Halt Fast! Early Stopping for Certified Robustness

Andrew C. Cullen
University of Melbourne
andrew.cullen@unimelb.edu.au

Paul Montague
DST Group, Adelaide

Benjamin I. P. Rubinstein
University of Melbourne

Abstract

Randomized Smoothing (RS) provides rigorous robustness guarantees for neural networks without architectural constraints, yet its adoption is limited by its computational costs. However much of this compute is wasted through over-sampling. To resolve this, we introduce **M**eta **E-V**alue (MEV) Certifications, which employ an adaptive streaming process with anytime statistical guarantees. Our framework establishes a controllable operating spectrum: in a *High-Fidelity* regime, meta-learned priors match 96–99% of a 10,000-sample certification's radius with up to $3\times$ fewer samples; in a *Maximum-Efficiency* regime, an adaptive efficient exit can deliver up to $11.6\times$ sample reductions while providing strong certification performance. This same framework can also be employed to dynamically allocate the query budget across samples depending upon resource constraints or sample-based triage. These advances demonstrate a realistic pathway towards high-throughput, scalable safety-critical deployments of certified robustness.

1 Introduction

For all of their transformative utility, neural networks remain notoriously sensitive to oftentimes semantically meaningless modifications (Szegedy et al., 2013). These modifications, now known as adversarial examples (Goodfellow et al., 2014; Madry et al., 2018), have triggered the realization that model decision boundaries often lack the semantic alignment required for safety-critical deployments.

While defenses have been proposed against these manipulations, they are fundamentally constrained within a technological arms race, where each new defense provides something new to attack (Goodfellow et al., 2014; Cullen et al., 2025). By contrast, Certified Robustness has emerged as a rigorous framework for mathematically guaranteeing that a model's prediction remains invariant within a defined neighborhood of an input (Lecuyer et al., 2019; Cohen et al., 2019; Cullen et al., 2022). In the case of classification systems, this formulation is typically defined in terms of a radius r such that for a model F we can guarantee that $F(x) = F(x')$ for all x' in the ball $B_p(x, r) = \{x' : \|x' - x\|_p \leq r\}$.

Among certifications, Randomised Smoothing (RS) is unique in that it can be applied to any model without architectural modifications. However, this flexibility comes at a cost—to complete a certification, a sample must be passed through the model tens of thousands of times, with each copy being offset by a small perturbation drawn from the Normal distribution. This high sample complexity, which is required to control for the Type-I (false positive) errors render real-time applications of certifications nigh-on impossible, especially for models at scale.

Recent advances have sought to alleviate this overhead through sequential testing and early stopping. Most notably, the introduction of **E-values** (Shafer & Vovk, 2019; Ramdas et al., 2023) and test martingales have allowed for anytime-valid certifications (Voráček, 2024). By framing certification as a super-martingale wealth process, sampling can be halted the moment sufficient evidence for robustness is accumulated without violating statistical safety. However, current E-value applications in robustness have primarily focused on binary hypothesis testing (e.g., asking if $r \geq c$ or not), reducing the certifier to a simple threshold-based classifier. This limitation robs the process of the discriminatory detail required to assess relative security across different samples.

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

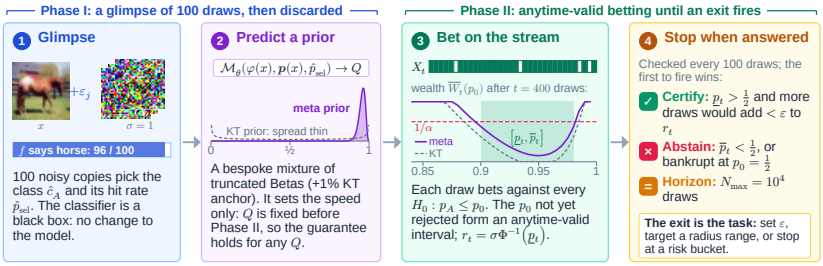


Figure 1: **Meta-RS Certification Architecture**. Phase I uses a strictly discarded 100-draw glimpse to predict an $\mathcal{F}_0$-measurable prior via $\mathcal{M}_\theta$ (Theorem 2). Phase II streams GPU batches ($B = 100$) with $< 100\,\mu$s CPU bound updates, terminating adaptively into fast rejection (5.5–10.7×), high-fidelity (1.08–2.58×), or plateau certification (7.8–11.6× speedup).

We suggest that this reduction divorces certifications from any potential real-world utility. The barrier to certifications being employed is not a product of certification size, but utility (Cullen et al., 2024a, 2026a). Deployments need the ability to rank samples by perceived risk (to support input triage); high throughput performance with rapid rejection of uncertifiable inputs; and the ability to perform dynamic resource triage, where samples that are producing impactful outputs are subject to greater scrutiny. A priori defined sampling cannot achieve any of these, and current anytime-valid certifications can only achieve the second.

Our MEV's anytime validity is not a universal precision amplifier, but rather is the adaptive resource routing engine required to resolve the operational limitations of fixed-horizon testing. MEV empowers practitioners to dynamically allocate compute to inputs and operational regimes that matter the most (Figure 1). As such, in this work we explore both the efficiency of MEV certifications, and their ability to adapt to application-based workflows, and in doing so introduce three key contributions:

1. **Method of Mixtures for Continuous Hypotheses:** We extend E-value certifications to a mixture-based multiple-hypothesis approach to match traditional certification workflows.
2. **Sample-Adaptive Meta-Learning:** We introduce an optimized E-value formulation, where a meta-learner predicts the prior distribution to enhance efficiency. We employ a lightweight meta-learner to analyze initial model glimpses, utilizing a Bayesian Negative Log-Likelihood (NLL) objective to fit the distribution of the smoothed model's success rate for a given input.
3. **Dual-Bankruptcy and Dynamic Rejection Exits:** We introduce task-adaptive termination conditions, that allow for early-termination based upon prespecified domain tasks, producing highly efficient certifications and rejections in a manner that optimizes global compute.

By improving efficiency, allowing for dynamic resource triage, and supporting streaming data, our approach provides the foundations for transforming certifications from a numerically costly concept to a viable framework for improving real-world security.

2 Related Work

Current state-of-the-art RS builds upon the Neyman-Pearson lemma (Cohen et al., 2019; Cullen et al., 2024b,a), providing tighter certificates than earlier Differential Privacy formulations (Lecuyer et al., 2019; Dwork et al., 2006). Mechanistically, RS transforms a base model f into a smoothed classifier g guaranteeing no class changes can occur within an ℓ_2 radius $r = \sigma\Phi^{-1}(p_A)$, where $p_A = \mathbb{P}_{\epsilon\sim\mathcal{N}(0,\sigma^2 I)}[f(x+\epsilon) = c_A]$ is the top-class probability. In practice, Phase I uses a small glimpse to select c_A, while Phase II bounds p_A from below via a Clopper-Pearson interval (Clopper & Pearson, 1934) $\underline{p_A}$. Tight bounds demand tens of thousands of Phase II draws set *a priori* to preserve Type-I error control and avoid the peeking problem (Johari et al., 2017)—massively over-sampling easy inputs while failing to certify marginal ones.

Sequential frequentist tests attempt to partition α over pre-specified checkpoints $\{n_1, \ldots, n_k\}$ via Bonferroni corrections or monotone alpha-spending functions (Lan & DeMets, 1983; Horváth et al., 2022), halting without violating Type-I error bounds. However, dividing α dilutes statistical power at each check, often requiring more samples than fixed-horizon RS; input-adaptive sample heuristics remain similarly conservative (Chen et al., 2022). Alternatively, E-values avoid the peeking problem entirely, enabling anytime-valid early stopping under arbitrary data-dependent criteria.

Voráček (2024) applied E-values to binary threshold tests ($r \geq r_0$) using the Krichevsky–Trofimov (KT) estimator (Krichevsky & Trofimov, 1981); however, KT is designed for *arbitrary* sequences, whereas RS streams reflect structured input manifolds. We therefore combine meta-learned prior prediction with the **Method of Mixtures** (Waudby-Smith & Ramdas, 2024; Grünwald et al., 2020) for continuous radius estimation, significantly accelerating anytime convergence (Appendix A). This work sits alongisde our contemperaneous work on certifying with E-values for sequence-to-sequence tasks (Cullen et al., 2026b).

3 Anytime-Valid Radius Certification

MEV reformulates RS as an **adaptive betting game** against a null hypothesis of non-robustness $H_0 : p \leq p_0$. To provide initial intuition, imagine a certifier betting against a bookmaker on the outcome of streaming noise perturbations $X_1, X_2, \ldots$. Phase I of the process involves taking a 100-sample glimpse of the input x. A lightweight Meta-Learner analyzes this glimpse (and, optionally, the feature representation $\phi(x)$) to predict a bespoke prior $Q(\theta)$ over success probabilities.

This prior is then employed to inform Phase II, in which the certifier accumulates wealth W_t under a test martingale as noise perturbations stream in. When the predicted prior correctly concentrates probability mass near the true success rate p, wealth grows exponentially fast, which is why our Phase I prediction of $Q(\theta)$ improves utility so much, relative to a worst-case KT prior. This accumulated wealth translates into an anytime-valid lower confidence bound $\underline{p_t}$, where higher wealth corresponds to tighter bounds. This wealth can be used to stop sampling if wealth either crosses the rejection threshold $1/\alpha$ (proving robustness) or falls below a bankruptcy threshold ϵ_{fail}. The latter of these indicates a lack of evidence for robustness, at which point the sample is conservatively rejected to preserve compute. This game-theoretic formulation ensures that statistical safety is governed strictly by probability theory, while computational efficiency is maximized by machine learning.

To evaluate the robustness of a model prediction of a class c_A at a point x, we consider the success probability $p = P(f(x + \epsilon) = c_A)$ of the smoothed classifier (we stress that we presuppose knowledge of c_A for mathematical convenience, and that the class is estimated appropriately in our algorithm). Assume that we observe an infinite sequence of i.i.d. Bernoulli trials $X_1, X_2, \ldots$, where each $X_i = \mathbb{I}[f(x + \epsilon_i) = c_A]$ denotes whether the i-th perturbation of x with noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2 I)$ aligns with the target class. To construct a certification, we must be able to test the null hypothesis $H_0 : p \leq p_0$ for any threshold $p_0 \in [0, 1]$ to construct an anytime-valid lower confidence bound on p.

Test Martingales and E-values We achieve this test by exploiting *E-values*, a hypothesis testing framework where a non-negative random variable E satisfies $\mathbb{E}_{H_0}[E] \leq 1$ (Vovk & Wang, 2021; Shafer, 2019). Within the context of our Bernoulli setting, the E-value is the likelihood ratio $E_i = q^{X_i}(1-q)^{1-X_i}/[p_0^{X_i}(1-p_0)^{1-X_i}]$ for a point null $H_0 : p = p_0$ and point alternative $H_1 : p = q$ (Waudby-Smith & Ramdas, 2024). Inspired by betting games (Shafer, 2019; Vovk & Wang, 2021), accumulating E-values over t draws with $h_t = \sum_{i=1}^{t} X_i$ successes yields the wealth process $W_t(p_0) = \prod_{i=1}^{t} E_i = q^{h_t}(1-q)^{t-h_t}/[p_0^{h_t}(1-p_0)^{t-h_t}]$, representing the sequence likelihood ratio between the alternative and null hypotheses.

The process $(W_t(p_0))_{t\geq 1}$ is a non-negative martingale with $\mathbb{E}[W_t] = 1$ under $p = p_0$. By Ville's inequality (Ville, 1939; Doob, 1940), the set of all p_0 for which the wealth has not yet crossed the rejection threshold $1/\alpha$ forms the confidence interval $C_t = \left\{p_0 \in [0, 1] : \max_{\tau \leq t} W_\tau(p_0) < \frac{1}{\alpha}\right\}$ where the Lower Confidence Bound (LCB) is $\underline{p_t} = \inf C_t$. Here $W_t^{\max}(p_0) \triangleq \max_{\tau \leq t} \bar{W}_\tau(p_0)$ denotes the running maximum wealth, so that equivalently $C_t = \left\{p_0 \in [0, 1] : W_t^{\max}(p_0) < \frac{1}{\alpha}\right\}$.

Theorem 1 (Soundness)**.** *For a target class c_A, significance level $\alpha \in (0, 1)$, and prior mixture Q, the lower confidence bound $\underline{p_t} = \inf\{p_0 : \max_{\tau \leq t} \overline{W}_\tau(p_0) < 1/\alpha\}$ satisfies $P(\exists t : p < \underline{p_t}) \leq \alpha$.*

This result follows from the *test martingale inversion* principle (Howard et al., 2021; Ramdas et al., 2023). For any fixed $p_0 \in [0, 1]$, the mixture wealth process $\bar{W}_t(p_0)$ is a non-negative martingale with $\bar{W}_0 = 1$ under the point null hypothesis $H_0 : p = p_0$. By Ville's inequality, the probability that the wealth ever exceeds $1/\alpha$ is bounded by α. Since the confidence set C_t is constructed by inverting this test, and by utilizing the running maximum of the wealth process, we ensure that the confidence bounds monotonically tighten over time, guaranteeing that C_t contains the true success probability p for all $t \geq 1$ with probability at least $1 - \alpha$.

As the mixture of E-values is log-convex with respect to p_0, the non-rejected hypotheses form a single contiguous interval $[\underline{p}_t, \bar{p}_t]$ (proof in Appendix B). This ensures that evaluating the anytime lower bound $\underline{p}_t = \inf C_t$ is a unimodal 1D root-finding problem solvable via Brent's method in $< 100\,\mu$s, introducing negligible computational overhead. Appendix E.1 confirms this soundness empirically, finding zero violations beyond reference noise. We note that no multiple-testing correction or union bound over the continuum of hypotheses p_0 is required. Because C_t is obtained by test inversion, coverage only requires that the single true value p is never rejected, an event whose probability is bounded by α by applying Ville's inequality to the one process $\bar{W}_t(p)$.

The Method of Mixtures In the context of certifications, we do not know *a priori* what the alternative q is, and yet knowing this on a sample-by-sample basis is crucially important for providing enough discriminatory information to support downstream applications. As such, we instead propose utilizing the *Method of Mixtures* (Grünwald et al., 2020; Waudby-Smith & Ramdas, 2024), which tests against the set of hypotheses defined over a prior distribution $Q(q)$ of alternative hypotheses, by way of the mixture E-value

$$\bar{W}_t(p_0) = \int_0^1 \frac{q^{h_t}(1-q)^{t-h_t}}{p_0^{h_t}(1-p_0)^{t-h_t}} dQ(q) \qquad = \frac{m(h_t, t)}{p_0^{h_t}(1-p_0)^{t-h_t}} \ , \tag{1}$$

where $m(h,t) = \int_0^1 q^h(1-q)^{t-h} dQ(q)$ is the integrated likelihood.

Lemma 1 (Mixture E-values)**.** *For any probability measure Q on $[0,1]$, the mixture $\bar{W}_t(p_0)$ is a non-negative martingale under $H_0 : p = p_0$, and thus an anytime-valid E-process.*

When Q is a Beta distribution $\text{Beta}(\beta, \gamma)$, the integrated likelihood has the closed form $m(h,t) = B(h+\beta, t-h+\gamma)/B(\beta,\gamma)$ (Lai, 1976), allowing numerical integration to be eschewed. While numerically convenient, a single Beta distribution is restricted to unimodal (or U-shaped bimodal) densities, which limits applicability given that real-world success probabilities often cluster into heterogeneous modes. However, we exploit the fact that any convex combination $E = \sum_{k=1}^K w_k E^{(k)}$ of E-values with $w_k \geq 0, \sum_{k=1}^K w_k = 1$ remains a valid E-value under H_0 by linearity of expectation (Vovk & Wang, 2021; Grünwald et al., 2020; Waudby-Smith & Ramdas, 2024) (proof in Appendix B) to extend Q to a K-component Beta mixture with total wealth

$$\overline{W}_t(p_0) = \sum_{k=1}^K w_k \cdot \left(\frac{m_k(h_t, t)}{p_0^{h_t}(1-p_0)^{t-h_t}} \right) \ . \tag{2}$$

To demonstrate that the Meta-Learner affects only search efficiency through the wealth accumulation velocity, and never influences statistical validity, we introduce the following Theorem.

Theorem 2 (Safety-Efficiency Separation)**.** *For any probability measure Q on $[0,1]$ that is fixed prior to Phase II (i.e., $\mathcal{F}_0$-measurable, see Section 4) and any arbitrary stopping time τ (data-dependent, heuristic, or fixed), the lower confidence bound $\underline{p}_\tau$ satisfies $P(p < \underline{p}_\tau) \leq \alpha$ unconditionally.*

Because Ville's maximal inequality uniformly bounds the running wealth by the trajectory supremum $\sup_{t \geq 1} \bar{W}_t(p)$, any stopping policy τ inherits the unconditional $(1-\alpha)$ coverage guarantee without requiring regularity conditions on τ (full proof in Appendix B). This cleanly separates statistical safety from sampling efficiency: practitioners can deploy arbitrary stopping heuristics—from high-fidelity look-ahead rules matching asymptotic budgets to aggressive plateau exits terminating when marginal radius gains diminish—without compromising certification validity.

4 Sample-Adaptive Meta-Learning

The efficacy of the Mixture E-value process depends on how $Q(q)$ allocates probability mass across alternatives to maximize wealth accumulation. Standard approaches use the KT estimator (Krichevsky & Trofimov, 1981) $\text{Beta}(0.5, 0.5)$, which guarantees minimax regret for worst-case arbitrary sequences (Xie & Barron, 2000). Yet certification inputs are **not arbitrary**: they lie on a data distribution $\mathcal{D}$ and are accompanied by the N_{sel}-sample Phase I glimpse, offering a strong information advantage over the conservative KT prior (Waudby-Smith & Ramdas, 2024; Grünwald et al., 2020).

Drawing upon this observation, we introduce the Sample-Adaptive Meta-Learner $\mathcal{M}_\theta$ to perform amortized Bayesian inference (Gershman & Goodman, 2014), leveraging Phase I information to

predict a bespoke prior $Q(\theta) = \mathcal{M}_\theta(\phi(x), \mathbf{p}(x), \hat{p}_{\text{sel}})$ that is then employed to support Phase II betting. Because the N_{sel} Phase I glimpse draws are *strictly discarded* before wealth accumulation begins, letting $\mathcal{F}_0 = \sigma\big(x, \epsilon_1^{\text{sel}}, \ldots, \epsilon_{N_{\text{sel}}}^{\text{sel}}\big)$ and $\mathcal{F}_t = \sigma(\mathcal{F}_0, X_1, \ldots, X_t)$ ensures that both c_A and $Q(\theta)$ are $\mathcal{F}_0$-measurable, while independent Phase II draws $X_i \mid \mathcal{F}_0 \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(p)$ keep $(\bar{W}_t(p_0))_{t\geq 1}$ a valid $\mathcal{F}_t$-supermartingale (Shafer & Vovk, 2019; Ramdas et al., 2023), leaving Theorem 1 untainted by data-dependency.

Our meta-learner ingests three distinct signals: the semantic context $\phi(x)$, representing the penultimate layer embedding of the base classifier for the clean image, providing a high-dimensional representation of the image's difficulty and class; the mean softmax vector $\mathbf{p}(x)$ over the N_{sel} Phase I draws, representing the classifier's confidence under noise; and the empirical success rate $\hat{p}_{sel}$ observed over the glimpse of N_{sel} samples.

