration and evaluation runs. This means that the main GPU cost comes from the forward passes needed to obtain the Monte Carlo samples. Once the logits are available, the remaining steps, including calibration, threshold selection, empirical risk estimation, and the computation of the certified lower and upper bounds, are lightweight post-processing operations that can be executed on CPU. Thus, the computational bottleneck is the number of model evaluations required by the underlying certificate, while the proposed proxy calibration itself adds negligible overhead beyond standard array operations. We highlight that we do not train or finetune any model and all the experiments are running on pre-trained models.

E.1 NLP Certificate

In Table 4, we use the edit-distance certificate from Huang et al. [2024], and provide a proxy certificate for r insertions. The smoothing scheme is to delete tokens with p_{del} probability. Here we provide theoretical support for the experiments and derivations.

Symmetry of the edit-distance ball. The Levenshtein (edit) distance satisfies the axioms of a metric, and in particular symmetry: $d_{\mathrm{edit}}(\boldsymbol{x}, \bar{\boldsymbol{x}}) = d_{\mathrm{edit}}(\bar{\boldsymbol{x}}, \boldsymbol{x})$ for all inputs $\boldsymbol{x}, \bar{\boldsymbol{x}}$. The edit-distance ball of radius r centred at $\boldsymbol{x}$ is

$$\mathcal{B}_r^{\mathrm{edit}}(\boldsymbol{x}) := \{\bar{\boldsymbol{x}} : d_{\mathrm{edit}}(\boldsymbol{x}, \bar{\boldsymbol{x}}) \leq r\}.$$

Symmetry of the distance immediately implies that $\bar{\boldsymbol{x}} \in \mathcal{B}_r^{\mathrm{edit}}(\boldsymbol{x})$ if and only if $\boldsymbol{x} \in \mathcal{B}_r^{\mathrm{edit}}(\bar{\boldsymbol{x}})$. In the notation of §2, this means $\left(\mathcal{B}^{\mathrm{edit}}\right)^{-1} = \mathcal{B}^{\mathrm{edit}}$: the edit-distance ball is its own inverse ball.

Table 10: Fallback ProxC on ImageNet/ResNet: proxy reject rate and average forwards per query. Speedups are relative to randomized smoothing with 1950 (certification-only) and 2000 (end-to-end) forwards.

		Certification-only		End-to-end	
r	Reject rate	Forwards	Speedup	Forwards	Speedup
0.00	24.2%	473	4.12×	523	3.82×
0.06	32.1%	627	3.11×	677	2.95×
0.12	38.6%	753	2.59×	803	2.49×
0.14	39.9%	779	2.50×	829	2.41×
0.18	45.7%	893	2.18×	943	2.12×
0.25	50.9%	992	1.97×	1042	1.92×
0.38	64.1%	1250	1.56×	1300	1.54×
0.50	75.0%	1462	1.33×	1512	1.32×

Table 11: False-approval risk and certified ratio of the proxy on ImageNet, stratified to the best and worst 100 classes.

r	Group	#points	Proxy risk	Proxy cert. ratio
0.12	top 100	4,577	0.0	0.5724
0.12	worst 100	10,654	0.0428	0.6555
0.25	top 100	7,023	0.0	0.5090
0.25	worst 100	8,025	0.0320	0.5499
0.50	top 100	12,229	0.0	0.2728
0.50	worst 100	2,613	0.0168	0.3750

Insertion and deletion balls are mutual inverses. The insertion-only and deletion-only sub-balls are not individually symmetric, but they are inverses of each other. If $\bar{\boldsymbol{x}}$ is reachable from $\boldsymbol{x}$ by r insertions, then $\boldsymbol{x}$ is reachable from $\bar{\boldsymbol{x}}$ by exactly r deletions, and vice versa. Writing $\mathcal{B}_r^{\mathrm{ins}}(\boldsymbol{x})$ and $\mathcal{B}_r^{\mathrm{del}}(\boldsymbol{x})$ for the sets of sequences reachable from $\boldsymbol{x}$ by at most r insertions and r deletions respectively,

$$\bar{\boldsymbol{x}} \in \mathcal{B}_r^{\mathrm{ins}}(\boldsymbol{x}) \iff \boldsymbol{x} \in \mathcal{B}_r^{\mathrm{del}}(\bar{\boldsymbol{x}}),$$

so that $\left(\mathcal{B}_r^{\mathrm{ins}}\right)^{-1} = \mathcal{B}_r^{\mathrm{del}}$ and $\left(\mathcal{B}_r^{\mathrm{del}}\right)^{-1} = \mathcal{B}_r^{\mathrm{ins}}$. Insertions and deletions are therefore dual operations under the inverse-ball map.

This is specifically important since we use inverse ball for single sample calibration in Prop. 2.

Upper and lower bounds under edit operations. Following Huang et al. [2024], the smoothed classifier admits explicit upper and lower bounds under edit-distance perturbations. Consider an input $\boldsymbol{x}$ and a perturbed input $\bar{\boldsymbol{x}}$ obtained using n_{del} deletions, n_{ins} insertions, and n_{sub} substitutions, with

$$n_{\mathrm{del}} + n_{\mathrm{ins}} + n_{\mathrm{sub}} = r.$$

Then, for any class y, the smoothed classifier score satisfies

$$p_{\mathrm{del}}^{n_{\mathrm{del}}-n_{\mathrm{ins}}}\left(p_y(\boldsymbol{x}) - 1 + p_{\mathrm{del}}^{n_{\mathrm{sub}}+n_{\mathrm{ins}}}\right) \;\le\; p_y(\bar{\boldsymbol{x}}) \;\le\; p_{\mathrm{del}}^{n_{\mathrm{del}}-n_{\mathrm{ins}}} p_y(\boldsymbol{x}) + 1 - p_{\mathrm{del}}^{n_{\mathrm{sub}}+n_{\mathrm{del}}}.$$

The certificate follows from comparing the lower bound on the predicted class score against the upper bound on the runner-up, analogously to the ℓ_2 certificate.

Under a fixed edit budget r, the conservative worst-case configuration is attained when all perturbations are substitutions: $n_{\mathrm{sub}} = r$ and $n_{\mathrm{del}} = n_{\mathrm{ins}} = 0$. Substituting into the general bounds yields the Levenshtein-distance certificate of Huang et al. [2024]:

$$p_y(\boldsymbol{x}) - 1 + p_{\mathrm{del}}^{r} \;\le\; p_y(\bar{\boldsymbol{x}}) \;\le\; p_y(\boldsymbol{x}) + 1 - p_{\mathrm{del}}^{r}.$$

Substitutions are the worst case because they combine the effect of both a shrinking lower bound and a growing upper bound simultaneously, without the compensating multiplicative factor that pure insertions or deletions introduce.

Per-operation certificates. The same framework yields tighter certificates when the threat model is restricted to a single operation type. For *substitution-only* perturbations ($n_{\mathrm{sub}} = r$, $n_{\mathrm{del}} = n_{\mathrm{ins}} = 0$):

$$p_y(\boldsymbol{x}) - 1 + p_{\mathrm{del}}^{r} \;\le\; p_y(\bar{\boldsymbol{x}}) \;\le\; p_y(\boldsymbol{x}) + 1 - p_{\mathrm{del}}^{r}.$$

Table 12: Per-class false-approval risk, certified ratio (CR), and risk conditional on approval (FA/appr = risk/CR) of the proxy on CIFAR-10, in % (setup of Table 13, $|\mathcal{D}_n| = 500$, averaged over 500 calibration splits). All classes have the same frequency (10%), so the marginal values (last row) are the averages over classes.