Of these, the latter provides the most direct evidence of the true success probability p. Based upon our experiments, we extract scalar proxies from the classifier confidence $\mathbf{p}(x)$ vector so that it returns either the margin $\mathbf{p}_{c_A}(x) - \max_{c \neq c_A} \mathbf{p}_c(x)$, taken with respect to the class c_A selected in Phase I, or the entropy, acting as proxies for the model's epistemic uncertainty. We take $\mathbf{p}(x)$ from the Phase I draws rather than from the clean image as the clean image's top-1 class differs from c_A for 29–62% of CIFAR-10 images ($\sigma = 0.25$–1). The embedding is optional: in our ablation (ImageNet, $\sigma = 0.5$; Table 16), dropping $\phi(x)$ leaves the samples and the average certified radius unchanged (5,897 samples against 5,935), while dropping $\hat{p}_{sel}$ as well costs 1.3% more samples than the full input set. $\mathcal{M}_\theta$ can thus be trained through a Kelly Criterion based loss, which is equivalent to minimizing the Negative Log-Likelihood (NLL) of the binomial sequence $X_{1:N}$ under the predicted mixture of Beta distributions by

$$\mathcal{L}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}}\left[\log \sum_{k=1}^{K} w_k \frac{B(h_N + \beta_k, N - h_N + \gamma_k)}{B(\beta_k, \gamma_k)}\right] + \lambda_t \, \mathbb{E}_{x\sim\mathcal{D}}\left[\sum_{k=1}^{K} \text{dist}(\hat{p}_{mle}, \mathcal{R}_k)\right], \quad (3)$$

where $(\beta_k, \gamma_k, w_k) = \mathcal{M}_\theta(\text{features})$, and λ_t controls a regularizing penalty that minimizes the ℓ_1 distance between $\hat{p}_{mle}$ and the nearest boundary of $\mathcal{R}_k$, where λ_t starts at 10.0 and linearly decays to 1.0 over the first 80% of training epochs. This formalism ensures that the expected growth rate of the wealth process is maximized (Kelly, 1956), while penalizing over-confident predictions as the log-wealth approaches $-\infty$ when the prior mass is zero at the true p. Consequently, the meta-learner learns to output broader, more resilient priors for high-variance inputs while maintaining aggressive, concentrated bets for unambiguous samples so that it safely maximizes wealth accumulation. As the Phase I glimpse is inherently noisy for every epacoh each training example uses one of the image's cached N_{sel}-draw blocks, chosen at random, as its Phase I realisation (with its own selected class c_A, glimpse $\hat{p}_{sel}$ and mean softmax $\mathbf{p}(x)$), and the count of c_A among the image's other draws as its target. This forces the meta-learner to learn a mapping $(\text{features}, \text{glimpse}) \to \text{Prior}$ that is robust to the variance of the initial sampling. To ensure the anytime-valid responsiveness of the mixture, we constrain the predicted Beta parameters to $\beta, \gamma \in [0.1, 500.0]$. The lower bound of 0.1 allows the meta-learner to predict distributions that are sharper than the non-informative KT prior, while the upper bound of 500 prevents a single sampling anomaly from triggering wealth bankruptcy.

The meta-learner in our experiments is an MLP with two hidden 512-unit layers (Appendix C) which maps these features to a K-component mixture prior. For each component k, the model predicts a weight w_k, Beta parameters (β_k, γ_k), and—as will be discussed in Section 4.1—the boundaries of the support region $\mathcal{R}_k$. However depth provides little impact, with 0, 1 or 4 hidden layers instead of 2, the samples stay within 0.6% of the two-layer model's, at the same average certified radius (Table 17). Our formalism provides significant flexibility in how the meta-learner is constructed. Across our experiments, we vary the mixture count K alongside three support modes: (i) **Full Support**, defined over $p \in [0, 1]$; (ii) **Hybrid Support**, a fixed partition pre-assigning components to robust $[0.5, 1.0]$ or non-robust $[0.0, 0.5]$ regions; and (iii) **Dynamic Support**, which learns sub-intervals $[a_k, b_k]$ concentrating betting resolution where the true p is predicted to lie. From this point on, we employ the naming scheme **Meta-{K}-{Support}**.

4.1 Truncated Betas

To realize the partitioned support modes introduced in the preceding section, we formalize the prior Q as a mixture of truncated Beta distributions. By restricting the probability mass of each

mixture component k to the specific (and potentially learned) interval $\mathcal{R}_k = [a_k, b_k]$, the meta-learner allocates probability mass to capture potential multi-modal distributions.

A truncated Beta prior is equivalent to a standard Beta distribution that is rejection-sampled to lie within $[a_k, b_k]$, subject to normalization by the mass of the original distribution contained within the region $Z_k = I_{b_k}(\beta_k, \gamma_k) - I_{a_k}(\beta_k, \gamma_k)$, where I_x is the regularized incomplete Beta function (the CDF of the Beta distribution). Under this prior, the integrated likelihood $m_k(h_t, t; \mathcal{R}_k)$—the marginal likelihood of the data given the regional prior Q_k—admits the closed form

$$m_k(h_t, t; \mathcal{R}_k) = \frac{B(h_t + \beta_k, t - h_t + \gamma_k)}{B(\beta_k, \gamma_k)} \cdot \frac{I_{b_k}(h_t + \beta_k, t - h_t + \gamma_k) - I_{a_k}(h_t + \beta_k, t - h_t + \gamma_k)}{Z_k} \quad , \tag{4}$$

combining the standard Beta update with the truncation correction (updating Equation 3 as in Algorithm 1). This yields localized likelihood ratios $E_k(p_0) = m_k(h_t, t; \mathcal{R}_k)/[p_0^{h_t}(1-p_0)^{t-h_t}]$ and meta-mixture wealth $W_t^{meta}(p_0) = \sum_{k=1}^{K} w_k E_k(p_0)$.

Rejection Martingales and Bankruptcy Exits The truncated beta framework enables both heuristic and mathematically rigorous exit strategies. Under the Hybrid Support mode, we concentrate a subset of mixture heads on the robust interval $[0.5, 1.0]$. If the true success probability p is significantly below 0.5, the likelihood ratios for these robust components will exponentially decay. At every check interval B, we monitor if the total wealth $W_t(0.5)$ falls below a failure threshold $\epsilon_{fail} = 0.1$—triggering what we dub a bankruptcy exit. If this occurs, the certifier immediately halts and rejects the sample. While these rejections may result in false negatives for samples near the decision boundary, such exits act as a high-speed rejection filter that significantly enhances global compute efficiency. This design decision reflects the operational reality of certifications—small certifications convey limited value, and conservative rejections on the decision boundary will not change how these samples are treated by any downstream process. In practice we require $t \geq 400$ Phase II draws and $\hat{p}_{mle} < 0.5$ before triggering early rejection, and concurrently monitor a fixed KT *reference bankruptcy* bettor $E_{KT}(0.5) \leq \epsilon_{fail}$ so borderline priors concentrated near $p = 0.5$ cannot stall without compromising Theorem 1 safety (formalized in Section 5; ablations in Appendix D.1).

The Safety Anchor: Guaranteeing Survival To guard against prior misspecification, we blend $W_t^{meta}(p_0)$ with a full-support Beta$(0.5, 0.5)$ KT safety anchor $E_{safety}(p_0) = m_{KT}(h_t, t)/[p_0^{h_t}(1-p_0)^{t-h_t}]$ at fixed weight $w_{safety} = 0.01$:

$$\bar{W}_t(p_0) = w_{safety} E_{safety}(p_0) + (1 - w_{safety}) W_t^{meta}(p_0) \quad . \tag{5}$$

As a convex combination of mixture E-values, $\bar{W}_t(p_0)$ directly inherits Theorem 1, adding $< 0.5\%$ overhead under native priors while guaranteeing convergence under severe out-of-distribution shift (Appendix F.2, Tables 17 and 19).

Practical Deployment Paradigms Training $\mathcal{M}_\theta$ itself is fast and requires limited computational resources ($<$ 2 minutes on a single CPU core), and generating its training predictions requires negligible compute (0.35% of base classifier training for ImageNet; see Appendix F.2 and Table 18). Our framework supports four operational paradigms: (i) *Post-Training* on a small held-out set of noisy predictions; (ii) *Inference-Time Adaptation* by logging Phase I glimpses offline; (iii) *Training-Free Static KT Priors* (Q_{KT}) requiring zero meta-learning while still outperforming fixed-horizon baselines; and (iv) *Zero-Shot Transfer*, where a meta-learner trained on a simple domain (e.g., MNIST) deploys directly to complex target domains (e.g., ImageNet) with minimal efficiency penalty (detailed in Appendix F).

5 The Certification Engine: Termination Policies

While Theorem 2 guarantees that our MEV's game-theoretic betting framework ensures unconditional statistical safety no matter where the system is stopped, the practical benefits of our technique require appropriately defining said stopping condition. At each check interval B (in our experiments $B = 100$ exclusively), the mixture wealth process can be inverted using the Brent-Dekker root finding method (Brent, 1971). This allows for the lower confidence bound $\underline{p_t} = \inf C_t$ to be evaluated, leading to the corresponding certified radius $r_t = \sigma\Phi^{-1}(\underline{p_t})$. However, this still raises the question: what observations of $\underline{p_t}$ should motivate stopping?

This cannot simply be the point where the confidence bounds stop contracting, as they contract at $\mathcal{O}(1/\sqrt{t})$. Instead, the terminations should be designed around the marginal return on compute: terminating when the anticipated radius gain is no longer worth the cost of additional sampling under noise. Because the certified radius scales directly with the perturbation scale σ, we define the stopping tolerance in units of noise ($\epsilon_t\sigma$), where the dimensionless factor ϵ_t represents a fraction of one standard deviation of smoothing noise. This ensures that stopping dynamics remain scale-invariant across σ.

To achieve this, we schedule ϵ_t via a linear decay over the sampling horizon N_{max}, where

$$\epsilon_t = \epsilon_{start} - (\epsilon_{start} - \epsilon_{end}) \cdot \min\left(1, \frac{t}{N_{max}}\right) \quad . \tag{6}$$

On inputs with strong class confidence, where $p_A \gg 0.5$, this bound clears 0.5 rapidly, allowing early termination. However for borderline inputs, sampling proceeds until ϵ_t anneals to ϵ_{end}, allowing the majority of the certification budget to be assigned to maximizing the radii associated with these marginal samples. In our work we set $\epsilon_{end} \in \{0.0126, \ldots, 0.4032\}$ to control the asymptotic tolerance, and $\epsilon_{start} = 2.38\,\epsilon_{end}$ to control relaxed exits. However, it must be stressed that these parameters, and even this schedule, can be changed under our MEV framework to support downstream application-focused tuning.

To support distinct operational priorities, we formulate two complementary certification policies gated by $\underline{p_t} > 0.5$. First, our **high-fidelity look-ahead policy** projects the achievable radius at N_{max} assuming remaining draws follow $\hat{p} = (h_t{+}0.5)/(t{+}1.0)$; letting $\hat{h}_{N_{max}} = h_t{+}\text{round}(\hat{p}(N_{max}{-}t))$, it halts when $r_{N_{max}}(\hat{h}_{N_{max}}) - r_t \leq \epsilon_t\sigma$. Tracking this asymptotic headroom prevents premature exit while the certificate is expanding, matching 96–99% of Cohen-10k's average certified radius with 1.08–$2.58\times$ fewer samples. Second, for resource-constrained or low-latency deployments, our **maximum-efficiency plateau policy** monitors marginal radius velocity over a lookback window of $L = 4$ checks (400 draws), halting when $r_t - r_{t-L\cdot B} \leq \epsilon_t\sigma$ to achieve 7.8–$11.6\times$ sample savings over Cohen-10k while dynamically matching Cohen-1k quality.

It is crucial to note that neither of these cases prohibits sampling in the non-robust region ($p_A \leq 0.5$), where further draws cannot yield a certificate. We therefore safely abort uncertifiable inputs once $t \geq 400$ Phase II draws where $\hat{p}_{mle} < 0.5$, if any of the following three rejection conditions fire: (1) **Upper Confidence Bound Exit**, where $\bar{W}_t(0.5) \geq 1/\alpha$ implies $\bar{p}_t < 0.5$, certifying non-robustness; (2) **Native Bankruptcy**, where $\bar{W}_t(0.5) \leq \epsilon_{fail} = 0.1$ halts due to insufficient evidence; or (3) **Reference Bankruptcy Exit**, where a concurrent KT bettor $E_{KT}(0.5) = m_{KT}(h_t, t)/0.5^t \leq \epsilon_{fail}$ prevents borderline priors ($p \approx 0.5$) from stalling, saving $4.5\times$ samples on boundary inputs. By Theorem 2, every policy—Look-Ahead, Plateau, or Early Rejection—is an anytime stopping time τ unconditionally preserving $(1 - \alpha)$ coverage.

Algorithm Algorithms 1 and 2 (Appendix C) demonstrate our training and sequential certification procedures, integrating MEV's prior initialization with anytime-valid monitoring to dynamically trade certification tightness against compute per input. Code to reproduce this work can be found at `https://github.com/andrew-cullen/Halt-Fast-Early-Stopping-for-Certified-Robustness`.

6 Results and Discussion

Why should we favor an anytime-valid approach over a fixed-horizon Clopper-Pearson bound, which is inherently tighter at a fixed N? The answer lies in viewing anytime validity not as a universal precision amplifier, but as an adaptive resource routing engine. We would argue that real-world adoption of RS is not changed by marginally improving certified radii. Instead, deployed models could significantly benefit from adaptive risk triage within a high-throughput certification framework.

To consider the operational extrema of the high-fidelity regime, and the maximum-efficiency regime, Table 1 presents results over the evaluation split (with results calculated as the median over 10 independent runs). Under the high-fidelity regime, Table 1 demonstrates that the global champion Meta-10-Margin-Full at $\epsilon = 0.0126\sigma$ reaches between 96 and 99% of the average certified radius (ACR) of Cohen-10k within 0.7 pp of its certified accuracy, while using between 1.08 and $2.58\times$ fewer samples. Our MEV approach requires fewer samples than the KT prior under the same stopping rule for all settings, while also producing higher ACRs. When compared to a fixed-N Cohen baseline tuned to achieve the exact same ACR, sequential stopping requires between 0.74 and $0.88\times$ the

Table 1: **Operational Certification Regimes** (**Meta-10-Margin-Full**, 10-seed median, eval split) vs. Cohen-10k (10,100 draws). *High-Fidelity* ($\epsilon = 0.0126\sigma$, look-ahead) matches 96–99% of Cohen-10k ACR; *Maximum-Efficiency* ($\epsilon_{start} = 0.1, \epsilon_{end} = 0.042, \Delta = 1.2$, plateau) yields 7.8–11.6× fewer draws at Cohen-1k quality (Appendix E).

		Cohen-10k	**High-Fidelity** ($\epsilon = 0.0126\sigma$)				**Max-Efficiency** (Plateau Exit)			
Dataset	σ	ACR	Samples	Fewer	ACR (%)	ΔAcc	Samples	Fewer	ACR	Acc (%)
MNIST	0.25	0.775	9,380	1.08×	99.0	+0.0	1,165	8.7×	0.605	99.6
	0.5	1.390	8,058	1.25×	98.0	+0.0	1,117	9.0×	1.113	99.4
	1	1.198	5,064	1.99×	96.7	-0.4	1,069	9.4×	1.016	91.7
CIFAR-10	0.25	0.396	6,177	1.64×	97.5	-0.3	873	11.6×	0.306	77.5
	0.5	0.507	5,033	2.01×	96.8	-0.5	1,035	9.8×	0.411	65.8
	1	0.511	3,922	2.58×	96.0	-0.7	1,291	7.8×	0.424	47.2
ImageNet	0.25	0.448	7,055	1.43×	98.2	-0.1	928	10.9×	0.348	69.3
	0.5	0.690	5,935	1.70×	97.8	-0.2	1,063	9.5×	0.550	61.4
	1	0.864	4,570	2.21×	97.2	-0.1	1,155	8.7×	0.699	46.7

samples, although as the dataset complexity decreases the results are more mixed (see Table 4). These results are unsurprising—even a prior concentrated at the true p certifies at $\sqrt{2\ln(1/\alpha)} = 3.72$ standard errors versus Clopper-Pearson's 3.09, which drives the increased computational cost of anytime validity. However adaptive stopping recovers this where there are significant divergences in model outputs, with the computational advantages of ImageNet driven by the 31–53% abstention rate, which is significantly different to the 93% of MNIST samples that elicit high confidence outputs.

Under the maximum efficiency, plateau exit regime of Table 1, our results demonstrate how the MEV approach trades radius for computational efficiency, with between 7.8 and 11.6× fewer samples required for certifications, relative to Cohen-10k. These results occur while reaching between 93–100% of Cohen-1k's ACR at 0.79–1.17× of its samples. However, unlike Cohen-1k, which rigidly commits all of its draws to every single input, the plateau exit is input-adaptive, with easier inputs stopping earlier, while borderline inputs draw further. The Pareto frontier of Figure 13 and the waterfall ablation within Figure 12 illustrate how users can smoothly interpolate between the high-fidelity regime and the maximum efficiency regime, in a domain specific manner.

These savings translate into wall-clock savings of up to 1.89× for MNIST, 2.53× for CIFAR-10, and 2.23× for ImageNet (See Appendix C.2 for further details). We note that the single-input latency on small models (like ResNet-18) is diminished by our current implementation's GPU kernel launch overheads, which vanishes as model size increases. ResNet-50 is between 1.26 and 1.95× faster, a speedup that we would expect to see increase with model size.

6.1 Early Rejection and Adaptive Specialists

In some operational contexts, early stopping should only be applied to *reject* non-robust inputs, while allowing certifiable inputs to run to the full 10,000-sample horizon. By disabling the precision and plateau exits and relying strictly on the UCB and dual-bankruptcy martingales of Section 5, our MEV framework acts as an adaptive rejection filter. Across all datasets and noise scales, uncertifiable inputs are safely aborted after an average of 944 to 1,833 samples (5.5–10.7× faster than standard RS), with a false rejection rate below 1.4% matching certified accuracies within 0.6 pp and incurring a minimal mutual radius penalty of just 0.61–3.72% (see Table 12 and Table 13 in Appendix E.1 for the complete ablation breakdown across Efficiency and Accuracy champions).

Beyond uniform early rejection, anytime validity aligns sampling budgets with downstream radius targets—e.g., tightening certificates in $[0.1, 0.5]$ when distinguishing radii above 1.2 offers little value, or vice versa—by gating stopping tolerance on $[R_{lcb}, R_{ucb}]$ (Appendix C.1). To demonstrate this for CIFAR-10 ($\sigma = 1.0$; Figure 2), we instantiate a **Small-R Specialist** ($r < 0.68$) and **Large-R Specialist** ($r > 1.35$) over an aggressive triage baseline (895 mean draws, 11.3× fewer than Cohen-10k). Each Specialist concentrates its full-horizon precision strictly inside its target zone while exiting rapidly outside it: Small-R matches Cohen-10k at low radii (40.87% vs. 41.17% at $r = 0.25$; 28.43% for Large-R) while exiting high-margin inputs in 201–311 conditional draws (3,937 overall, 2.57× speedup), while Large-R exits low-margin inputs in 274–327 draws (2,926 overall, 3.45× speedup) while devoting $\sim$10,100 draws at high radii to recover Cohen-10k coverage (15.77% vs. 16.17% at $r = 1.35$, vs. 3.97% for Small-R), aborting uncertifiable inputs in 1,170 draws (8.6× fewer).

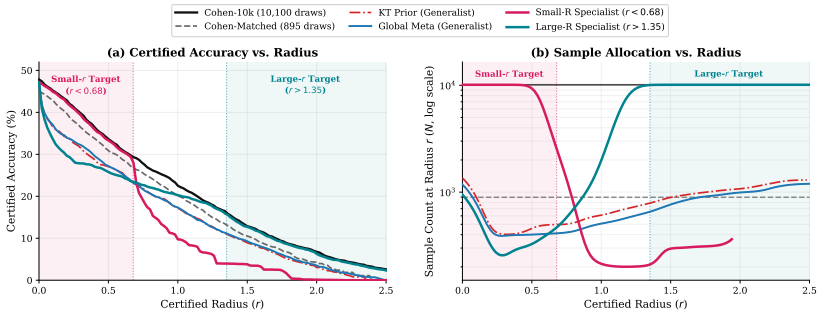


Figure 2: **Specialist Triage** (CIFAR-10, $\sigma = 1.0$, 3,000 eval images, 10-seed median + min–max bands). (a) Certified accuracy vs. r: Small-R ($r < 0.68$) and Large-R ($r > 1.35$) match Cohen-10k inside their target zones while shedding accuracy outside. (b) Conditional cost $\mathbb{E}[N \mid R_{\text{cert}} = r]$: Small-R invests $\sim 10^4$ draws at small r (201–311 at large r), and vice versa for Large-R.

Table 2: **Frequentist sequential baselines vs. E-values** (ImageNet, $\sigma = 0.25$, 2,000 eval images). All sequential methods share the Global champion rule ($\epsilon = 0.0126\sigma$, look-ahead) and rejection exits; multi-stage tests use 10 checkpoints, while LIL and E-values check every $B = 100$ draws. Meta-RS achieves $1.43\times$ speedup over Cohen-10k at near-identical ACR (0.440 vs. 0.448; full $\sigma \in \{0.50, 1.00\}$ sweep in Table 11).

σ	Method	Looks	Acc (%)	Avg Cert	Avg All	ACR	Speedup	FR (%)
0.25	Cohen-10k	1	69.4	10,100	10,100	0.448	1.00×	0.0
	Cohen-1k	1	69.0	1,100	1,100	0.358	9.18×	0.6
	Bonferroni (uniform)	10	69.4	10,100	9,610	0.438	1.05×	0.0
	Bonferroni (geometric)	10	69.4	10,100	9,615	0.424	1.05×	0.0
	α-spending (OBF)	10	69.4	10,100	9,686	0.444	1.04×	0.0
	α-spending (Pocock)	10	69.4	10,100	9,610	0.437	1.05×	0.0
	LIL boundary	all	69.4	10,100	9,616	0.299	1.05×	0.0
	KT prior (E-value)	all	69.3	7,905	7,354	0.425	1.37×	0.1
	Meta-RS (Global)	all	69.3	7,587	7,055	0.440	1.43×	0.1

6.2 Comparison against Frequentist Sequential Testing Baselines

To evaluate whether the gains of our game-theoretic E-value framework stem strictly from anytime-validity or from our sample-adaptive mixture prior, we benchmark Meta-RS against standard frequentist sequential testing paradigms across ImageNet ($\sigma \in \{0.25, 0.50, 1.00\}$, 2,000 eval images per scale): (1) **Multi-Stage Bonferroni Checks** over $t \in \{1\text{k}, 2\text{k}, \ldots, 10\text{k}\}$ splitting α uniformly or geometrically; (2) **Multi-Stage α-Spending Functions** using Lan–DeMets O'Brien–Fleming (OBF) and Pocock schedules; and (3) **Continuous Confidence Sequences** via sub-Gaussian LIL-stitching boundaries (Howard et al., 2021) checked every 100 draws.

As shown in Table 2 for $\sigma = 0.25$ (and Table 11 in Appendix E across all σ), Meta-RS requires $1.43\times$, $1.70\times$, and $2.21\times$ fewer samples than Cohen-10k at $\sigma = 0.25$, 0.5, and 1.0, outperforming the cheapest frequentist sequential test by 1.36–$1.71\times$. The multi-stage tests reach comparable ACR (0.424–0.444 vs. 0.440 at $\sigma = 0.25$) but save only 1.05–$1.30\times$ samples over Cohen-10k because intermediate checks rarely accumulate sufficient evidence to halt early. Meanwhile, continuous LIL boundaries suffer severe radius degradation (0.299 vs. 0.440 at $\sigma = 0.25$). For multi-stage frequentist tests, dividing α across 10 evaluation checkpoints dilutes statistical power at each point, widening confidence intervals. In contrast, Meta-RS leverages its sample-adaptive mixture prior to accumulate evidence rapidly while preserving anytime validity.

6.3 Case-Study: Operational Triage Framework

We now consider a scenario where the exact certification radius matters less than routing inputs into an operational risk tier. Rather than committing the full budget to every input, our MEV can halt certifications whenever $[LCB_t, UCB_t]$ lies entirely inside one of four partitions: **Bucket A (Non-Robust,** $r = 0$) for immediate rejection or human review; **Bucket B (Low,** $0 < r \leq 1.0$) for automated scrutiny; **Bucket C (Medium,** $1.0 < r \leq 1.5$) for monitored deployment; and **Bucket D (High,** $r > 1.5$) for autonomous deployment. Under this operational triage framework, we tested Cohen et al. (2019) at $10,000$ samples, a KT Prior Method-of-Mixtures approach with a constant

Table 3: **Operational triage** (CIFAR-10, $\sigma = 1.0$, 3,000 eval images; buckets at $r \in \{0, 1, 1.5\}\sigma$, referenced to Cohen-10k). Tiered races use $\alpha_{A..D} = (0.05, 0.05, 0.025, 0.01)$; the relaxed cascade adds ± 0.05 boundary proximity ($\alpha = 0.01$, yielding *ambiguous* labels verified against 100,000-draw $\hat{p}$). Avg N (incl. Phase I) and Bucket A cost reported as 10-seed medians.

Strategy	FWER	Avg N	Avg N, A	Definitive	Correct	Ambig. supp.	Undecided
Cohen-10k (ref.)	–	10,100	10,100	100.0	100.0	0.0	0.0
KT, CI ($\alpha = 0.01$)	0.010	2,424	2,435	87.0	87.0	0.0	13.0
KT tiered race	0.135	2,004	2,128	89.3	89.2	0.0	10.7
KT relaxed cascade	0.145	866	1,087	80.9	80.9	19.1	0.0
Meta tiered race	≤ 0.067	1,889	1,892	90.0	89.8	0.0	9.9
Meta relaxed cascade	≤ 0.077	796	937	81.7	81.6	18.3	0.0

$\alpha = 0.01$, and two application specific approaches. The Tiered Spec Race assigns each bin a specific budget $\{\alpha_A, \ldots, \alpha_D\} = \{0.05, 0.05, 0.025, 0.01\}$, representing the levels of tolerance of a false-positive in each category, and performs four independent E-processes for each operational bucket. Samples are assigned to bucket i at timestep T if its LCB and UCB are contained within the associated bin with $M_i(X_{1:T}) \geq 1/\alpha'_i$, where $\alpha'_i = \alpha_i \cdot (0.2 + 0.8 \cdot w_{meta,i})$.

The Relaxed Tiered Cascade follows the above process, however, if samples remain unresolved after 400 samples the system attempts to prove proximity to any bin boundary $p_b \in \{0.5, 0.8413, 0.9332\}$ (corresponding to $r = \{0, 1, 1.5\}$), through a dual-rejection process considering $H_0^- : p = p_b - \epsilon$ and $H_0^+ : p = p_b + \epsilon$, using an $\epsilon = 0.05$. If $M(X; p_b - \epsilon) \geq 1/\alpha$ and $M(X; p_b + \epsilon) \geq 1/\alpha$, and the empirical mean $\hat{p}_{mle}$ lies within $[p_b - \epsilon, p_b + \epsilon]$, the sample is mathematically trapped within the $[p_b - \epsilon, p_b + \epsilon]$ buffer. This follows as the mixture wealth is quasi-convex in p_0 and minimized in the vicinity of $\hat{p}_{mle}$, so the set of non-rejected hypotheses is an interval (Section 3); if both endpoints are rejected while the interval intersects the buffer, it must lie entirely within it.

If this is achieved, the sample is classified into the corresponding bin, while also being provisioned with a secondary label denoting boundary ambiguity. This choice is a deliberate intervention to avoid allocating redundant samples to distinguish infinitesimally close probabilities. Table 3 summarizes the performance of this scheme on CIFAR-10 at $\sigma = 1.0$ for $3,000$ images. The Meta relaxed cascade decides every input in an average of 796 samples. Of these, 81.7% receive a definitive bucket. This approach is 8% cheaper than the KT prior running the same cascade, and $3.0\times$ faster than KT without the triage, reinforcing the operational advantage of our tiered, application-specific risk allocation framework, that cannot be matched by standard certification workflows.

Transferability and Offline Batch Allocation Appendix F shows MNIST/CIFAR-10 Meta-Learners transfer zero-shot to ImageNet with 100% median decision agreement at 0.8–2.4% sample overhead (Table 14), and Appendix G formulates offline knapsack budget reallocation ($2.43\times$ savings under fixed budgets).

7 Conclusion

In this work, we demonstrated that randomized smoothing can be transformed from a static, computationally exhaustive process into a dynamic, application-aware certification pipeline optimized for **strategic resource management**. While computational cost has always presented significant barriers for certified robustness, our E-value meta-learner produces substantial empirical gains in this space (in throughput on every backbone we tested, and in single-input latency on ImageNet), without compromising robustness. Under the High-Fidelity regime, Meta-RS achieves certified accuracy and radius within 1% of Cohen-10k while reducing samples from $1.08\times$ to $2.58\times$ (up to 25% fewer samples than the KT prior under the same stopping rule), translating to significant wall-clock savings. Under the Maximum-Efficiency regime, the plateau exit delivers an 89% reduction in sample requirements, validating the utility of our framework for adaptive resource management.

However, headline computational metrics do not tell the full story of this work. Our framework demonstrates that it is possible to dynamically balance robustness with compute on a sample-by-sample basis, through early rejections or downstream-focused triage. While anytime-validity incurs a marginal cost regarding the realizable tightness of bounds, our results highlight the overwhelming utility of transitioning to meta-learned E-values. The true value of this approach lies not just in its computational efficiency, but in proving that dynamic, application-specific triage is finally possible, marking a definitive leap toward the real-world deployment of certified machine learning.

ACKNOWLEDGMENTS

This work was supported by the Australian Defence Science and Technology (DST) Group via the Advanced Strategic Capabilities Accelerator (ASCA) program. This research was supported by The University of Melbourne's Research Computing Services and the Petascale Campus.

IMPACT STATEMENT

This work explores the potential for improving the computational efficiency of mechanisms for achieving Certified Robustness, with the aim of significantly enhancing the range of applicability for these systems. Improving the robustness of models, and lowering the energy consumption of our validation processes, has clear societal implications, especially as we move to a landscape where AI is increasingly integrated into society. However, with that said, we feel that there are two key societal concerns with pursuing robustness research, which we must note here.

The first of which is that there are some applications where a lack of robustness in a model may be positive. In a world where broad scale surveillance is increasingly normalized, it may well be the case that adversarial attacks may induce privacy, creating a net public good.

The second relates to how works like this position risk and harm. A common precept within the Adversarial Machine Learning community is to assume a particular threat model, with the nature of academic comparisons often incentivizing us to then follow in the footsteps of those who came before us. However, in doing so, we inadvertently create—and, crucially, present—a myopic view of the risk landscape (Cullen et al., 2025, 2026a). In essence, we portray to practitioners that risk is concentrated within the areas in which we act as a community, when our investigations may be more motivated by historic alignment to academic norms and mathematical convenience. This work considers ℓ_2 perturbations, which while aligned with acoustic threat models, still represent a restriction relative to the overall threat landscape.

We emphasize the above point not just for the risks of erroneous portrayals of risk to practitioners, but also because our focus on these spaces inherently biases real attacker behavior away from these threat models. After all, if an attacker understands that an ℓ_2 threat model is likely defended against, they're naturally incentivized to consider an alternative pathway for model manipulation.

With these points made, we still believe that research into defenses, and in particular certified defenses, induce a net societal gain. Improving robustness to natural or adversarial perturbations will improve the performance of deployed systems.

REFERENCES

Brent, R. P. An Algorithm with Guaranteed Convergence for Finding a Zero of a Function. *The Computer Journal*, 14(4):422–425, 1971.

Carlini, N., Tramer, F., Dvijotham, K. D., Rice, L., Sun, M., and Kolter, J. Z. (Certified!!) Adversarial Robustness for Free! In *International Conference on Learning Representations (ICLR)*, 2023.

Chen, R., Li, J., Yan, J., Li, P., and Sheng, B. Input-Specific Robustness Certification for Randomized Smoothing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6295–6303, 2022.

Chiang, P.-y., Ni, R., Abdelkader, A., Zhu, C., Studer, C., and Goldstein, T. Certified Defenses for Adversarial Patches. *arXiv preprint arXiv:2003.06693*, 2020.

Clopper, C. J. and Pearson, E. S. The Use of Confidence or Fiducial Limits Illustrated in the case of the Binomial. *Biometrika*, 26(4):404–413, 1934.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified Adversarial Robustness via Randomized Smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

Cullen, A. C., Montague, P., Liu, S., Erfani, S. M., and Rubinstein, B. I. Double Bubble, Toil and Trouble: Enhancing Certified Robustness through Transitivity. *Advances in Neural Information Processing Systems*, 35:19099–19112, 2022.

Cullen, A. C., Liu, S., Montague, P., Erfani, S. M., and Rubinstein, B. I. P. Et Tu Certifications: Robustness Certificates Yield Better Adversarial Examples. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024a.

Cullen, A. C., Montague, P., Liu, S., Erfani, S. M., and Rubinstein, B. I. It's Simplex! Disaggregating Measures to Improve Certified Robustness. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 2886–2900. IEEE, 2024b.

Cullen, A. C., Montague, P., Erfani, S. M., and Rubinstein, B. I. P. Position: Certified Robustness Does Not (Yet) Imply Model Security. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 81185–81198. PMLR, 13–19 Jul 2025.

Cullen, A. C., Marchant, N. G., Montague, Paul Xie, J., and Rubinstein, B. I. P. Adversarial Risk in the Generative AI era necessitates dropping the small Epsilon Ball. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 39, 2026a.

Cullen, A. C., Marchant, N. G., Montague, Paul Xie, J., and Rubinstein, B. I. P. What Was That Again? Certified Robustness for Automatic Speech Recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 39, 2026b.

Doob, J. L. Regularity Properties of Certain Families of Chance Variables. *Transactions of the American Mathematical Society*, 47(3):455–486, 1940.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography Conference*, TCC, pp. 265–284. Springer, 2006.

Gershman, S. and Goodman, N. Amortized Inference in Probabilistic Reasoning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36, 2014.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572*, 2014.

Grünwald, P., de Heide, R., and Koolen, W. M. Safe Testing. In *2020 Information Theory and Applications workshop (ITA)*, pp. 1–54. IEEE, 2020.

Hein, M. and Andriushchenko, M. Formal Guarantees on the Robustness of a Classifier Against Adversarial Manipulation. In *Advances in Neural Information Processing Systems*, volume 30 of *NeurIPS*, 2017.

Horváth, M. Z., Mueller, M. N., Fischer, M., and Vechev, M. Boosting randomized smoothing with variance reduced classifiers. In *International Conference on Learning Representations*, 2022.

Howard, S. R., Ramdas, A., McAuliffe, J., and Sekhon, J. Time-Uniform, Nonparametric, Nonasymptotic Confidence Sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021.

Johari, R., Koomen, P., Pekelis, L., and Walsh, D. Peeking at A/B Tests: Why it matters, and what to do about it. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1517–1525, 2017.

Kelly, J. L. A New Interpretation of Information Rate. *The Bell System Technical Journal*, 35(4): 917–926, 1956.

Krichevsky, R. and Trofimov, V. The Performance of Universal Encoding. *IEEE Transactions on Information Theory*, 27(2):199–207, 1981.

Krizhevsky, A., Hinton, G., et al. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009.

Lai, T. L. On Confidence Sequences. *The Annals of Statistics*, pp. 265–280, 1976.

Lan, K. G. and DeMets, D. L. Discrete Sequential Boundaries for Clinical Trials. *Biometrika*, pp. 659–663, 1983.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified Robustness to Adversarial Examples with Differential Privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672. IEEE Computer Society, 2019.

Leino, K., Wang, Z., and Fredrikson, M. Globally-Robust Neural Networks. In *International Conference on Machine Learning*, pp. 6212–6222. PMLR, 2021.

Levine, A. and Feizi, S. (de)Randomized Smoothing for Certifiable Defense against Patch Attacks. *Advances in Neural Information Processing Systems*, 33:6465–6475, 2020.

Li, B., Chen, C., Wang, W., and Carin, L. Certified Adversarial Robustness with Additive Noise. In *Advances in Neural Information Processing Systems*, volume 32, pp. 9459–9469. NeurIPS, 2019.

Lyu, Z., Guo, M., Wu, T., Xu, G., Zhang, K., and Lin, D. Towards Evaluating and Training Verifiably Robust Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4308–4317, 2021.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*, 2018.

Mirman, M., Gehr, T., and Vechev, M. Differentiable Abstract Interpretation for Provably Robust Neural Networks. In *International Conference on Machine Learning*, pp. 3578–3586. PMLR, 2018.

Mohapatra, J., Weng, T.-W., Chen, P.-Y., Liu, S., and Daniel, L. Towards Verifying Robustness of Neural Networks against a family of Semantic Perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 244–252, 2020.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 8024–8035. NeurIPS, 2019.

Ramdas, A., Grünwald, P., Vovk, V., and Shafer, G. Game-Theoretic Statistics and Safe Anytime-Valid Inference. *Statistical Science*, 38(4):576–601, 2023.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Salman, H., Li, J., Razenshteyn, I., Zhang, P., Zhang, H., Bubeck, S., and Yang, G. Provably Robust Deep Learning via Adversarially Trained Smoothed Classifiers. In *Advances in Neural Information Processing Systems*, volume 32, pp. 11292–11303. NeurIPS, 2019a.

Salman, H., Yang, G., Zhang, H., Hsieh, C.-J., and Zhang, P. A Convex Relaxation Barrier to Tight Robustness Verification of Neural Networks. In *Advances in Neural Information Processing Systems*, 2019b.

Salman, H., Sun, M., Yang, G., Kapoor, A., and Kolter, J. Z. Denoised Smoothing: A Provable Defense for Pretrained Classifiers. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:21945–21957, 2020.

Shafer, G. The Language of Betting as a Strategy for Statistical and Scientific Communication. *arXiv preprint arXiv:1903.06991*, 2019.

Shafer, G. and Vovk, V. *Game-Theoretic Foundations for Probability and Finance*. John Wiley & Sons, 2019.

Shi, Z., Jin, Q., Zhang, H., Kolter, Z., Jana, S., and Hsieh, C.-J. Formal Verification for Neural Networks with General Nonlinearities via Branch-And-Bound. In *2nd Workshop on Formal Verification of Machine Learning (WFVML 2023)*, 2023.

Singh, G., Gehr, T., Püschel, M., and Vechev, M. An Abstract Domain for Certifying Neural Networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–30, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing Properties of Neural Networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tsuzuku, Y., Sato, I., and Sugiyama, M. Lipschitz-Margin Training: Scalable Certification of Perturbation Invariance for Deep Neural Networks. In *Advances in Neural Information Processing Systems*, volume 31. NeurIPS, 2018.

Ville, J. *Etude Critique de la Notion de Collectif*, volume 3. Gauthier-Villars Paris, 1939.

Voráček, V. Treatment of Statistical Estimation Problems in Randomized Smoothing for Adversarial Robustness. *Advances in Neural Information Processing Systems*, 37:133464–133486, 2024.

Vovk, V. and Wang, R. E-values: Calibration, Combination and Applications. *The Annals of Statistics*, 49(3):1736–1754, 2021.

Wang, S., Zhang, H., Xu, K., Lin, X., Jana, S., Hsieh, C.-J., and Kolter, J. Z. Beta-CROWN: Efficient Bound Propagation with Per-Neuron Split Constraints for Neural Network Robustness Verification. *Advances in Neural Information Processing Systems*, 34, 2021.

Waudby-Smith, I. and Ramdas, A. Estimating Means of Bounded Random Variables by Betting. *Journal of the Royal Statistical Society: Series B-Statistical Methodology*, 86(1):1–27, 2024.

Weng, L., Zhang, H., Chen, H., Song, Z., Hsieh, C.-J., Daniel, L., Boning, D., and Dhillon, I. Towards Fast Computation of Certified Robustness for ReLU Networks. In *International Conference on Machine Learning*, pp. 5276–5285. PMLR, 2018.

Xie, Q. and Barron, A. R. Asymptotic Minimax Regret for Data Compression, Gambling, and Prediction. *IEEE Transactions on Information Theory*, 46(2):431–445, 2000.

Xu, K., Shi, Z., Zhang, H., Wang, Y., Chang, K.-W., Huang, M., Kailkhura, B., Lin, X., and Hsieh, C.-J. Automatic Perturbation Analysis for Scalable Certified Robustness and Beyond. *Advances in Neural Information Processing Systems*, 33, 2020.

Zhai, R., Dan, C., He, D., Zhang, H., Gong, B., Ravikumar, P., Hsieh, C.-J., and Wang, L. Macer: Attack-Free and Scalable Robust Training via Maximizing Certified Radius. In *International Conference on Learning Representations*, 2020.

Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. Efficient Neural Network Robustness Certification with General Activation Functions. In *Neural Information Processing Systems (NeurIPS)*, 2018.

A Expanded Related Work

A.1 Certification Mechanisms

First introduced by Lecuyer et al. (2019), randomized smoothing based certified robustness builds upon Monte Carlo estimators of the expectation of a class prediction. While the original formulation was constructed in terms of differential privacy (Dwork et al., 2006), recent approaches have improved performance through Rényi divergence (Li et al., 2019) and parameterising worst-case behaviors (Cohen et al., 2019; Salman et al., 2019a; Cullen et al., 2022).

Significant research has focused on improving the underlying base classifier f to be more amenable to randomized smoothing. Salman et al. (2019a) employs adversarial training to harden the model against perturbed inputs, while **MACER** (Zhai et al., 2020) directly optimizes a robustness loss that encourages large classification margins. Our work is orthogonal to these training-time interventions; we focus on the *inference-time* efficiency of the certification process itself to significantly reduce

sample complexity for *any* base model, including those that have undergone the training-time interventions of MACER and Salman et al. (2019a). The same holds for denoised and diffusion-based smoothing, where a (potentially large) denoiser is prepended to the base classifier (Salman et al., 2020; Carlini et al., 2023). As each Monte-Carlo sample then requires a costly denoising pass, these approaches further underscore the importance of reducing sample complexity.