	$r = 0.12$ ($\alpha_r = 2.97$)			$r = 0.25$ ($\alpha_r = 1.60$)		
Class	Risk	CR	FA/appr	Risk	CR	FA/appr
airplane	3.27	59.9	5.5	2.25	44.9	5.0
automobile	3.37	67.7	5.0	2.59	54.7	4.7
bird	2.73	32.3	8.5	1.18	16.7	7.1
cat	2.75	23.0	12.0	0.82	7.9	10.4
deer	3.16	25.4	12.4	1.09	11.1	9.8
dog	2.08	35.0	6.0	0.74	18.4	4.0
frog	2.38	43.2	5.5	1.20	27.3	4.4
horse	2.27	55.9	4.1	1.34	43.1	3.1
ship	2.51	66.0	3.8	1.96	51.5	3.8
truck	3.46	62.1	5.6	2.63	48.2	5.5
marginal	2.80	47.0	5.95	1.58	32.4	4.88

Table 13: Effect of the calibration-set size $n = |\mathcal{D}_n|$ on CIFAR-10 (ResNet-110, $\sigma = 0.5$, $\alpha = 5\%$). Proxy risk and certified ratio (CR) in %, mean $\pm$ std over 500 random calibration splits; the test set is all images not used for calibration. Infeasible: $1/(n+1) > \alpha_r$, so the proxy approves nothing.

	$r = 0$ $\alpha_r = 5.00$, RS CR 81.3		$r = 0.12$ $\alpha_r = 2.97$, RS CR 71.1		$r = 0.25$ $\alpha_r = 1.60$, RS CR 61.1		$r = 0.5$ $\alpha_r = 0.41$, RS CR 44.4	
n	Risk	CR	Risk	CR	Risk	CR	Risk	CR
25	3.84 ± 3.53	58.9 ± 17.4	infeasible		infeasible		infeasible	
50	3.86 ± 2.67	61.9 ± 12.3	1.99 ± 1.92	38.9 ± 12.2	infeasible		infeasible	
100	5.02 ± 2.29	68.2 ± 8.8	2.91 ± 1.68	46.3 ± 8.4	0.99 ± 0.97	25.6 ± 8.6	infeasible	
200	4.89 ± 1.49	68.5 ± 5.7	2.43 ± 1.05	44.6 ± 5.7	1.53 ± 0.88	31.3 ± 5.7	infeasible	
500	5.00 ± 0.93	69.4 ± 3.4	2.80 ± 0.76	47.0 ± 3.9	1.58 ± 0.61	32.4 ± 3.7	0.41 ± 0.30	13.6 ± 3.5
1000	4.96 ± 0.74	69.3 ± 2.6	2.90 ± 0.57	47.7 ± 2.7	1.50 ± 0.41	32.0 ± 2.6	0.40 ± 0.22	13.9 ± 2.5
2000	5.00 ± 0.54	69.5 ± 1.7	2.93 ± 0.42	47.9 ± 1.9	1.55 ± 0.31	32.5 ± 1.8	0.41 ± 0.17	14.2 ± 1.7
5000	4.98 ± 0.45	69.5 ± 1.2	2.96 ± 0.34	48.1 ± 1.4	1.59 ± 0.26	32.8 ± 1.3	0.41 ± 0.13	14.3 ± 1.1

For *insertion-only* perturbations ($n_{\text{ins}} = r$, $n_{\text{del}} = n_{\text{sub}} = 0$):

$$p_{\text{del}}^{-r}(p_y(\boldsymbol{x}) - 1 + p_{\text{del}}^{r}) \;\le\; p_y(\bar{\boldsymbol{x}}) \;\le\; p_{\text{del}}^{-r}\, p_y(\boldsymbol{x}).$$

For *deletion-only* perturbations ($n_{\text{del}} = r$, $n_{\text{ins}} = n_{\text{sub}} = 0$):

$$p_{\text{del}}^{r}\, p_y(\boldsymbol{x}) \;\le\; p_y(\bar{\boldsymbol{x}}) \;\le\; p_{\text{del}}^{r}\, p_y(\boldsymbol{x}) + 1 - p_{\text{del}}^{r}.$$

These expressions reveal qualitatively different distortions: substitutions induce additive shifts, whereas insertions and deletions induce multiplicative rescalings, consistent with the inverse-ball duality established above.