In all works, the primary metric for evaluating Randomized Smoothing is the **Certified Accuracy** at radius r, defined as the fraction of the test set that is both correctly classified by the smoothed model and has a certified radius $R \geq r$

$$\text{Acc}(r) = \mathbb{P}_{(x,y)\sim\mathcal{D}}[g(x) = y \text{ and } R(x) \geq r] \tag{7}$$

where g is the smoothed classifier.

A.1.1 Interval Bound Propagation

In the absence of probabilistic methods, conservative certificates upon the impact of norm-bounded perturbations can be constructed by way of either interval bound propagation (IBP) which propagates interval bounds through the model; or convex relaxation, which utilizes linear relaxation to construct bounding output polytopes over input bounded perturbations. In contrast to randomized smoothing, which constructs isotropic measures of ℓ_p-robustness, interval bound propagation and its associated techniques attempt to propagate the potential influence of all possible perturbations through the model, producing an anisotropic measure of the potential response of a model to any potential perturbation (Salman et al., 2019b; Mirman et al., 2018; Weng et al., 2018; Zhang et al., 2018; Singh et al., 2019; Mohapatra et al., 2020). Of these, IBP is more general, while convex relaxation typically provides tighter bounds (Lyu et al., 2021).

Utilizing these techniques requires introducing an augmented loss function during training to promote tight output bounds (Xu et al., 2020)—creating significant architectural friction relative to RS style certifications, which can be applied to any model architecture. Bound propagation schemes have also been limited in the types of network architectures that they can efficiently construct bounds through. While general activation functions such as tanh and sigmoid have been supported since CROWN (Zhang et al., 2018), efficiently extending these approaches to general nonlinearities and architectures remains an active area of research (Shi et al., 2023). Moreover both exhibit a time and memory complexity that makes them infeasible for complex model architectures or high-dimensional data (Wang et al., 2021; Chiang et al., 2020; Levine & Feizi, 2020).

A.1.2 Global Lipschitz

Global Lipschitz takes an alternative approach to constructing certifications, a point that they distinguish through the framing of local and global robustness. The guarantees provided by prior works, which can take the form

$$\|\mathbf{x} - \mathbf{x}'\|_p \leq \epsilon \implies F(\mathbf{x}) = F(\mathbf{x}') \tag{8}$$

are considered to be local properties, that relate $\mathbf{x}$ and ϵ. Lipschitz based techniques instead attempt to construct their certifications in terms of *global* robustness, where

$$\forall \mathbf{x}_1, \mathbf{x}_2 : \|\mathbf{x}_1 - \mathbf{x}_2\|_p \leq \epsilon \implies F(\mathbf{x}_1) \overset{\perp}{=} F(\mathbf{x}_2) \ . \tag{9}$$

Here $\perp$ is the marker for an *abstained* class prediction, and $c_1 \overset{\perp}{=} c_2$ denotes that either $c_1 = \perp$, $c_2 = \perp$, or $c_1 = c_2$. In essence such a form of certification involves constructing a model that has not only an infinitesimally thin decision boundary, but a margin between the regions associated with each class, where ϵ then becomes the shortest ℓ_p distance to span the boundary. Several attempts have been made to use Lipschitz bounds during training to promote robustness. These include constructing provable lower bounds on the norm of the input manipulation required to change classifier decisions based upon the network architecture (Hein & Andriushchenko, 2017); modifying the loss associated with logits different than the ground-truth class (Tsuzuku et al., 2018); and GloRoNets, which add an additional logit corresponding to the predicted class at a point (Leino et al., 2021). While these techniques can be an order of magnitude faster than randomized smoothing, they are both less flexible—in terms of the architectures they support—and often produce smaller certifications than randomized smoothing (Leino et al., 2021).

A.2 Anytime Valid Mechanisms

As an alternative to betting-based martingales, anytime-valid confidence sequences can be constructed using the Law of the Iterated Logarithm (LIL). Of particular note is the *stitching* construction from Howard et al. (2021), which provides a boundary u_t such that the probability of the empirical mean ever crossing u_t is bounded by α. While LIL-stitching is robust to prior mis-specification, it is generally less efficient than Mixture E-values when a reasonably accurate prior (like our Meta-Learner) is available, as it lacks the ability to aggressively allocate its convergence on specific regions of the hypothesis space. In our testing (Table 2), LIL-stitching needs only 1.05–$1.30\times$ fewer samples than Cohen-10k and reaches 67–74% of its average certified radius, against 1.43–$2.21\times$ at $\geq 97\%$ for the meta-learned E-values.

B Theoretical Proofs and Soundness

This section provides complete formal proofs for the anytime-validity guarantees, the Method of Mixtures, and the Safety-Efficiency Separation theorem presented in Section 3.

B.1 Proof of Lemma 1 (Mixture Martingale Property)

Proof. Let $H_0 : p = p_0$. For any single alternative hypothesis $q \in (0, 1)$, the likelihood ratio process

$$L_t(p_0; q) = \prod_{i=1}^{t} \frac{q^{X_i}(1-q)^{1-X_i}}{p_0^{X_i}(1-p_0)^{1-X_i}} = \frac{q^{h_t}(1-q)^{t-h_t}}{p_0^{h_t}(1-p_0)^{t-h_t}} \tag{10}$$

is a product of independent, non-negative random variables. Under H_0, each term has conditional expectation:

$$\mathbb{E}_{H_0}\left[\frac{q^{X_t}(1-q)^{1-X_t}}{p_0^{X_t}(1-p_0)^{1-X_t}} \;\middle|\; \mathcal{F}_{t-1}\right] = p_0 \cdot \frac{q}{p_0} + (1-p_0) \cdot \frac{1-q}{1-p_0} = q + (1-q) = 1 \ . \tag{11}$$

Hence $(L_t(p_0; q))_{t\geq 1}$ is a non-negative $\mathcal{F}_t$-martingale with initial wealth $L_0(p_0; q) = 1$.

Now consider the mixture E-value $\bar{W}_t(p_0) = \int_0^1 L_t(p_0; q)dQ(q)$, where Q is a probability measure on $[0, 1]$ that is $\mathcal{F}_0$-measurable. Because $L_t(p_0; q) \geq 0$ for all t and q, Tonelli's theorem justifies exchanging the conditional expectation and the integral:

$$\mathbb{E}_{H_0}\left[\bar{W}_t(p_0) \mid \mathcal{F}_{t-1}\right] = \mathbb{E}_{H_0}\left[\int_0^1 L_t(p_0; q)dQ(q) \;\middle|\; \mathcal{F}_{t-1}\right] \tag{12}$$

$$= \int_0^1 \mathbb{E}_{H_0}\left[L_t(p_0; q) \mid \mathcal{F}_{t-1}\right] dQ(q) \tag{13}$$

$$= \int_0^1 L_{t-1}(p_0; q)dQ(q) = \bar{W}_{t-1}(p_0) \ . \tag{14}$$

Since $\bar{W}_0(p_0) = \int_0^1 1dQ(q) = 1$, $(\bar{W}_t(p_0))_{t\geq 1}$ is a non-negative martingale with respect to $(\mathcal{F}_t)_{t\geq 1}$, proving Lemma 1. □

B.2 Proof of Convexity of Mixture E-values

Proof. Let $E^{(1)}, \ldots, E^{(K)}$ be non-negative random variables satisfying $\mathbb{E}_{H_0}[E^{(k)}] \leq 1$ for all $k \in \{1, \ldots, K\}$. For any set of non-negative weights w_k with $\sum_{k=1}^K w_k = 1$, linearity of expectation yields:

$$\mathbb{E}_{H_0}\left[\sum_{k=1}^K w_k E^{(k)}\right] = \sum_{k=1}^K w_k \mathbb{E}_{H_0}\left[E^{(k)}\right] \leq \sum_{k=1}^K w_k \cdot 1 = 1 \ . \tag{15}$$

Since $\sum_{k=1}^K w_k E^{(k)} \geq 0$, the convex combination is also an E-value for H_0. When applied to martingales $(W_t^{(k)})_{t\geq 1}$, linearity of conditional expectation similarly guarantees that the convex combination is a martingale. □

B.3 Proof of Theorem 1 (Anytime-Valid Soundness and Contiguity)

Proof. Let p denote the true success probability of the smoothed classifier for target class c_A. By Lemma 1, $(\bar{W}_t(p))_{t\geq 1}$ is a non-negative martingale with $\bar{W}_0(p) = 1$. By Ville's maximal inequality for non-negative supermartingales (Ville, 1939):

$$\mathbb{P}\left(\sup_{t\geq 1} \bar{W}_t(p) \geq \frac{1}{\alpha}\right) \leq \alpha \cdot \bar{W}_0(p) = \alpha \ . \tag{16}$$

The confidence set at step t is defined as $C_t = \{p_0 \in [0,1] : W_t^{\max}(p_0) < 1/\alpha\}$, where $W_t^{\max}(p_0) = \max_{s\leq t} \bar{W}_s(p_0)$. The true parameter p is excluded from C_t if and only if $W_t^{\max}(p) \geq 1/\alpha$. The event that coverage fails at any point in the infinite sequence satisfies:

$$\{\exists t \geq 1 : p \notin C_t\} = \left\{\exists t \geq 1 : \max_{s\leq t} \bar{W}_s(p) \geq \frac{1}{\alpha}\right\} = \left\{\sup_{t\geq 1} \bar{W}_t(p) \geq \frac{1}{\alpha}\right\} \ . \tag{17}$$

Consequently, $\mathbb{P}(\exists t \geq 1 : p \notin C_t) \leq \alpha$, which implies $\mathbb{P}(\forall t \geq 1 : p \in C_t) \geq 1 - \alpha$. In particular, because $\underline{p}_t = \inf C_t \leq p$ whenever $p \in C_t$, we have $\mathbb{P}(\exists t \geq 1 : p < \underline{p}_t) \leq \alpha$.

To show that C_t is a contiguous interval, consider the mapping $p_0 \mapsto \log L_t(p_0; q) = h_t \log q + (t - h_t)\log(1-q) - h_t \log p_0 - (t - h_t)\log(1 - p_0)$. For any $h_t \in [0, t]$, the function $f(p_0) = -h_t \log p_0 - (t - h_t)\log(1 - p_0)$ has second derivative

$$f''(p_0) = \frac{h_t}{p_0^2} + \frac{t - h_t}{(1 - p_0)^2} \geq 0 \ , \tag{18}$$

which is strictly positive on $(0, 1)$ whenever $t \geq 1$. Hence $p_0 \mapsto L_t(p_0; q)$ is log-convex in p_0. By the Prékopa–Leindler inequality and properties of log-convex mixtures, the mixture $\bar{W}_t(p_0) = \int_0^1 L_t(p_0; q)dQ(q)$ is log-convex, and therefore quasiconvex (unimodal) in p_0. Since the pointwise maximum of quasiconvex functions preserves quasiconvexity, $W_t^{\max}(p_0)$ is quasiconvex in p_0. Its sub-level set $C_t = \{p_0 \in [0, 1] : W_t^{\max}(p_0) < 1/\alpha\}$ is therefore convex, which in $\mathbb{R}$ is a single contiguous interval $[\underline{p}_t, \bar{p}_t]$. Thus $\underline{p}_t = \inf C_t$ is uniquely determined by 1D root-finding. □

B.4 Proof of Theorem 2 (Safety-Efficiency Separation)

Proof. Let τ be an arbitrary stopping time adapted to the filtration $(\mathcal{F}_t)_{t\geq 0}$, allowing values in $\mathbb{N} \cup \{\infty\}$. By definition, on the event $\{\tau = t\}$, the lower bound evaluated at stopping is $\underline{p_\tau} = \underline{p_t}$. Therefore:

$$\{p < \underline{p_\tau}\} = \bigcup_{t=1}^{\infty} \Big(\{\tau = t\} \cap \{p < \underline{p_t}\}\Big) \subseteq \bigcup_{t=1}^{\infty} \{p < \underline{p_t}\} = \big\{\exists t \geq 1 : p < \underline{p_t}\big\} \ . \tag{19}$$

By monotonicity of probability measures:

$$\mathbb{P}(p < \underline{p_\tau}) \leq \mathbb{P}\left(\exists t \geq 1 : p < \underline{p_t}\right) = \mathbb{P}\left(\sup_{t\geq 1} \bar{W}_t(p) \geq \frac{1}{\alpha}\right) \leq \alpha \ , \tag{20}$$

where the final inequality is Ville's maximal inequality (Theorem 1).

Notice that this inequality holds unconditionally without requiring $\mathbb{E}[\tau] < \infty$ or uniform integrability, because it is bounded by the pathwise supremum over the entire filtration. Hence, the statistical validity of the certified radius $R_\tau = \sigma\Phi^{-1}(\underline{p_\tau})$ is completely invariant to the choice of stopping rule τ. □

C Algorithms

To provide further details to the processes outlined in Sections 4 and 5, we present the full expanded algorithm for the training of the Meta-Learner using a Kelly-Optimal inspired process in Algorithm 1, and the overall certification approach in Algorithm 2.

Datasets and Backbone Architecture Our experiments consider attacks against MNIST (LeCun et al., 1998) (GNU v3.0 license), CIFAR-10 (Krizhevsky et al., 2009) (MIT license), and ImageNet (Russakovsky et al., 2015) (which uses a custom, non-commercial license). In the case of models defended by randomised smoothing, each model was trained in PyTorch (Paszke et al., 2019)

Algorithm 1 Meta-Learner Training (Kelly-Optimal Betting)

Input: Training dataset $\mathcal{D}_{train}$, noise level σ, base classifier f, augmentation factor M, epochs E.
Phase I: Data Collection (Offline)
$\mathcal{S} \leftarrow \emptyset$
for each image $x^{(i)} \in \mathcal{D}_{train}$ **do**
    Extract embedding $\phi^{(i)}$ of the clean image.
    Draw $N_{sel} + N_{max}$ samples and split the first ones into disjoint N_{sel}-draw Phase I blocks $j = 1, \ldots, J$; for each block record its selected class $c_A^{(i,j)}$, glimpse $\hat{p}_{sel}^{(i,j)}$, mean softmax $\mathbf{p}^{(i,j)}$ and the count $h_N^{(i,j)}$ of $c_A^{(i,j)}$ among the other N_{max} draws.
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\phi^{(i)}, \{(\mathbf{p}^{(i,j)}, \hat{p}_{sel}^{(i,j)}, h_N^{(i,j)})\}_{j=1}^{J})\}$ ▷ Store ground-truth tuples
end for
Phase II: Kelly Optimization (Online)
Initialize Meta-Learner $\mathcal{M}_\theta$.
for epoch $e = 1$ to E **do**
    Sample an index set $\mathcal{I} \subset \{1, \ldots, |\mathcal{S}|\}$ of size n uniformly at random.
    for $i \in \mathcal{I}$ **do**
        Draw a block j uniformly at random; $(\mathbf{p}^{(i)}, \hat{p}_{sel}^{(i)}, h_N^{(i)}) \leftarrow (\mathbf{p}^{(i,j)}, \hat{p}_{sel}^{(i,j)}, h_N^{(i,j)})$ ▷ A real Phase I glimpse
        $(\mathbf{w}^{(i)}, \mathbf{z}^{(i)}) \leftarrow \mathcal{M}_\theta(\phi^{(i)}, \mathbf{p}^{(i)}, \hat{p}_{sel}^{(i)})$. ▷ Predict weights and raw parameters
        $\boldsymbol{\beta}^{(i)}, \boldsymbol{\gamma}^{(i)} \leftarrow \text{Clamp}(\text{Softplus}(\mathbf{z}^{(i)}) + 0.1, [0.1, 500.0])$. ▷ Survival bias & Stability
    end for
    // Minimize negative expected log-wealth (Kelly Loss)
    $\lambda_e \leftarrow \max(1.0, 10.0 \cdot (1 - \frac{e}{0.8E}))$ ▷ Decay Penalty
    $\hat{p}_{mle}^{(i)} \leftarrow h_N^{(i)}/N$ ▷ Ground-truth rate used by the containment penalty
    $\kappa_k^{(i)} \leftarrow I_{b_k}(h_i + \beta_k, N - h_i + \gamma_k) - I_{a_k}(h_i + \beta_k, N - h_i + \gamma_k)$ ▷ Truncation correction
    $\mathcal{L}(\theta) = -\frac{1}{n} \sum_i \left[\log \sum_k w_k \cdot \frac{P(h_i|N,\beta_k,\gamma_k) \cdot \kappa_k^{(i)}}{Z(a_k,b_k,\beta_k,\gamma_k)} - \lambda_e \sum_k \text{dist}(\hat{p}_{mle}^{(i)}, \mathcal{R}_k) \right]$
    // $P(\cdot)$ is the beta-binomial likelihood; dist$(\hat{p}, \mathcal{R}_k)$ is the ℓ_1 distance from $\hat{p}$ to the nearest boundary of $\mathcal{R}_k$ (zero if $\hat{p} \in \mathcal{R}_k$).
    $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$.
end for
Return: Trained Meta-Learner $\mathcal{M}_\theta$.

We trained three backbone models as the base classifiers, each of which were trained to robust under Gaussian noise through augmentations drawn from $\mathcal{N}(0, \sigma^2 I)$. For MNIST and CIFAR-10, we employed a ResNet-18 architecture, which was modified for a single input channel for MNIST. In the case of ImageNet, we considered a ResNet-50 architecture. Features employed by the Meta-Learner are extracted from the final `avgpool` layer of their respective backbones (512-dimensional for ResNet-18, or 2048-dimensional with a linear projection to 512-dimensional for ResNet-50).

Meta-Learner Architecture The Meta model $\mathcal{M}_\theta$ builds upon a `LayerNorm` stage followed by a linear projection to 512 dimensions. This is then followed by a sequential MLP mapping, containing two linear layers (512 units) with `LayerNorm` and `ReLU` activations, before branching into independent linear heads for mixture weights (π), Beta parameters (β, γ), and optionally, dynamic support boundaries (a, b).

Optimization and Training As discussed previously, the meta-learner is trained to maximize the Kelly Criterion Expected Log-Wealth over full bitstream sequences. To achieve this, optimization occurred through an Adam optimizer for 60 epochs with a learning rate of 10^{-3} and a cosine schedule, gradient clipping at 1.0, batch size 64, and no weight decay. Validation loss is checked every epoch, and the best validation epoch restored (patience 15), with the containment penalty weight λ decayed from 10.0 to 1.0 over 80% of training. The safety anchor weight is fixed to 0.01. We report results over 10 independent meta-learner training seeds. Data splits are strictly disjoint: `meta-train` uses 5,000 images for MNIST and CIFAR-10 (10,000 for ImageNet); `meta-val` (used for champion selection) uses 1,000 images (2,000 for ImageNet); and the final reported `eval` split uses 3,000 images for MNIST and CIFAR-10, and 2,000 images for ImageNet. Every table and evaluation in

Algorithm 2 Sequential Radius Estimation with Dynamic Fast-Exits

Input: Test image x, noise level σ, target classifier f, Meta-Learner $\mathcal{M}$.
Parameters: Significance α, batch size B, max samples N_{max}, failure threshold ϵ_{fail}, parameters $\epsilon_{start}, \epsilon_{end}$.
Phase I: Holdout and Prior Initialization
Extract $\phi(x)$ of the clean image from f.
Draw N_{sel} samples to identify target class c_A (the most frequent class), the empirical mean $\hat{p}_{sel}$ and the mean softmax $\mathbf{p}(x)$ over the N_{sel} draws.
$\mathcal{M}(\phi(x), \mathbf{p}(x), \hat{p}_{sel}) \to \{w_k, \beta_k, \gamma_k, \mathcal{R}_k\}_{k=1}^{K}$.
Phase II: Sequential Certification
$W_0(p_0) \leftarrow 1$ and $W_0^{\max}(p_0) \leftarrow 1$ for all $p_0 \in [0,1]$; $h_0 \leftarrow 0$; $r_{lcb} \leftarrow 0$; $\mathcal{H} \leftarrow [\,]$ ▷ Initialize wealth, successes, radius, and radius history (ordered list)
for $t = 1, 2, \ldots, N_{max} - N_{sel}$ **do**
  Draw $\epsilon_t \sim \mathcal{N}(0, \sigma^2 I)$; $X_t \leftarrow \mathbb{I}[f(x + \epsilon_t) = c_A]$.
  $h_t \leftarrow h_{t-1} + X_t$; $\hat{p}_{mle} \leftarrow h_t / t$.
  Compute current mixture wealth $\bar{W}_t(p_0)$ via Truncated Betas (Eq. 5) and update running max $W_t^{\max}(p_0) \leftarrow \max(W_{t-1}^{\max}(p_0), \bar{W}_t(p_0))$.
  if $t \pmod B == 0$ **or** $t = N_{max} - N_{sel}$ **then**
    $\underline{p_t} \leftarrow$ BrentSolver$(W_t^{\max}(p_0) = 1/\alpha)$; $r_{lcb} \leftarrow \sigma\Phi^{-1}(\max(\underline{p_t}, 0.5))$.
    $r_{mle} \leftarrow \sigma\Phi^{-1}(\max(\hat{p}_{mle}, 0.5))$; Append r_{lcb} to $\mathcal{H}$.
    if $W_t(0.5) \geq 1/\alpha$ and $\hat{p}_{mle} < 0.5$ and $t \geq 400$ **then Return** $r = 0$ ▷ UCB Exit
    else if $\hat{p}_{mle} < 0.5$ and $t \geq 400$ and ($\bar{W}_t(0.5) \leq \epsilon_{fail}$ or $E_{KT}(0.5) \leq \epsilon_{fail}$) **then Return** $r = 0$ ▷ Bankruptcy and reference bankruptcy exits
    else if $r_{lcb} > 0$ **then**
      $\epsilon_t \leftarrow [\epsilon_{start} - (\epsilon_{start} - \epsilon_{end}) \cdot \frac{t}{N_{max} - N_{sel}}] \cdot b(r_{mle})$.
      LookAhead $\leftarrow (r_{N_{max}}(\hat{h}_{N_{max}}) - r_{lcb}) \leq \epsilon_t \sigma$.
      Plateau $\leftarrow |\mathcal{H}| \geq 4$ **and** $(\mathcal{H}_{last} - \mathcal{H}_{last-3}) \leq \epsilon_t \sigma$.
      if *LookAhead* **or** *Plateau* **then Return** $r = r_{lcb}$
      end if
    end if
  end if
end for
Return $r = r_{lcb}$.

this paper uses this identical, verified `eval` split. The base models for MNIST and CIFAR-10 are trained ResNet-18s, while ImageNet employs the released ResNet-50 checkpoints from Cohen et al. (2019). Each meta-learner was trained on one CPU core: the Global champion's architecture takes a median of 42, 49 and 120 s for MNIST, CIFAR-10 and ImageNet (at most 170 s), and the median over all the architectures we trained is 69, 93 and 192 s (at most 12 minutes).