For instance, to apply Prop. 2 on the insertion-only ball, we need to take the lower bound of the deletion-only ball, or the upper bound of the insertion ball itself which means that $\mathrm{c}^{\downarrow}(\alpha, \mathcal{B}^{-1}) = p_{\text{del}}^{r}\alpha$, and $\mathrm{c}^{\uparrow}(\alpha, \mathcal{B}^{-1}) = p_{\text{del}}^{-r}\alpha$. In our experiments the only ball that could be certified effectively was the insertion ball.

Insertions as a valid proxy threat model. The backbone certificate is derived over the full edit-distance ball $\mathcal{B}_r^{\text{edit}}(\boldsymbol{x})$, but the proxy in Table 4 is calibrated against r-insertion perturbations. This choice is valid by containment: any sequence reachable from $\boldsymbol{x}$ by r insertions has Levenshtein distance at most r from $\boldsymbol{x}$, so

$$\mathcal{B}_r^{\text{ins}}(\boldsymbol{x}) \;\subseteq\; \mathcal{B}_r^{\text{edit}}(\boldsymbol{x}).$$

A certificate valid over the full edit-distance ball therefore remains valid over the insertion-only sub-ball, and the proxy guarantee of Prop. 2 carries over without modification. Insertions are a natural and semantically meaningful subset of the edit-distance threat model, covering realistic perturbations such as token injection or paraphrase expansion, and their ball is a strict subset of the edit-distance ball certified by the backbone.

F Related Works

Certified robustness. The vulnerability of neural networks to adversarial perturbations has motivated a large literature on defenses and robustness verification. Purely empirical defenses are difficult to trust in isolation: e.g. Athalye et al. [2018] showed that defenses relying on gradient masking can be circumvented by adaptive attacks. Certified robustness replaces this empirical evaluation with a formal implication: if a point is certified, then the prediction is guaranteed to be constant within the specified perturbation set. One line of work obtains such guarantees by reasoning directly about the network, using satisfiability or mixed-integer formulations [Katz et al., 2017, Tjeng et al., 2019], convex outer relaxations [Wong and Kolter, 2018], and layer-wise linear bounds [Weng et al., 2018, Zhang et al., 2018]. These methods are sound and often tight for the architectures they support, but their cost and implementation dependence make them difficult to apply to large modern pipelines, or to black-box systems where the model internals are unavailable. Our work keeps the soundness of an existing certificate as a calibration signal, but changes its deployment role: instead of verifying every test point, the certificate is used to control the marginal false-approval rate of a cheaper proxy.

Randomized smoothing. Randomized smoothing is the most relevant certificate for our setting because it is model-agnostic and therefore applies to arbitrary classifiers through noisy inference. Cohen et al. [2019] proposed the standard Gaussian certificate for ℓ_2 robustness and showed that smoothing scales to ImageNet. Subsequent work broadened the certificate in several directions: Lee et al. [2019] derived tight certificates for broader randomized ensembles and discrete perturbations, Yang et al. [2020] developed a general recipe for smoothing with different perturbation geometries and smoothing distributions. Other work has improved the trained smoothed classifier itself, for example through adversarially trained smoothed classifiers [Salman et al., 2019]. Smoothing has also been used to certify quantities beyond the top-label robustness radius, such as predictive confidence [Kumar et al., 2020], and has been extended to structured outputs such as segmentation [Fischer et al., 2021]. Denoised smoothing prepends a denoiser before the classifier so that pretrained models can be certified without changing the classifier; for the denoiser we can typically use a diffusion model [Salman et al., 2020, Carlini et al., 2022]. In summary, so far randomized smoothing seems to scale on larger datasets, and models, and return better certified accuracy compared to verifications, and exact certificates. The main limitation shared by these approaches is the same one we target: the smoothed class probabilities have to be estimated by Monte Carlo sampling at calibration and test time, and finite-sample confidence corrections can require hundreds or thousands of forward passes per input. Recent work reduces this cost for the certificate itself through confidence sequences and early stopping [Voráček, 2024]. Such techniques are complementary to ours: they can reduce the cost of the backbone certificate, and hence both our calibration cost and the fallback cost, without changing the proxy calibration. Our framework is orthogonal to the choice of smoothing certificate; it asks how to use such certificates less often while still retaining a formal, distribution-free validity statement.