Differentiable Truncation via Series Expansion Training the meta-learner with a Kelly-optimal loss on truncated Beta distributions requires a differentiable implementation of the regularized incomplete Beta function, $I_x(a, b)$. As implementations of I_x are typically not compatible with automatic differentiation, we employ a 4^{th}-order Taylor series expansion to approximate the integral mass in the cases where derivatives are required. For a given support boundary $x \in [a_k, b_k]$, the log-mass is approximated as

$$\log I_x(a,b) \approx a \log x + b \log(1-x) - \log a - \log B(a,b) + \log\left(1 + \sum_{n=1}^{4} T_n\right) \quad , \tag{21}$$

where a and b here denote generic Beta shape parameters (i.e., (β_k, γ_k) or their posterior updates $(h + \beta_k, t - h + \gamma_k)$), not the truncation bounds a_k, b_k, and T_n corresponds to the Pochhammer ratios

$$T_1 = \frac{a+b}{a+1} x, \quad T_n = T_{n-1} \cdot \frac{a+b+n-1}{a+n} x \ . \tag{22}$$

To ensure numerical stability across the entire $[0, 1]$ probability range, we leverage the symmetry property $I_x(a, b) = 1 - I_{1-x}(b, a)$. When the truncation boundary is $x > 0.5$, the model computes the mass in the complement space using the same expansion, preventing the vanishing gradient issues associated with the high-probability tails of the Beta distribution.

Table 4: **Matched quality: samples to reach a fraction of Cohen-10k's ACR**, relative to Cohen-N (the smallest fixed N that reaches it; < 1: cheaper than Cohen-N). Sequential methods: upper envelope of their ϵ ladder ($\epsilon \in [0.0016, 1.61]\sigma$, look-ahead, horizon 10,000); Meta: the Global champion's architecture, median over 10 seeds; eval split. – = not reached within the horizon.

		Meta / Cohen-N				KT / Cohen-N			
Dataset	σ	80%	85%	90%	95%	80%	85%	90%	95%
MNIST	0.25	1.06	1.06	1.03	1.04	1.61	1.65	1.67	–
	0.5	1.15	1.15	1.08	1.08	1.65	1.67	1.64	1.69
	1	1.59	1.55	1.39	1.35	1.95	1.97	1.79	1.81
CIFAR-10	0.25	1.35	1.14	1.01	0.90	1.75	1.51	1.38	1.39
	0.5	1.59	1.30	1.10	0.94	1.92	1.64	1.38	1.36
	1	1.84	1.58	1.25	1.09	2.20	1.88	1.55	1.39
ImageNet	0.25	1.06	0.96	0.88	0.85	1.46	1.35	1.29	1.44
	0.5	1.17	0.99	0.85	0.79	1.50	1.31	1.19	1.27
	1	1.28	0.98	0.81	0.70	1.56	1.23	1.05	1.06

Crucially, this Taylor approximation is never used during online test-time certification. Test-time evaluation of mixture wealth $\bar{W}_t(p_0)$ and LCB root finding (Algorithm 2) employs an exact, high-precision binding. Consequently, the Taylor expansion introduces *zero numerical approximation error* into test-time LCB calculations or statistical coverage guarantees.

Certification For certifications, we set the error probability α to 0.001, corresponding to a 99.9% confidence interval. Unless otherwise stated, the selection glimpse took place over $N_{sel} = 100$, and similarly the checking interval B was also set to 100. All certifications were capped at $10,000$ max samples.

C.1 Radius-Specialized Stopping Dynamics

To demonstrate how anytime validity enables targeted computational allocation across difficulty regimes, we instantiate two specialists—**Small-R** ($r < r_{\text{small}}$) and **Large-R** ($r > r_{\text{large}}$)—built on an aggressive triage stopping schedule ($\epsilon_{start} = 0.30 \to \epsilon_{end} = 0.10$, aggression $\Delta = 1.2$, plateau lookback $L = 4$ checks at 5% relative growth). On CIFAR-10 ($\sigma = 1.0$), the regime thresholds $(r_{\text{small}}, r_{\text{large}}) = (0.68, 1.35)$ are fixed without test-set tuning as the held-out `meta_val` terciles of Cohen-10k certified radii. At each check t once $\underline{p_t} > 0.5$, precision termination evaluates

$$(R_{mle} - R_{lcb}) \leq \Delta \cdot \epsilon_t^{base} \cdot b_t \ , \tag{23}$$

where $R_{mle} = \sigma\Phi^{-1}(\hat{p}_{mle})$, $R_{lcb} = \sigma\Phi^{-1}(\underline{p_t})$, $R_{ucb} = \sigma\Phi^{-1}(\bar{p}_t)$, and the bias b_t is gated directly by the anytime-valid confidence interval $[R_{lcb}, R_{ucb}]$:

- **Small-R Specialist ($r < 0.68$):** While $R_{lcb} < 0.68$ (the lower bound has not yet cleared the low-radius regime), we set $b_t = b_{\text{in}} = 0.05$ and suppress the plateau exit, devoting extra iterations to tighten certificates near the boundary. Once $R_{lcb} \geq 0.68$, low-radius coverage is already guaranteed, so $b_t = b_{\text{out}} = 3.0$ (with plateau active) triggers rapid early exit on high-margin inputs.
- **Large-R Specialist ($r > 1.35$):** While $R_{ucb} > 1.35$ (the input can still plausibly achieve a large radius), we set $b_t = b_{\text{in}} = 0.15$ and suppress the plateau exit to maximize high-radius tightness. As soon as $R_{ucb} \leq 1.35$ rules out the target regime at level α, $b_t = b_{\text{out}} = 5.0$ triggers immediate early exit on low-margin inputs.

Because gating depends solely on $\mathcal{F}_t$-adapted bounds, Theorem 2 guarantees unconditional $(1 - \alpha)$ validity.

As was demonstrated in Figure 2 (Section 6.1), the generalist triage baseline achieves 47.17% certified accuracy (0.381 ACR) in 895 draws—outperforming KT ($1{,}015$ draws) and fixed-horizon Cohen-Matched at 895 draws (45.07% accuracy)—while Small-R and Large-R respectively attain overall ACRs of 0.348 and 0.424.

C.2 Computational Costs

Table 5: **Unit costs of certification** (H100, 2080 Ti; one CPU thread each). GPU: from Cohen-10k's live runs (median over the three σ); a step's time includes launching it, waiting for it and copying its predictions to the CPU. One input per step: the time of one step of the given size. Many inputs per step: each step draws the next B samples of as many inputs as fill a draw batch, and the time is per input per B draws. The fit uses the steps of 500 draws and the full steps: on small backbones the fixed cost of a step dominates a step of 100 draws. Precision: the timing benchmark, one step of the cache's draw batch ($10{,}100$; ImageNet $1{,}010$); emul.: no native bf16. The live runs use fp32 for MNIST and CIFAR-10 and bf16 for ImageNet. CPU: one check of the stopping rule, for KT and the Global champion (median over configurations and both priors), in the one-input runs with $B = 100$ and timed alone over the same inputs, with `scipy.special.ndtri` for the normal quantile (as run) or `scipy.stats.norm.ppf` (identical decisions); the meta-learner's forward pass, once per input after Phase I, alone or for many inputs at once. The machines' CPU rows come from different inputs (their own caches).

	H100			2080 Ti	
Part	MNIST	CIFAR-10	ImageNet	MNIST	CIFAR-10
GPU, one input per step: ms per step					
100 draws	1.58	1.59	8.88	2.30	2.33
500 draws	3.07	3.20	39.2	5.86	6.49
10,000 (ImageNet: 1,000) draws, the full step	42.5	46.0	76.8	93.1	105
GPU, many inputs per step: ms per input per B draws					
$B = 100$	0.42	0.46	7.71	0.94	1.05
$B = 500$	2.12	2.29	38.6	4.68	5.26
GPU, one input per step $\approx$ fixed + per draw $\times$ draws (fit)					
Fixed, ms per step	0.99	0.95	1.63	1.26	1.30
Per draw, μs	4.15	4.50	75.2	9.18	10.4
GPU, precision: μs per draw in one step of the benchmark's draw batch					
fp32	4.20	4.53	328	8.91	11.9
bf16	0.49	0.61	75.8	emul.	emul.
CPU, one thread: μs					
One check (`ndtri`), in the runs	95.0	108	119	120	141
One check (`ndtri`), timed alone	56.5	71.9	60.1	101	124
One check (`norm.ppf`), timed alone	223	224	217	415	404
Meta-learner forward pass, per input (one input)	460	457	683	509	502
Meta-learner forward pass, per input (many inputs)	167	165	534	190	183

While decreasing the sample count must intuitively increase the efficiency, some of these gains are offset by the wall-clock overhead introduced by MEV's decision-making logic. In practice, for large models, any additional overhead from MEV is significantly outweighed by the cost of forward passes. However, on numerically over-provisioned systems working with small models, our current systems GPU kernel launch costs incur more of an impact than for larger models—although this could be alleviated in production code. At inference time, the Meta-RS model evaluates the mean softmax over the $N_{sel} = 100$ Phase I draws, and optionally the clean image's embedding, followed by a single forward pass through the two-hidden-layer MLP to predict the image-specific prior. Once the prior is established, Brent-Dekker root finding is performed every B draws to evaluate the anytime-valid lower confidence bound.

We measured certification end to end on one H100, and on an RTX 2080 Ti workstation for MNIST and CIFAR-10. The GPU computes the noisy predictions a live run needs while certification proceeds check by check on one CPU thread (Table 6; Table 5 gives the unit cost of each component). Each sequential method is compared with Cohen-10k drawing its samples in batches of the same size under the same schedule. The schedules are evaluated for either one input at a time or many inputs per GPU step, with the checks either after each step or overlapped with the next.

For one ImageNet on an H100 (ResNet-50 in bf16), Cohen-10k takes 0.78 s when it draws its $10{,}000$ samples in batches of $1{,}000$. The Global champion, checking every 100 draws, takes 0.62 / 0.52 / 0.40 s at $\sigma = 0.25$ / 0.5 / 1: 20–49% less, or 32–55% less than Cohen-10k drawing in the same batches of 100 (Table 6). It draws $6{,}845$ / $5{,}822$ / $4{,}429$ samples, and a step of 100 costs the ResNet-50 only 15% more per sample than a step of $1{,}000$ (Table 5). The checks and the meta-learner's forward pass take 1.4–1.5% of the time. However, it is also possible to saturate GPU vram by certifying many inputs together. Under these conditions the champion is 1.45–$2.23\times$ faster than Cohen-10k sampling the same way, at least 97% of its saving in samples.

Table 6: **Time per input of live certification on one H100**, ms (eval: the first 300 MNIST, 300 CIFAR-10, 100 ImageNet inputs per σ; MNIST fp32, CIFAR-10 fp32, ImageNet bf16; Meta: the Global champion, Meta-10-Margin-Full, seed 0). Draws and checks: per input; Cohen-10k draws $10{,}100$ (Phase I included). One input at a time: each GPU step draws the next B samples of one input, then the CPU checks them (Cohen-10k draws in steps as large as the GPU takes: MNIST $10{,}000$, CIFAR-10 $10{,}000$, ImageNet $1{,}000$). GPU: launching the steps, waiting for them and copying the predictions back; Checks: the stopping rule on one CPU thread (`scipy.special.ndtri`); Meta-l.: the meta-learner's forward pass, once per input; Total: their sum. Many inputs at once: each GPU step draws the next B samples of as many inputs as fill a draw batch, then the CPU checks them; CPU: checks and forward passes; Overlap: the total when the next step is queued before the checks, which hides them (an input that stops wastes the batch queued for it). $\times$: Cohen-10k's time over Meta's; one input at a time against Cohen-10k in full steps, many inputs at once against Cohen-10k drawing in the same steps with overlap (like for like). Samples at other intervals: Table 7; every schedule, and KT: Table 8.

					One input at a time					Many inputs at once				
Dataset	σ		Draws	Checks	GPU	Checks	Meta-l.	Total	$\times$	GPU	CPU	Total	Overlap	$\times$
MNIST	0.25	Cohen-10k	10,100	–	43.9	–	–	44.1		42.7	–	42.8	42.7	
		Meta, $B = 100$	9,460	93.6	152	7.86	0.47	160	0.28	39.8	4.54	44.3	40.1	1.06
		Meta, $B = 500$	9,900	19.6	61.9	1.70	0.45	64.0	0.69	41.6	1.19	42.8	41.7	1.02
	0.5	Cohen-10k	10,100	–	43.9	–	–	44.0		42.7	–	42.8	42.7	
		Meta, $B = 100$	8,240	81.4	132	7.79	0.46	140	0.31	34.9	4.83	39.7	35.1	1.22
		Meta, $B = 500$	8,868	17.5	55.5	1.75	0.45	57.7	0.76	37.2	1.35	38.5	37.9	1.13
	1	Cohen-10k	10,100	–	43.9	–	–	44.1		42.7	–	42.8	42.7	
		Meta, $B = 100$	5,266	51.7	84.5	5.95	0.46	90.9	0.48	22.4	4.15	26.5	22.6	1.89
		Meta, $B = 500$	6,053	11.9	38.3	1.43	0.45	40.1	1.10	25.3	1.21	26.5	27.2	1.57
CIFAR-10	0.25	Cohen-10k	10,100	–	47.4	–	–	47.6		46.3	–	46.4	46.3	
		Meta, $B = 100$	6,183	60.8	99.4	6.27	0.57	106	0.45	28.5	4.19	32.7	28.8	1.61
		Meta, $B = 500$	6,875	13.6	45.1	1.44	0.45	47.0	1.01	31.3	1.21	32.5	32.8	1.41
	0.5	Cohen-10k	10,100	–	47.4	–	–	47.6		46.2	–	46.3	46.3	
		Meta, $B = 100$	5,002	49.0	80.4	5.36	0.46	86.2	0.55	23.1	3.75	26.9	23.4	1.98
		Meta, $B = 500$	5,700	11.2	37.6	1.29	0.45	39.3	1.21	25.9	1.10	27.0	27.8	1.66
	1	Cohen-10k	10,100	–	47.4	–	–	47.6		46.2	–	46.4	46.2	
		Meta, $B = 100$	3,894	37.9	62.6	4.22	0.45	67.2	0.71	18.0	2.98	21.0	18.3	2.53
		Meta, $B = 500$	4,393	8.59	29.2	1.01	0.45	30.7	1.55	19.9	0.91	20.8	22.0	2.11
ImageNet	0.25	Cohen-10k	10,100	–	777	–	–	778		780	–	780	779	
		Meta, $B = 100$	6,845	67.5	610	7.98	0.73	619	1.26	529	5.16	534	536	1.45
		Meta, $B = 500$	7,370	14.5	579	1.77	0.74	581	1.34	569	2.00	571	592	1.32
	0.5	Cohen-10k	10,100	–	777	–	–	777		779	–	779	778	
		Meta, $B = 100$	5,822	57.2	517	6.99	0.68	525	1.48	450	4.59	454	457	1.70
		Meta, $B = 500$	6,555	12.9	515	1.60	0.63	518	1.50	507	2.09	509	533	1.46
	1	Cohen-10k	10,100	–	777	–	–	777		779	–	779	778	
		Meta, $B = 100$	4,429	43.3	393	5.35	0.62	399	1.95	342	3.81	346	350	2.23
		Meta, $B = 500$	5,100	10.0	401	1.27	0.67	403	1.93	394	1.62	396	428	1.82

Lightweight backbones A check costs the same whatever the backbone: 46–76 μs on one CPU thread, for the KT prior as for the Global champion. A ResNet-18 prediction costs about 4 μs at full batch, so on small models the checks are no longer negligible. Measured live (Tables 5 and 6), with a check every 100 draws they take 5–7% of the time when inputs are certified one at a time, and 10–15% when 100 inputs share each GPU step and the GPU waits for them; queueing the next GPU step before the checks hides most of this. With a check every 500 draws they take 2–4%. Against Cohen-10k drawing in batches of the same size, the Global champion is 1.06–$1.89\times$ faster on MNIST and 1.61–$2.53\times$ faster on CIFAR-10 when many inputs are certified at once, at least 97% of its saving in samples (at least 96% on an RTX 2080 Ti). What small backbones do cost is single-input latency. A GPU step has a fixed cost of about 1 ms on top of 4–5 μs per draw, so a step of 100 images costs nearly four times as much per draw as one of $10{,}000$; a sequential method takes one step per check, while Cohen-10k can take one step in all. One input at a time with a check every 100 draws is therefore only 0.28–$0.71\times$ as fast as Cohen-10k drawing all its samples at once, and 0.69–$1.55\times$ with a check every 500 draws, at the cost in samples shown in Table 7. On the slower 2080 Ti the per-draw cost weighs more against the fixed cost, and one input at a time is 0.42–$1.11\times$ as fast at 100 draws per check.

Check interval A run can stop only at a check, so checking every B draws trades checks for samples (Table 7). Every 500 draws instead of 100 means a fifth of the checks for 5–15% more samples. Every $1{,}000$ draws means a tenth of the checks for 6–29% more. The ACR rises slightly in both cases, by at most 0.4%. Since the checks take at most 15% of the time at $B = 100$ (Table 6), the

Table 7: **Check interval** (eval, Global rule). The Global champion (Meta-10-Margin-Full) certifies with the interval roots computed every B Phase II draws instead of 100. N/N_{100}, checks/checks$_{100}$ and ACR/ACR$_{100}$: average samples, checks and ACR relative to $B = 100$ for the same seed, median over 10 seeds, range over the three σ; N/N_{KT}: relative to KT at the same B. The wall clock of $B = 100$ and 500 is measured in Table 6.

Dataset	B	N/N_{100}	Checks/checks$_{100}$	ACR/ACR$_{100}$	N/N_{KT}
MNIST	100	1.000	1.00	1.000	0.895–0.992
	200	1.003–1.053	0.50–0.53	1.000–1.001	0.906–0.992
	500	1.048–1.153	0.21–0.23	1.002–1.004	0.921–0.993
	1000	1.055–1.258	0.11–0.13	1.003–1.004	0.938–0.995
CIFAR-10	100	1.000	1.00	1.000	0.906–0.932
	200	1.034–1.056	0.52–0.53	1.001	0.915–0.940
	500	1.112–1.152	0.22–0.23	1.002–1.003	0.923–0.948
	1000	1.186–1.292	0.12–0.13	1.003–1.004	0.942–0.959
ImageNet	100	1.000	1.00	1.000	0.931–0.959
	200	1.021–1.039	0.51–0.52	1.000–1.001	0.939–0.964
	500	1.082–1.125	0.22–0.23	1.003	0.948–0.968
	1000	1.129–1.232	0.11–0.12	1.004	0.962–0.976

protocol's $B = 100$ is the faster choice whenever many inputs are certified together, taking 4–15% less time than $B = 500$. A longer interval helps only when inputs are certified one at a time on a small backbone: there $B = 500$ is 2.2–2.5× faster per input than $B = 100$, because each GPU step is five times larger. On the ResNet-50 the two are within 7%.

D Champion Selection Processes

To evaluate the operational versatility of our framework, we identify three distinct *Champion* archetypes. Each represents a specific optimization of our Meta-learner architecture designed to address different real-world deployment constraints. In short, the Global Champion demonstrates generalisability of the meta-learner, and provides a universal speedup for standard certification; while the Specialists demonstrate the framework's ability to maximize performance.

Selection is conducted on the `meta_val` split (median over 10 seeds) from the grid $K \in \{1, 3, 6, 10\} \times \{\text{entropy}, \text{margin}\} \times \{\text{full}, \text{hybrid}, \text{dynamic}\} \times \epsilon \in \{0.0126, \ldots, 0.4032\}\sigma$ under the look-ahead policy:

- **Accuracy Champion:** The configuration minimizing average samples subject to ACR $\geq 0.95 \times$ ACR$_{\text{Cohen-10k}}$ and certified accuracy within 1.0 pp of Cohen-10k.
- **Efficiency Champion:** The configuration minimizing average samples subject to ACR $\geq 0.85 \times$ ACR$_{\text{Cohen-10k}}$ and certified accuracy within 2.0 pp of Cohen-10k.
- **Global Champion:** A single architecture and single ϵ/σ selected across all nine dataset and noise configurations to minimize mean samples while satisfying the accuracy constraint in all nine: **Meta-10-Margin-Full at** $\epsilon = 0.0126\sigma$.

Table 9 summarizes the resulting champion configurations.

D.1 Reference Bankruptcy Dynamics across Champions

Borderline uncertifiable inputs ($p_A \approx 0.45$–0.52) present a unique challenge for meta-learned priors: because the meta-learner correctly identifies uncertainty near the decision boundary, its wealth $\bar{W}_t(0.5)$ accumulates or decays at a low rate. Without the reference bankruptcy exit, the efficiency champions at $\sigma = 1.0$ spent 4.5× as many samples as KT on inputs that KT safely abstains on via bankruptcy, and approximately one in six of these inputs stalled until the 10,000-sample horizon. Introducing the concurrent KT reference bettor $E_{KT}(0.5) = m_{KT}(h_t, t)/0.5^t$ resolves this stalling without altering certification validity. Across all 26,040 evaluated configurations (spanning every architecture, stopping rule, and seed in our grid), no configuration requires more samples with the exit enabled; the median configuration requires 8% fewer samples (up to 42% reduction on boundary-heavy regimes), sample expenditures on abstentions fall by 5–64%, the average certified radius shifts by less than 10^{-4}, and certified accuracy changes by at most 0.3 pp.

Table 8: **Wall-clock time of live certification on one H100 80GB HBM3** (**M**: MNIST, first 300 eval inputs, fp32; **C-10**: CIFAR-10, first 300 eval inputs, fp32; **INet**: ImageNet, first 100 eval inputs, bf16; per σ; Meta: Global champion, Meta-10-Margin-Full, seed 0; range over the three σ). Certification runs check by check (every decision is the evaluation's) while the GPU computes the noisy predictions a live run needs. Serial: one input at a time, each batch of B draws checked once it is in; Overlap: the next batch queued before the check; Batched: the next B draws of as many inputs as fill one draw batch, in one GPU step, all checked after it; Batch. + ovlp.: the next step queued before the checks. Speedups are like for like: over Cohen-10k drawing in batches of the same B under the same schedule. Cohen-10k rows: its time in full draw batches over its time in batches of B, the cost of drawing in small batches. 1-input: the better of Serial and Overlap against Cohen-10k drawing its $10{,}000$ in batches as large as the GPU takes (the draw batch: MNIST $10{,}000$, CIFAR-10 $10{,}000$, ImageNet $1{,}000$), one input at a time (single-input latency; a sequential method can only draw up to its next check). Samples: Cohen-10k's draws over the method's, the ceiling for a like-for-like speedup. Meta's times include the meta-learner's forward pass, once per input after Phase I (0.1–1.5% of the time). μs/check: one check on one CPU thread, timed alone over the same inputs (in brackets: with `scipy.stats.norm.ppf` in place of `scipy.special.ndtri` for the normal quantile; the decisions are identical).

Data	Prior	B	Samples	Serial	Overlap	Batched	Batch. + ovlp.	1-input	μs/check
M	Cohen-10k	100	1.00	0.28	0.29	1.03	1.03	0.29	–
	Cohen-10k	500	1.00	0.70	0.71	1.03	1.03	0.71	–
	KT	100	1.06–1.71	0.99–1.57	1.04–1.69	0.96–1.45	1.05–1.69	0.30–0.49	57 (224)
	KT	500	1.01–1.55	0.98–1.48	1.01–1.44	1.00–1.52	1.01–1.47	0.72–1.04	57 (224)
	Meta	100	1.07–1.92	1.00–1.75	1.06–1.88	0.97–1.61	1.06–1.89	0.30–0.54	56 (223)
	Meta	500	1.02–1.67	0.98–1.57	1.02–1.53	1.00–1.61	1.02–1.57	0.72–1.10	56 (223)
C-10	Cohen-10k	100	1.00	0.29–0.30	0.30–0.31	1.03	1.03	0.30–0.31	–
	Cohen-10k	500	1.00	0.72–0.73	0.73–0.74	1.03	1.03	0.73–0.74	–
	KT	100	1.53–2.37	1.43–2.20	1.50–2.31	1.34–2.03	1.50–2.28	0.45–0.71	71 (224)
	KT	500	1.39–2.12	1.34–2.02	1.32–1.90	1.36–2.07	1.34–1.96	0.97–1.46	71 (224)
	Meta	100	1.63–2.59	1.53–2.38	1.63–2.53	1.42–2.21	1.61–2.53	0.49–0.77	72 (224)
	Meta	500	1.47–2.30	1.40–2.14	1.39–2.04	1.42–2.22	1.41–2.11	1.02–1.55	72 (224)
INet	Cohen-10k	100	1.00	0.86–0.87	0.87	1.00	1.00	0.87	–
	Cohen-10k	500	1.00	0.98	0.98	1.00	1.00	0.98	–
	KT	100	1.41–2.11	1.40–2.08	1.40–2.06	1.40–2.08	1.39–2.06	1.21–1.80	60 (217)
	KT	500	1.31–1.88	1.31–1.87	1.26–1.74	1.31–1.87	1.26–1.74	1.28–1.83	60 (217)
	Meta	100	1.48–2.28	1.46–2.25	1.47–2.23	1.46–2.25	1.45–2.23	1.27–1.95	60 (217)
	Meta	500	1.37–1.98	1.37–1.97	1.32–1.82	1.37–1.97	1.32–1.82	1.34–1.93	60 (217)

E Detailed Results and Aggregated Performance

The full suite of certification and efficiency results for individual datasets and noise levels is provided in Figures 3 through to 10. These figures are included to demonstrate the framework's consistent performance across three distinct data manifolds (MNIST, CIFAR-10, ImageNet) and various noise levels. By visualizing the *Average Sample Count* (dashed lines) alongside the *Certified Accuracy* (solid lines), we provide empirical proof that the meta-learned priors effectively deliver 1.08–$2.58\times$ speedups in the High-Fidelity regime (and 7.8–$11.6\times$ under the Maximum-Efficiency plateau exit) over Cohen et al. (2019) while maintaining a similar accuracy profile. When comparing against the KT prior, our Meta-learner is able to reduce the required number of samples to both certify and reject samples in the majority of settings, although in some settings the KT prior remains more efficient (e.g., the Global Champion for ImageNet at $\sigma = 0.5$, and MNIST and CIFAR-10 at $\sigma = 1.0$).

It is also important to note that that in several configurations (e.g., ImageNet at $\sigma = 0.5$) Cohen-1k attains a higher certified accuracy and mean radius than the meta-learned certifiers at a comparable average sample count. However, we stress that *a priori* knowledge of this number of samples is impossible, and a fixed-horizon certification cannot support early rejection or application-specific stopping conditions. This serves as the primary evidence for the system's operational viability in high-throughput environments.

Table 9: **Champion configurations by dataset and noise level**, selected on meta_val (median over 10 seeds) from the grid $K \in \{1, 3, 6, 10\} \times$ {entropy, margin} $\times$ {full, hybrid, dynamic} $\times$ $\epsilon \in \{0.0126, 0.0252, \ldots, 0.4032\}\sigma$ (look-ahead form, $\Delta = 1.0$). Accuracy: the fewest samples with ACR $\geq 0.95\times$ Cohen-10k's and accuracy within 1 pp of it; Efficiency: ACR $\geq 0.85\times$ and within 2 pp; Global: one architecture and one ϵ (in units of σ) for every configuration, minimising the mean samples subject to the accuracy constraint in all nine: Meta-10-Margin-Full at $\epsilon = 0.0126\sigma$. The meta-learner is trained separately for each configuration.

		Efficiency champion		Accuracy champion	
Dataset	σ	Architecture	ϵ/σ	Architecture	ϵ/σ
MNIST	0.25	Meta-10-Entropy-Hybrid	0.1008	Meta-1-Entropy-Hybrid	0.0252
	0.5	Meta-3-Entropy-Hybrid	0.1008	Meta-3-Entropy-Hybrid	0.0252
	1	Meta-3-Margin-Full	0.0504	Meta-10-Entropy-Dynamic	0.0126
CIFAR-10	0.25	Meta-3-Entropy-Hybrid	0.1008	Meta-10-Margin-Dynamic	0.0252
	0.5	Meta-1-Entropy-Hybrid	0.0504	Meta-1-Margin-Hybrid	0.0252
	1	Meta-10-Margin-Dynamic	0.0504	Meta-3-Margin-Full	0.0126
ImageNet	0.25	Meta-3-Margin-Full	0.1008	Meta-1-Margin-Dynamic	0.0252
	0.5	Meta-6-Entropy-Dynamic	0.1008	Meta-6-Margin-Dynamic	0.0252
	1	Meta-1-Entropy-Dynamic	0.1008	Meta-1-Margin-Dynamic	0.0252

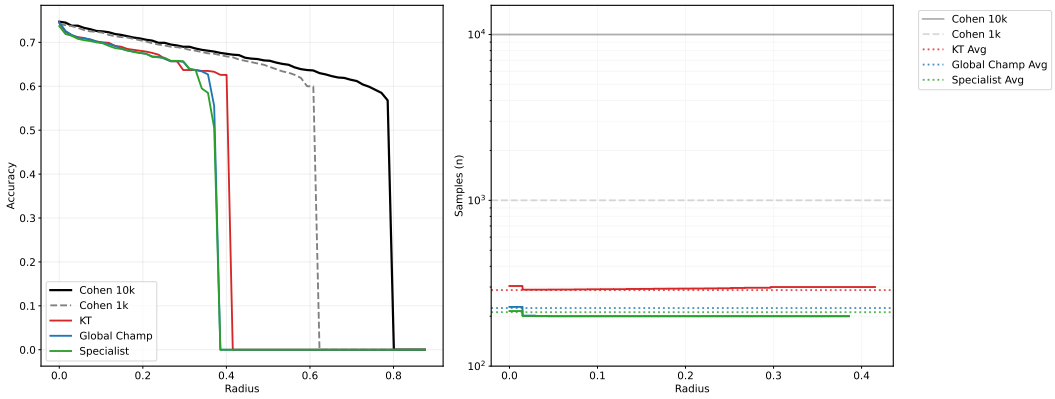


Figure 3: Certification and efficiency results for ImageNet at $\sigma = 0.25$. Curves compare the Anytime-Valid Global Champ and Specialist Meta-models against Cohen-10k/1k and KT Prior baselines.

E.1 Early-Rejection Only Ablation

To isolate the contribution of our UCB and dual-bankruptcy martingales from early certification, we evaluate early rejection in isolation: precision and plateau exits are disabled, so an input runs to the full 10,000-draw horizon unless a rejection exit fires first. Table 12 reports performance for the Efficiency champions, and Table 13 reports performance for the Accuracy champions across all nine benchmark configurations (median over 10 seeds). In both cases, uncertifiable inputs are aborted in 943–1,835 samples (5.5–$10.7\times$ faster than standard RS) with a false rejection rate under 1.4%, confirming that test martingale bankruptcy safely identifies non-robust inputs with near-zero false alarms.

Empirical Validity Check We evaluate the empirical soundness of our anytime-valid lower confidence bounds by comparing every certificate's bound against the true probability p_A estimated from all 100,100 cached draws of the same input (Section G). Across 252,863 certificates issued on the evaluation split by KT, Cohen-10k, and the Global champion (spanning ten seeds and nine configurations), only 36 lower bounds (0.014%) exceeded this high-draw estimate. None exceeded it by more than three standard errors of the estimate itself (at most 0.0018), and 4 of the 36 marginal discrepancies occurred in Cohen-10k's standard Clopper-Pearson baseline. Two practical limitations apply: the estimate shares its first 10,100 draws with the certification, and inputs with p_A very close to 1 (above 0.999 for 93% of MNIST inputs at $\sigma = 0.25$) cannot show a violation since a bound from at most 10,100 draws lies at least 0.0006 below them. This check therefore rules out gross violations,

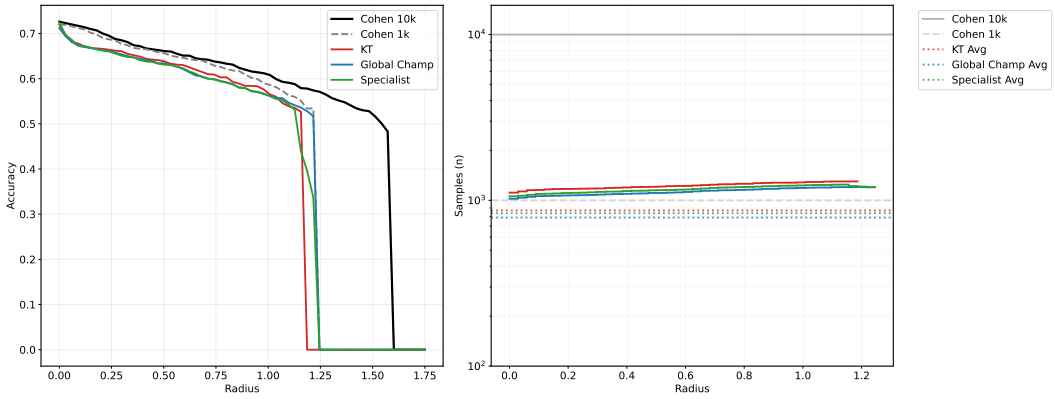


Figure 4: Certification and efficiency results for ImageNet at $\sigma = 0.5$. Curves compare the Anytime-Valid Global Champ and Specialist Meta-models against Cohen-10k/1k and KT Prior baselines.

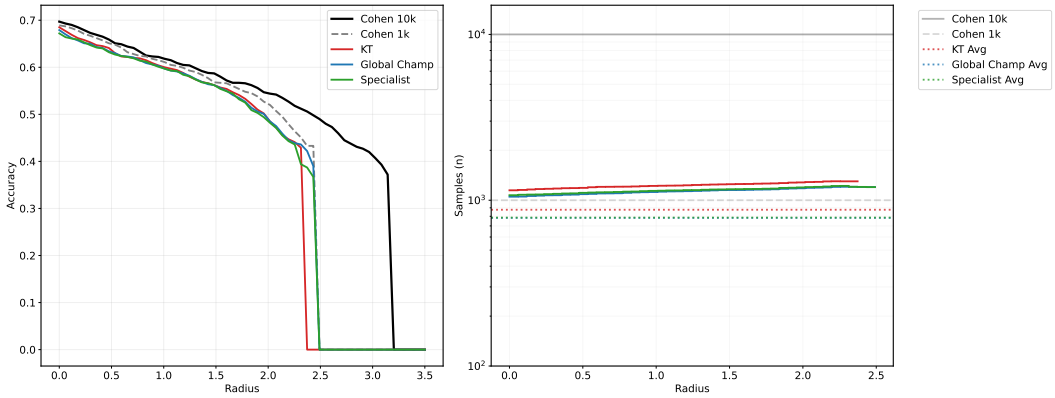


Figure 5: Certification and efficiency results for ImageNet at $\sigma = 1.0$. Curves compare the Anytime-Valid Global Champ and Specialist Meta-models against Cohen-10k/1k and KT Prior baselines.

confirming that test martingale inversion operates strictly within theoretical error bounds with zero abnormal failure modes.

F Ablation Studies and Sensitivity Analysis

This section provides empirical justification for the architectural and heuristic choices in the Meta-RS framework. All studies are conducted using ImageNet at $\sigma = 0.25$ unless otherwise specified.

Zero-Shot Cross-Dataset Transferability To evaluate cross-domain generalization, Table 14 reports the performance of Meta-Learners trained on MNIST or CIFAR-10 deployed zero-shot on ImageNet. To ensure architecture- and dataset-agnostic transfer between models, the meta-learner model uses scalar Phase I glimpse statistics—the margin of the mean Phase I softmax and the glimpse hit rate $\hat{p}_{sel}$—without the penultimate-layer embedding $\phi(x)$, to eliminate dimensional dependencies. Across all experiments, zero-shot evaluation of an MNIST- or CIFAR-10-trained Meta-Learner on ImageNet achieves 98.0% to 99.6% (and 100% median) decision agreement with native ImageNet models while maintaining statistical validity at level $1 - \alpha$ under Theorem 1. Transferring across domains requires only between 0.8% and 2.4% more samples than native models, remaining up to $2.21\times$ faster than Cohen-10k when trained on CIFAR-10, and $2.04\times$ faster when transferred from MNIST. Figure 11 visualizes the log-log correlation between termination latencies, demonstrating that the meta-learner captures fundamental statistical properties of classifier conviction that generalize across architectures.

Component-Wise Efficiency Attribution Figure 12 attributes the impact of each stopping component for ImageNet at $\sigma = 0.25$. Look-ahead stopping by itself requires an average of $7{,}780$ samples,

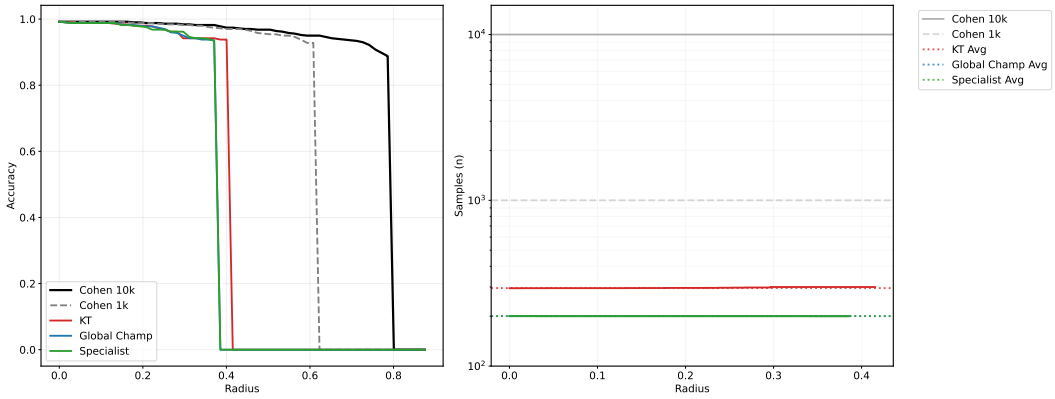


Figure 6: Certification and efficiency results for MNIST at $\sigma = 0.25$. Curves compare the Anytime-Valid Global Champ and Specialist Meta-models against Cohen-10k/1k and KT Prior baselines.