Relaxed and distributional robustness. Classical certification is pointwise: every certified input receives a local guarantee. This is the strongest statement, but also the most expensive one. Several works have considered relaxations that replace worst-case pointwise robustness with a distributional or probabilistic guarantee. Rice et al. [2021] studied robustness notions between average-case (random) and worst-case perturbation. In contrast, our work provides efficiency for the worst-case perturbation which is a different (and stronger) guarantee. We assume every test point is perturbed at its worst. Blohm et al. [2025] proposed probably approximately global (PAG) robustness certification, which also relaxes the soundness to a marginal guarantee. Their guarantee connects the confidence of the prediction with its robust radius – they propose a ρ-κ robustness guarantee, ensuring that with confidence higher than κ, with high probability x has a robust radius above ρ. This work is conceptually close to ours in that it also trades pointwise guarantee for a statistically meaningful marginal guarantee. The distinction is that we do not aim to certify the global robustness of the classifier itself, and we do not offer a guaranteed mapping between confidence and robust radii. While we use a simple thresholded proxy as our example, one can define more complex functions and adapt it to our framework for free; additionally, we aim to replace the expensive certificate, with a cheaper function and not to state a confidence-radii front for the classifier. Our proxy is allowed to accept many inputs cheaply, while its probability of accepting a genuinely non-robust input is controlled over the future data distribution. Intuitively, the distinction is analogous to the difference between conformal prediction and post-hoc calibration methods such as Platt scaling: PAG constructs

a distributional guarantee over robustness events, whereas our method calibrates a proxy certificate to imitate an expensive certificate under a controlled marginal false-approval risk.

Conformal prediction and conformal risk control. Conformal prediction provides distribution-free finite-sample guarantees under exchangeability [Vovk et al., 2005, Angelopoulos and Bates, 2021, Angelopoulos et al., 2024]. In classification, set-valued predictors with valid coverage were studied by Sadinle et al. [2018] and Romano et al. [2020], and the same idea has since been used as a general-purpose uncertainty quantification tool. Extensions beyond exact exchangeability, such as the weighted and approximate guarantees studied by Barber et al. [2022], are relevant whenever deployment differs from calibration. Conformal risk control (CRC) extends the conformal guarantee from miscoverage to general monotone risks [Angelopoulos et al., 2022]. Instead of requiring the loss to be the indicator of a missed label, CRC calibrates a parameter so that the expected value of a user-chosen risk is below a target level. This is the main statistical primitive in our paper: the event we need to control is not ordinary miscoverage, but false approval by a proxy certificate. Related risk-control ideas also appear in risk-controlling prediction sets [Bates et al., 2021], in the Learn then Test framework for finite-sample control of more general risks [Angelopoulos et al., 2025], and in selective prediction where a model learns when to abstain [Geifman and El-Yaniv, 2017, 2019]. Our use of CRC differs from standard selective classification because the controlled failure event is defined relative to adversarial robustness, and the test point may be a perturbation of an exchangeable clean point rather than itself exchangeable with the calibration data.