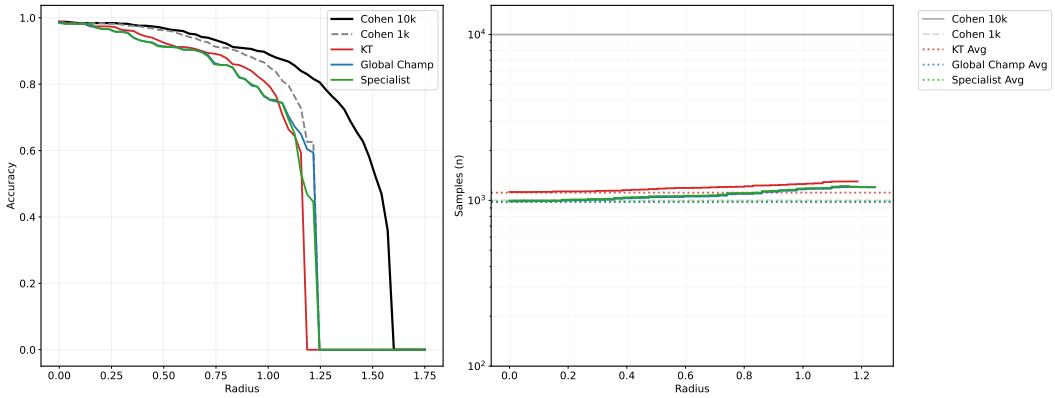


Figure 7: Certification and efficiency results for MNIST at $\sigma = 0.5$. Curves compare the Anytime-Valid Global Champ and Specialist Meta-models against Cohen-10k/1k and KT Prior baselines.

decreasing to 7,280 when introducing the UCB exit. Adding reference bankruptcy further brings this down to 7,054 samples, where the savings are concentrated entirely on uncertifiable samples (6,976 $\rightarrow$ 4,621). Activating the plateau exit further brings down the overall sample demand to 1,165 draws, at the cost of diminished ACR—demonstrating again the value of practitioners navigating the trade-offs between the high-fidelity and maximum-efficiency regimes within production systems.

The Precision-Efficiency Pareto Frontier Figure 13 maps sample complexity against radius precision target ϵ. As the requirement for tightness increases ($\epsilon \rightarrow 0.01$), the sample cost grows approximately as a power law (linear on log-log axes), confirming that practitioners can reliably trade sample budgets for certified radii, or relax precision for high-throughput operational deployments.

F.1 Parameter Influence

To consider parameterisation, we consider meta-model variants across components K, feature modes, support ranges, and stopping tolerances ϵ/σ (Table 15). Across the full evaluation grid, the architectural choices ($K \in \{1, 3, 6, 10\}$, Entropy vs. Margin, and Dynamic, Full, or Hybrid support) achieve remarkably consistent performance, requiring on average $0.84\times$ of the samples of the KT prior at identical certified accuracy. The primary driver of the trade-off between sample complexity and certified radius is the stopping tolerance ϵ/σ, which smoothly scales relative sample demand from $0.76\times$ to $0.94\times$ of KT.

Ablation on Penultimate Layer Embeddings vs. Glimpse Signal To isolate the utility of the classifier's penultimate-layer embedding $\phi(x)$, we retrain the Global champion (Meta-10-Margin-Full) for all ten seeds with subsets of its inputs and certify ImageNet at $\sigma = 0.5$ under the Global

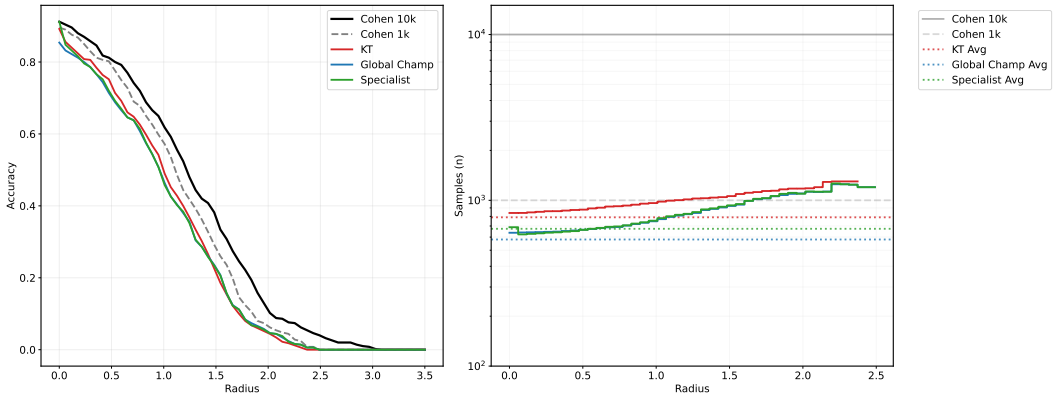


Figure 8: Certification and efficiency results for MNIST at $\sigma = 1.0$. Curves compare the Anytime-Valid Global Champ and Specialist Meta-models against Cohen-10k/1k and KT Prior baselines.

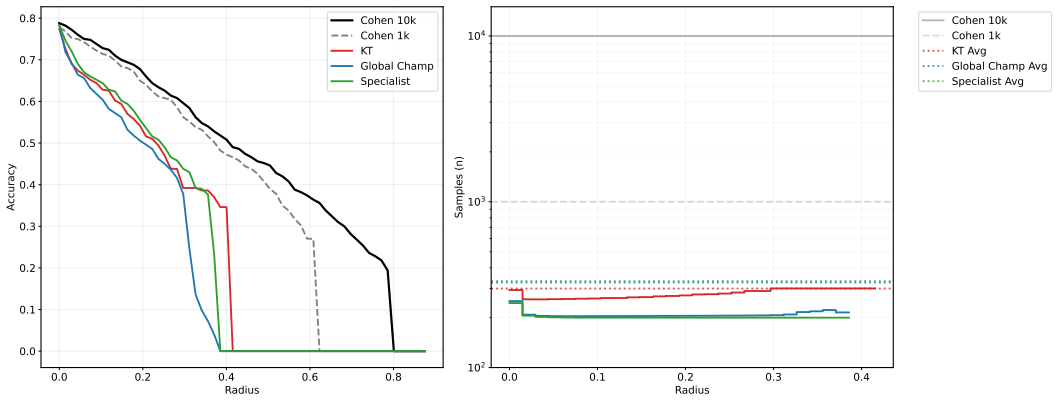


Figure 9: Certification and efficiency results for CIFAR10 at $\sigma = 0.25$. Curves compare the Anytime-Valid Global Champ and Specialist Meta-models against Cohen-10k/1k and KT Prior baselines.

rule (Table 16). Removing $\phi(x)$ does not hurt: with only the Phase I softmax proxy and $\hat{p}_{sel}$ the meta-learner needs 5,897 samples against 5,935 with $\phi(x)$, at the same ACR; dropping $\hat{p}_{sel}$ as well costs 1.3% more. Every variant, including the clean-image softmax of the original method and ϕ averaged over the Phase I draws, needs 4.5–6.4% fewer samples than the KT prior, at a higher ACR. In this setting the benefit of the learned prior comes from the Phase I statistics rather than from the embedding, and the clean-image forward pass for $\phi(x)$ can be dropped.

F.2 Meta-Learner Design

Tables 17 and 18 vary the Global champion's depth, safety-anchor weight and training data; each variant is trained for all ten seeds in every configuration and run under the Global rule. Depth does not matter: with no hidden layer (the heads read the input projection directly), one or four, the samples stay within 0.6% of the champion's two layers, at the same ACR. Nor does the anchor weight with the native prior: 0 and 0.001 are within 0.2% of 0.01, and 0.1 costs up to 0.5% more. The training data can be cut substantially. Truncating each image's Phase II stream from 10,000 to 1,000 draws changes nothing, and 500 images instead of 5,000 (10,000 for ImageNet) cost 0.1–5.5% more samples, still fewer than KT for every seed. With 1,000 images and 1,000-draw streams, generating the training data takes 0.35% of the GPU compute of training the ImageNet classifier, at 2.7–3.2% more samples; for MNIST and CIFAR-10, whose classifiers train in minutes, it takes 9% and 3.6%. Training the meta-learner itself takes at most 190 s on one CPU core for every variant in Table 18.

The anchor matters when the prior is misspecified (Table 19). Trained on MNIST and used on ImageNet, the prior without the anchor certifies a smaller median average radius than KT at every σ, and at $\sigma = 0.25$ one of its ten seeds never left its initialisation in training (its validation loss did not

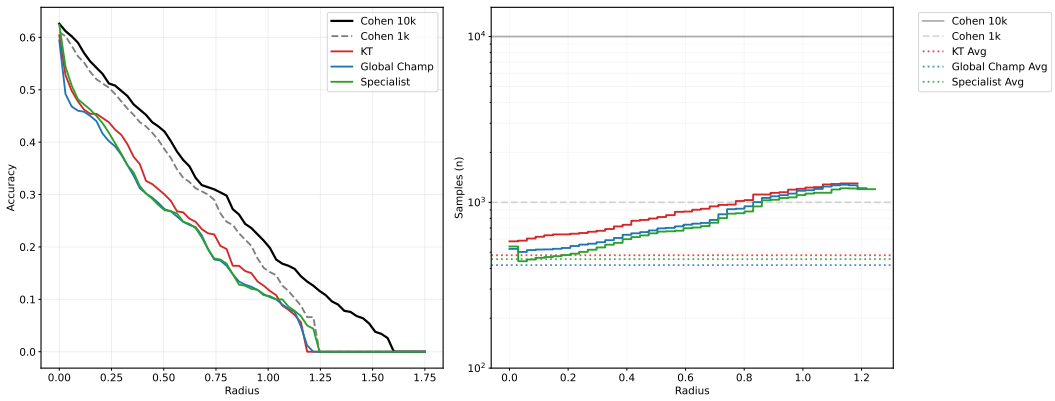


Figure 10: Certification and efficiency results for CIFAR10 at $\sigma = 0.5$. Curves compare the Anytime-Valid Global Champ and Specialist Meta-models against Cohen-10k/1k and KT Prior baselines.

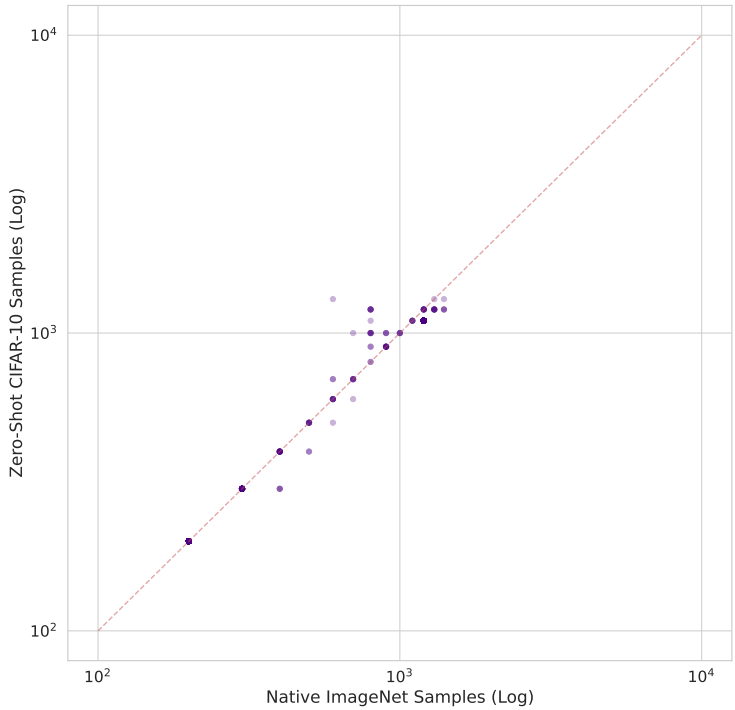


Figure 11: **Zero-Shot Transferability.** A log-log scatter plot correlating termination latencies of an ImageNet-native prior vs. a CIFAR-trained prior tested on ImageNet. These results suggest that the meta-learner captures universal statistical properties of classifier conviction that transcend specific datasets.

change from the first epoch) and certified no input on any of the three datasets. With the anchor, the same seed trains normally, and every seed stays within 1.1% of KT's radius and needs at most 3.3% more samples than KT. Priors trained on CIFAR-10 are insensitive to the anchor on every target. We have not verified the mechanism; a likely one is that the KT component keeps the training loss finite, and its gradient informative, when the learned components put no mass near the true p.

G Reallocating a Fixed Sampling Budget

Cohen et al. (2019) draw the same number of Monte Carlo samples N for every input. Many inputs do not need them: an input whose selected class c_A has p_A near or below $\frac{1}{2}$ will never receive a useful radius, while for an input with p_A near 1 the certified radius keeps growing with N. We therefore ask a different question from early stopping: *for the same total budget*, $N_{eval} \times B$ Phase II draws, can a

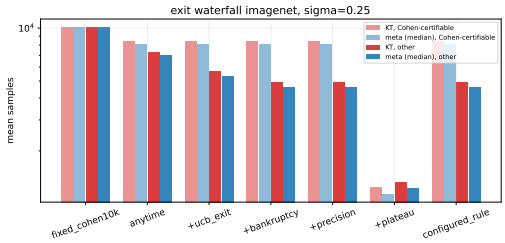


Figure 12: **Component-Wise Efficiency Attribution Waterfall** (ImageNet, $\sigma = 0.25$). Stepwise reduction in sample complexity from Look-Ahead stopping alone ($7,780$ draws), to adding the UCB exit ($7,280$), Reference Bankruptcy ($7,054$), and the Maximum-Efficiency Plateau exit ($1,165$). Bankruptcy savings concentrate exclusively on uncertifiable samples ($6{,}976 \rightarrow 4{,}621$).

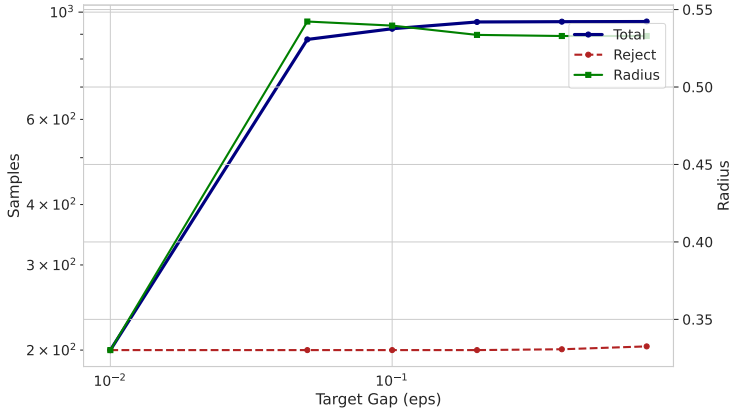


Figure 13: **Precision-Efficiency Pareto Frontier.** This plot maps the scaling of sample complexity against the radius precision target ϵ. As the requirement for tightness increases ($\epsilon \rightarrow 0.01$), the sample cost grows approximately as a power law (linear on log-log axes), allowing practitioners to choose an operating point based on their computational budget.

policy that decides per input how many draws to spend certify a larger average radius than Cohen-B, and where should the samples go?

Where samples buy radius. With the one-sided Clopper-Pearson bound at level α, an input whose n draws all vote for c_A is certified at $\sigma\Phi^{-1}(\alpha^{1/n})$: at $\alpha = 0.001$ this is 2.46σ after 10^3 draws, 3.20σ after 10^4 and 3.81σ after 10^5, with no ceiling. An input with p_A close to $\frac{1}{2}$ needs about $(z_{1-\alpha}/2)^2/(p_A - \frac{1}{2})^2$ draws before it can be certified at all, and its radius is capped at $\sigma\Phi^{-1}(p_A)$ (for example 0.25σ at $p_A = 0.6$). Samples near $\frac{1}{2}$ therefore buy certified accuracy (an abstention becomes a small certificate), and samples near 1 buy radius.

Allocation policies. Every policy observes the $N_{sel} = 100$ Phase I draws of each input, then fixes $n_i \in \{0, 20, 50, \ldots, 10^5\}$ Phase II draws per input (0 means abstain) so that $\sum_i n_i = N_{eval}B$:

- **Uniform** (Cohen-B): $n_i = B$.
- **Expected-radius allocation** under a belief Q_i about p_A maximises the expected sum of radii,

$$\max_{n_1,\ldots,n_{N_{eval}}} \sum_i \mathbb{E}_{p\sim Q_i}\, \mathbb{E}_{k\sim \mathrm{Bin}(n_i,p)}\left[r(k,n_i)\right] \quad \text{subject to} \quad \sum_i n_i = N_{eval}B, \tag{24}$$

  where $r(k,n) = \Phi^{-1}(\underline{p_A})$ for the Clopper-Pearson bound $\underline{p_A} > \frac{1}{2}$ and 0 otherwise. This is a knapsack. We solve its Lagrangian relaxation: each input independently maximises $\mathbb{E}\, r(n) - \lambda n$, and λ is found by bisection so that the budget is spent. We use two beliefs. *Phase I votes* is the Beta posterior of the 100 Phase I votes. *Meta-learned* is the prior $Q(\theta)$ that $\mathcal{M}_\theta$ predicts from $\phi(x)$ and the Phase I mean softmax.
- **Pilot + tilt** γ: a pilot of $0.2B$ draws per input with futility exits from 100 draws, then the rest shared in proportion to $e^{\gamma u}$, where $u = \Phi^{-1}(\hat{p})$ is rescaled to $[0,1]$. $\gamma < 0$ favours inputs near $p = \frac{1}{2}$ and $\gamma > 0$ inputs near $p = 1$.

- **Bounds**: *hindsight* gives each input its best n knowing every input's realised radius at every n, and the *true-p oracle* allocates by expected radius at each input's 10^5-draw estimate of p_A. Neither is a procedure; they show how much reallocation could gain.

Validity. With the Clopper-Pearson certificate, each input's bound uses only Phase II draws whose number n_i was fixed before they were drawn, from Phase I information. Conditional on Phase I, it is an ordinary fixed-n test at level α. The same allocations are also certified with the anytime-valid E-value $\bar{W}_t(\frac{1}{2})$, which remains valid when n_i depends on the input's own draws (the pilot policies). We compare with Meta-RS at the same *average* number of samples per input.

Setup. Eval split: 3,000 images for MNIST and CIFAR-10 and 2,000 for ImageNet, each with 100,100 cached draws ($N_{sel} = 100$ Phase I and 10^5 Phase II). We use $\alpha = 0.001$, budgets $B \in \{1{,}000,\ 10{,}000\}$, and ten meta-learner seeds; the meta-learner is the Global champion. Every policy spends the same total as Cohen-B up to rounding. We summarise a policy by the number of samples uniform Cohen needs to reach the same ACR, relative to the policy's own (from Cohen's ACR-versus-N curve; $> 1\times$ means the policy is more sample-efficient), and by ΔACR against Cohen-B with a paired bootstrap 95% confidence interval over images.

Results. Table 20 shows the main comparison.

- **Reallocation with Cohen's certificate beats uniform sampling in every configuration and budget**, for all ten seeds. To match the meta-learned allocation's ACR, uniform Cohen needs 1.30–2.43× the samples on ImageNet, 1.36–1.79× on CIFAR-10 and 1.01–1.15× on MNIST (ΔACR up to $+0.055$ at ImageNet $\sigma = 1.0$, $B = 1{,}000$). This is 21–65% of the hindsight headroom: 53–65% on ImageNet, 42–60% on CIFAR-10 and 21–45% on MNIST. MNIST leaves little to reallocate, because 93% of its inputs at $\sigma = 0.25$ have $p_A > 0.999$.
- **The learned prior adds to the Phase I votes.** The meta-learned allocation matches or exceeds the allocation from the Phase I votes alone in 16 of 18 settings; in the other two the votes lead by at most 0.0001 ACR. The gap is largest at larger σ and budget, on ImageNet: 2.43× against 2.21× at $\sigma = 1.0$, $B = 10{,}000$.
- **The certificate matters as much as the allocation.** Certified with the E-value instead of Clopper-Pearson, the same allocation keeps an advantage only on ImageNet (1.14–1.63×) and in five of six CIFAR-10 settings (1.02–1.13×); it loses on MNIST. Meta-RS, which also stops adaptively but pays the anytime premium on every input, is below uniform Cohen at the same average budget everywhere (0.25–0.93×). The premium is $\sqrt{2\ln(1/\alpha)} = 3.72$ standard errors for a prior concentrated at the true p, against Clopper-Pearson's 3.09.
- **Radius is bought with a little certified accuracy.** Maximising expected radius abstains on some borderline inputs, so certified accuracy falls by 0–2.0 percentage points (largest on CIFAR-10 at $B = 1{,}000$).

Where the samples go. Table 21 breaks the budget down by p_A. Every good allocation (hindsight, the true-p oracle, Phase I votes and the meta-learner) spends at most 2% of the budget on $p_A \le \frac{1}{2}$, less than uniform on 0.5–0.9, and more than uniform on $p_A > 0.99$ (up to 5× as many draws per input). On ImageNet at $\sigma = 0.5$ the meta-learned allocation gives 65% of the budget to the 31% of inputs with $p_A > 0.999$, about 21,000 draws each; hindsight gives them 83%. The tilt sweep in Table 22 tests the opposite strategy directly: favouring inputs near $\frac{1}{2}$ ($\gamma = -4$) is the worst policy in every configuration (0.28–0.74×), and the best tilt is always $\gamma \ge 1$. Extra samples raise the ACR most where p_A is close to 1, not where the decision is close.