Robust conformal prediction. Standard conformal guarantees do not directly survive adversarial test-time perturbations, since the perturbed test input is no longer exchangeable with the clean calibration data. A common strategy is to replace the conformity score by a worst-case score over the perturbation set. Gendler et al. [2021] combine conformal prediction with randomized smoothing in this way, obtaining adversarially robust coverage under ℓ_2 perturbations. Ghosh et al. [2023] propose a probabilistic variant that controls coverage for most perturbations rather than the worst case, improving the efficiency-robustness trade-off. Zargarbashi et al. [2024] derive tighter robust conformal sets by bounding the worst-case change in conformity scores, covering both evasion and poisoning perturbations. Jeary et al. [2024] use verifiers to obtain robust conformal guarantees, and Massena et al. [2025] pursue efficient robust conformal prediction through Lipschitz-bounded networks. In a smoothing-based line, Yan et al. [2024] revisit randomized-smoothed conformal prediction, identifying practical validity issues and improving set efficiency. The closest recent lines are Zargarbashi et al. [2025], who show that robust conformal prediction can be obtained from a single randomly augmented sample by certifying the conformal procedure itself, and Zargarbashi and Bojchevski [2025], who use a single binary certificate for robust conformal prediction. Our work can be viewed as moving this single-sample idea from prediction sets to proxy certification. We do not construct a set that covers the true label; instead, we calibrate a proxy decision rule so that approving an unsafe perturbed input is rare. More precisely, robust conformal risk control [Angelopoulos et al., 2022] requires the worst-case loss over the perturbation set for every calibration point, which is itself a certification problem, and single-sample robust conformal prediction [Zargarbashi et al., 2025] controls the coverage of a prediction set. Relative to these, the new ingredients of our work are: (i) the anchored false-approval risk (Prop. 2), defined on the pair of received point and latent clean anchor, which lets a sound certificate evaluated only at clean calibration points serve as the calibration signal, while the guarantee holds at the perturbed test point; and (ii) adapting the single-sample argument from coverage to general bounded monotone risks, which requires the concavity of $\mathrm{c}^{\uparrow}$ (Lemma 1).

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly describe the proposed framework, the conformal risk control guarantee (bounded marginal false-approval rate), both the standalone and fallback certificate use cases, and the empirical scope (vision transformers and language models). Aspirational goals are distinguished from proven guarantees.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper includes a dedicated limitations paragraph (*Fundamental limitations* in § 3.4) that discusses the gap between marginal and instance-wise guarantees, the dependence on proxy quality, and the exchangeability assumption underlying the conformal risk control framework.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: All theoretical results are stated with explicit assumptions (e.g., exchangeability of the calibration set). Complete proofs are provided in the appendix, with proof sketches in the main text to convey intuition.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The calibration procedure, proxy tuning via conformal risk control, evaluation protocol, and all hyperparameter choices are described in sufficient detail in the paper and appendix. Code is also released at `https://github.com/soroushzargar/ProxC`.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The code is publicly available at `https://github.com/soroushzargar/ProxC`. All experiments rely on publicly available pretrained models and standard benchmarks.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper specifies the calibration set sizes, risk level α, proxy choices, and evaluation metrics. Full details are provided in the appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Variance is reported for key experiments. Variance is generally very small, which is expected given the theoretical guarantees of the conformal risk control framework, and this is noted in the paper.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Formal hardware specifications (GPU type: NVIDIA H100 and A100) are reported in § E.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research is foundational methodology for robustness certification and conforms with the NeurIPS Code of Ethics. No human subjects, sensitive data, or dual-use concerns are involved.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: This is foundational methodology research on robustness certification without a direct path to specific deployment applications. The work improves the efficiency of existing safety tools and does not introduce new capabilities with foreseeable negative societal impacts.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release new models or datasets. The released code implements a post-hoc certification method applied to existing publicly available models, posing no elevated misuse risk.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All pretrained models and datasets used in the experiments are cited with references to their original papers. Their usage is consistent with their intended research purposes.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: The code implementing the proposed proxy certificate framework is released at `https://github.com/soroushzargar/ProxC` with documentation sufficient to reproduce the experiments.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A]

    Justification: The paper does not involve human subjects research and requires no IRB review.

    Guidelines:

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

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs appear in the paper only as one of the evaluated classifier architectures (i.e., as objects to be certified), not as a component of the proposed method. LLMs were used solely for writing assistance and proofreading, which does not require declaration per NeurIPS policy.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.