Implications. At a fixed total budget, the gain in certified radius comes from *deciding where to spend samples*, using information available after Phase I, combined with Cohen's fixed-n certificate. It does not come from sequential testing, whose anytime-valid premium outweighs the savings in this metric. The two mechanisms are complementary. Meta-learned allocation needs the total budget in advance and returns its certificates as a batch. Anytime-valid stopping makes online decisions per input (early acceptance and rejection, triage, optional continuation) and certifies near-Cohen-10k accuracy with a fraction of the samples. In both, the meta-learner's predicted distribution of p_A is what decides the spending.

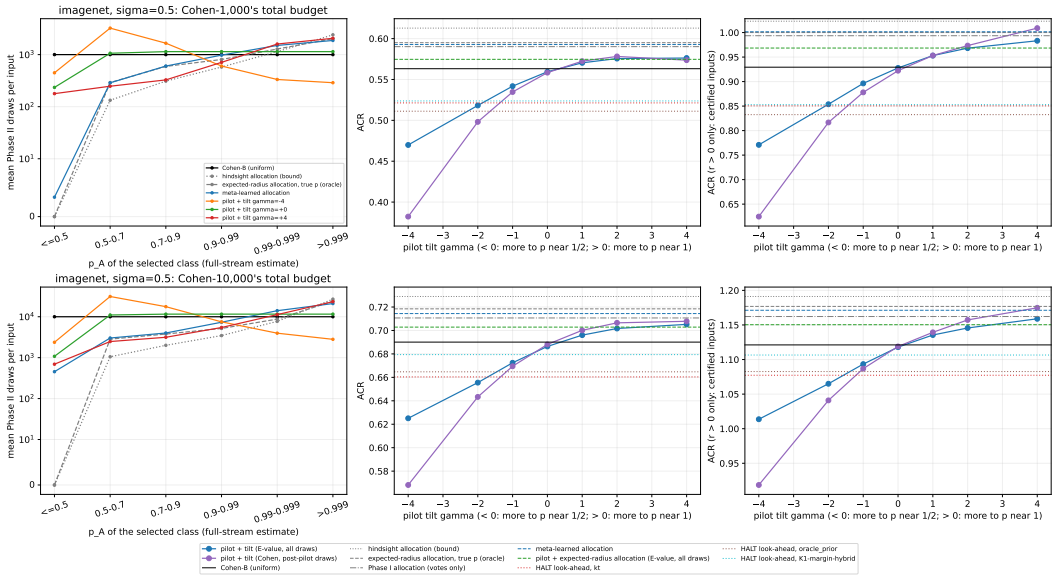


Figure 14: **Budget reallocation on ImageNet, $\sigma = 0.5$.** Left: mean Phase II draws per input by p_A for each allocation policy (rows: $B = 1{,}000$ and $10{,}000$). Middle and right: ACR, and ACR over certified inputs only, against the pilot tilt γ, with the other policies as horizontal lines.

Table 10: **Certification on the eval split** (3,000 images for MNIST and CIFAR-10, 2,000 for ImageNet; $\alpha = 0.001$, $N_{sel} = 100$, checks every 100 draws, horizon 10,000 Phase II draws). Champions are selected on the meta_val split (Appendix D); KT is shown under each champion's stopping rule (ϵ in the Rule column). Meta rows: median over 10 seeds of the meta prior's own results ($^{\dagger n}$: n seeds would fall back to KT on meta_val). Acc: certified accuracy at $r = 0$ (%); ACR: average certified radius (0 for abstentions and wrong classes); Avg All / Cert / Rej: mean samples over all, certified and abstained inputs, including Phase I.

Method	Rule	Acc	ACR	Avg All	Avg Cert	Avg Rej
MNIST, $\sigma = 0.25$						
Cohen-10k	fixed N	99.6	0.775	10,100	10,100	10,100
Cohen-1k	fixed N	99.6	0.602	1,100	1,100	1,100
KT prior	$\epsilon = 0.0126\sigma$	99.6	0.733	9,460	9,465	2,000
Global: Meta-10-Margin-Full	$\epsilon = 0.0126\sigma$	99.6	0.767	9,380	9,385	1,200
KT prior	$\epsilon = 0.0252\sigma$	99.6	0.729	8,870	8,874	2,000
Accuracy: Meta-1-Entropy-Hybrid	$\epsilon = 0.0252\sigma$	99.6	0.763	8,781	8,786	1,200
KT prior	$\epsilon = 0.1008\sigma$	99.6	0.674	4,180	4,181	2,000
Efficiency: Meta-10-Entropy-Hybrid	$\epsilon = 0.1008\sigma$	99.6	0.689	3,152	3,152	1,650
MNIST, $\sigma = 0.5$						
Cohen-10k	fixed N	99.4	1.390	10,100	10,100	10,100
Cohen-1k	fixed N	99.3	1.126	1,100	1,100	1,100
KT prior	$\epsilon = 0.0126\sigma$	99.4	1.321	8,524	8,536	1,280
Global: Meta-10-Margin-Full	$\epsilon = 0.0126\sigma$	99.4	1.361	8,058	8,070	1,080
KT prior	$\epsilon = 0.0252\sigma$	99.4	1.312	7,599	7,610	1,280
Accuracy: Meta-3-Entropy-Hybrid	$\epsilon = 0.0252\sigma$	99.4	1.351	7,052	7,061	1,220
KT prior	$\epsilon = 0.1008\sigma$	99.4	1.216	3,351	3,354	1,280
Efficiency: Meta-3-Entropy-Hybrid	$\epsilon = 0.1008\sigma$	99.4	1.239	2,630	2,633	1,220
MNIST, $\sigma = 1$						
Cohen-10k	fixed N	92.0	1.198	10,100	10,100	10,100
Cohen-1k	fixed N	90.8	1.070	1,100	1,100	1,100
KT prior	$\epsilon = 0.0126\sigma$	91.6	1.149	5,657	5,979	1,414
Global: Meta-10-Margin-Full	$\epsilon = 0.0126\sigma$	91.7	1.159	5,064	5,354	1,186
Accuracy: Meta-10-Entropy-Dynamic	$\epsilon = 0.0126\sigma$	91.7	1.159	5,061	5,351	1,186
KT prior	$\epsilon = 0.0504\sigma$	91.6	1.074	2,031	2,078	1,414
Efficiency: Meta-3-Margin-Full	$\epsilon = 0.0504\sigma$	91.7	1.082	1,711	1,751	1,185
CIFAR-10, $\sigma = 0.25$						
Cohen-10k	fixed N	77.9	0.396	10,100	10,100	10,100
Cohen-1k	fixed N	76.7	0.330	1,100	1,100	1,100
KT prior	$\epsilon = 0.0126\sigma$	77.5	0.378	6,627	7,040	1,979
Global: Meta-10-Margin-Full	$\epsilon = 0.0126\sigma$	77.5	0.386	6,177	6,562	1,831
KT prior	$\epsilon = 0.0252\sigma$	77.5	0.373	5,237	5,526	1,979
Accuracy: Meta-10-Margin-Dynamic	$\epsilon = 0.0252\sigma$	77.5	0.382	4,785	5,046	1,835
KT prior	$\epsilon = 0.1008\sigma$	77.5	0.338	2,046	2,052	1,979
Efficiency: Meta-3-Entropy-Hybrid	$\epsilon = 0.1008\sigma$	77.5	0.343	1,661	1,646	1,833
CIFAR-10, $\sigma = 0.5$						
Cohen-10k	fixed N	66.2	0.507	10,100	10,100	10,100
Cohen-1k	fixed N	64.1	0.435	1,100	1,100	1,100
KT prior	$\epsilon = 0.0126\sigma$	65.7	0.484	5,485	6,237	1,490
Global: Meta-10-Margin-Full	$\epsilon = 0.0126\sigma$	65.8	0.490	5,033	5,722	1,331
KT prior	$\epsilon = 0.0252\sigma$	65.7	0.476	3,951	4,414	1,490
Accuracy: Meta-1-Margin-Hybrid	$\epsilon = 0.0252\sigma$	65.6	0.482	3,532	3,939	1,367
KT prior	$\epsilon = 0.0504\sigma$	65.7	0.458	2,515	2,708	1,490
Efficiency: Meta-1-Entropy-Hybrid	$\epsilon = 0.0504\sigma$	65.6	0.462	2,124	2,265	1,369
CIFAR-10, $\sigma = 1$						
Cohen-10k	fixed N	47.8	0.511	10,100	10,100	10,100
Cohen-1k	fixed N	45.7	0.448	1,100	1,100	1,100
KT prior	$\epsilon = 0.0126\sigma$	47.1	0.486	4,330	5,669	1,312
Global: Meta-10-Margin-Full	$\epsilon = 0.0126\sigma$	47.2	0.491	3,922	5,133	1,170
Accuracy: Meta-3-Margin-Full	$\epsilon = 0.0126\sigma$	47.2	0.491	3,920	5,131	1,170
KT prior	$\epsilon = 0.0504\sigma$	47.1	0.449	1,785	1,994	1,312
Efficiency: Meta-10-Margin-Dynamic	$\epsilon = 0.0504\sigma$	47.2	0.452	1,549	1,716	1,171

Table 10: **Certification on the eval split** (*continued: ImageNet*; columns and protocol as above).

Method	Rule	Acc	ACR	Avg All	Avg Cert	Avg Rej
ImageNet, $\sigma = 0.25$						
Cohen-10k	fixed N	69.4	0.448	10,100	10,100	10,100
Cohen-1k	fixed N	69.0	0.358	1,100	1,100	1,100
KT prior	$\epsilon = 0.0126\sigma$	69.3	0.425	7,354	7,905	1,180
Global: Meta-10-Margin-Full	$\epsilon = 0.0126\sigma$	69.3	0.440	7,055	7,587	1,103
KT prior	$\epsilon = 0.0252\sigma$	69.3	0.422	6,216	6,666	1,180
Accuracy: Meta-1-Margin-Dynamic$^{\dagger 1}$	$\epsilon = 0.0252\sigma$	69.3	0.436	5,921	6,352	1,096
KT prior	$\epsilon = 0.1008\sigma$	69.3	0.387	2,644	2,774	1,180
Efficiency: Meta-3-Margin-Full	$\epsilon = 0.1008\sigma$	69.3	0.394	2,094	2,183	1,105
ImageNet, $\sigma = 0.5$						
Cohen-10k	fixed N	61.6	0.690	10,100	10,100	10,100
Cohen-1k	fixed N	60.6	0.563	1,100	1,100	1,100
KT prior	$\epsilon = 0.0126\sigma$	61.3	0.656	6,297	7,285	1,282
Global: Meta-10-Margin-Full	$\epsilon = 0.0126\sigma$	61.4	0.675	5,935	6,866	1,186
KT prior	$\epsilon = 0.0252\sigma$	61.3	0.649	5,092	5,842	1,282
Accuracy: Meta-6-Margin-Dynamic	$\epsilon = 0.0252\sigma$	61.4	0.668	4,765	5,465	1,184
KT prior	$\epsilon = 0.1008\sigma$	61.3	0.593	2,120	2,285	1,282
Efficiency: Meta-6-Entropy-Dynamic	$\epsilon = 0.1008\sigma$	61.3	0.603	1,730	1,836	1,199
ImageNet, $\sigma = 1$						
Cohen-10k	fixed N	46.8	0.864	10,100	10,100	10,100
Cohen-1k	fixed N	45.6	0.720	1,100	1,100	1,100
KT prior	$\epsilon = 0.0126\sigma$	46.7	0.823	4,909	6,583	995
Global: Meta-10-Margin-Full	$\epsilon = 0.0126\sigma$	46.7	0.840	4,570	6,123	934
KT prior	$\epsilon = 0.0252\sigma$	46.7	0.812	3,706	4,865	995
Accuracy: Meta-1-Margin-Dynamic	$\epsilon = 0.0252\sigma$	46.6	0.829	3,405	4,456	943
KT prior	$\epsilon = 0.1008\sigma$	46.7	0.730	1,539	1,771	995
Efficiency: Meta-1-Entropy-Dynamic	$\epsilon = 0.1008\sigma$	46.6	0.740	1,294	1,446	943

Table 11: **Complete Frequentist Sequential Baselines vs. E-Values across Noise Scales** (ImageNet, eval: 2,000 images per $\sigma \in \{0.25, 0.50, 1.00\}$). Every sequential method uses the Global champion's stopping rule ($\epsilon = 0.0126\sigma$, look-ahead) and the same rejection exits; the multi-stage tests look at 10 equally spaced checkpoints including the last; the LIL boundary checks every 100 draws; α-spending uses Lan–DeMets O'Brien–Fleming (OBF) and Pocock functions. Avg Cert / All: mean samples over certified and all inputs (incl. Phase I); Looks: all = every 100 draws; ACR: average certified radius; Speedup: Cohen-10k's 10,100 samples over Avg All; FR: inputs Cohen-10k certifies but the method does not (%). Meta-RS: median over 10 seeds.

σ	Method	Looks	Acc (%)	Avg Cert	Avg All	ACR	Speedup	FR (%)
0.25	Cohen-10k	1	69.4	10,100	10,100	0.448	1.00×	0.0
	Cohen-1k	1	69.0	1,100	1,100	0.358	9.18×	0.6
	Bonferroni (uniform)	10	69.4	10,100	9,610	0.438	1.05×	0.0
	Bonferroni (geometric)	10	69.4	10,100	9,615	0.424	1.05×	0.0
	α-spending (OBF)	10	69.4	10,100	9,686	0.444	1.04×	0.0
	α-spending (Pocock)	10	69.4	10,100	9,610	0.437	1.05×	0.0
	LIL boundary	all	69.4	10,100	9,616	0.299	1.05×	0.0
	KT prior (E-value)	all	69.3	7,905	7,354	0.425	1.37×	0.1
	Meta-RS (Global)	all	69.3	7,587	7,055	0.440	1.43×	0.1
0.5	Cohen-10k	1	61.6	10,100	10,100	0.690	1.00×	0.0
	Cohen-1k	1	60.6	1,100	1,100	0.563	9.18×	1.5
	Bonferroni (uniform)	10	61.5	10,100	9,018	0.677	1.12×	0.2
	Bonferroni (geometric)	10	61.4	10,092	9,014	0.657	1.12×	0.2
	α-spending (OBF)	10	61.5	10,100	9,170	0.686	1.10×	0.2
	α-spending (Pocock)	10	61.5	10,100	9,017	0.675	1.12×	0.2
	LIL boundary	all	61.4	10,100	9,010	0.481	1.12×	0.2
	KT prior (E-value)	all	61.3	7,285	6,297	0.656	1.60×	0.4
	Meta-RS (Global)	all	61.4	6,866	5,935	0.675	1.70×	0.3
1	Cohen-10k	1	46.8	10,100	10,100	0.864	1.00×	0.0
	Cohen-1k	1	45.6	1,100	1,100	0.720	9.18×	2.6
	Bonferroni (uniform)	10	46.7	10,100	7,856	0.850	1.29×	0.1
	Bonferroni (geometric)	10	46.8	10,100	7,860	0.828	1.28×	0.0
	α-spending (OBF)	10	46.8	10,100	8,154	0.859	1.24×	0.0
	α-spending (Pocock)	10	46.7	10,100	7,854	0.848	1.29×	0.1
	LIL boundary	all	46.7	10,100	7,797	0.642	1.30×	0.1
	KT prior (E-value)	all	46.7	6,582	4,909	0.823	2.06×	0.2
	Meta-RS (Global)	all	46.7	6,123	4,570	0.840	2.21×	0.2

Table 12: **Early rejection only**: precision and plateau exits disabled, so an input is certified only at the 10,000-draw horizon unless a rejection exit (UCB, bankruptcy, reference bankruptcy) fires first; run on every eval input, with no Cohen oracle. Meta: the Efficiency champion of each configuration (Table 9), median over 10 seeds; KT under the same rule. Cert: certified accuracy (%); FR: inputs Cohen-10k certifies but the method rejects (%); δr: mean radius gap to Cohen-10k on inputs both certify (%); Exit: mean samples on the inputs the method rejects (incl. Phase I).

		Cohen-10k	Cert (%)		FR (%)		δr (%)		Exit (rejected)	
Dataset	σ	Cert (%)	Meta	KT	Meta	KT	Meta	KT	Meta	KT
MNIST	0.25	99.6	99.6	99.6	0.0	0.0	0.61	4.95	1,650	2,000
	0.5	99.4	99.4	99.4	0.0	0.0	1.39	4.17	1,220	1,280
	1	92.0	91.7	91.6	0.4	0.5	1.91	3.31	1,185	1,414
CIFAR-10	0.25	77.9	77.5	77.5	0.4	0.4	2.12	4.22	1,833	1,979
	0.5	66.2	65.6	65.7	0.9	0.8	3.02	4.65	1,369	1,490
	1	47.8	47.2	47.1	1.4	1.5	3.72	5.33	1,171	1,312
ImageNet	0.25	69.4	69.3	69.3	0.1	0.1	1.50	4.46	1,105	1,180
	0.5	61.6	61.3	61.3	0.4	0.4	2.08	4.47	1,199	1,282
	1	46.8	46.6	46.7	0.3	0.2	2.54	4.57	944	995

Table 13: **Early rejection only**: precision and plateau exits disabled, so an input is certified only at the 10,000-draw horizon unless a rejection exit (UCB, bankruptcy, reference bankruptcy) fires first; run on every eval input, with no Cohen oracle. Meta: the Accuracy champion of each configuration (Table 9), median over 10 seeds; KT under the same rule. Cert: certified accuracy (%); FR: inputs Cohen-10k certifies but the method rejects (%); δr: mean radius gap to Cohen-10k on inputs both certify (%); Exit: mean samples on the inputs the method rejects (incl. Phase I).

		Cohen-10k	Cert (%)		FR (%)		δr (%)		Exit (rejected)	
Dataset	σ	Cert (%)	Meta	KT	Meta	KT	Meta	KT	Meta	KT
MNIST	0.25	99.6	99.6	99.6	0.0	0.0	0.62	4.95	1,200	2,000
	0.5	99.4	99.4	99.4	0.0	0.0	1.39	4.17	1,220	1,280
	1	92.0	91.7	91.6	0.4	0.5	2.01	3.31	1,186	1,414
CIFAR-10	0.25	77.9	77.5	77.5	0.4	0.4	2.17	4.22	1,835	1,979
	0.5	66.2	65.6	65.7	0.9	0.8	3.02	4.65	1,367	1,490
	1	47.8	47.2	47.1	1.4	1.5	3.71	5.33	1,170	1,312
ImageNet	0.25	69.4	69.3	69.3	0.1	0.1	1.46	4.46	1,096	1,180
	0.5	61.6	61.4	61.3	0.3	0.4	2.10	4.47	1,184	1,282
	1	46.8	46.6	46.7	0.3	0.2	2.56	4.57	943	995

Table 14: **Zero-shot transfer** (target ImageNet, eval: 2,000 images per σ). The prior is Meta-3-Margin-Dynamic without $\phi(x)$ (inputs: the margin of the Phase I mean softmax and $\hat{p}_{sel}$, so its dimension does not depend on the dataset), trained on the source dataset at the same σ and run under the Global rule ($\epsilon = 0.0126\sigma$); median over 10 seeds. FR: inputs Cohen-10k certifies but the method does not (%); Agreement: same certify/abstain decision as the native prior of the same seed (%); Speedup: Cohen-10k's samples over Avg All.

σ	Prior	Acc (%)	ACR	FR (%)	Agreement (%)	Avg Cert	Avg All	Speedup
0.25	Cohen-10k	69.4	0.448	–	–	10,100	10,100	1.00×
	Native (ImageNet)	69.3	0.440	0.1	100.0	7,540	7,011	1.44×
	Zero-shot from CIFAR-10	69.3	0.436	0.1	100.0	7,721	7,177	1.41×
	Zero-shot from MNIST	69.3	0.437	0.1	100.0	8,036	7,459	1.35×
0.5	Cohen-10k	61.6	0.690	–	–	10,100	10,100	1.00×
	Native (ImageNet)	61.4	0.676	0.3	100.0	6,826	5,899	1.71×
	Zero-shot from CIFAR-10	61.4	0.671	0.3	100.0	6,950	6,003	1.68×
	Zero-shot from MNIST	61.4	0.656	0.3	100.0	7,296	6,297	1.60×
1	Cohen-10k	46.8	0.864	–	–	10,100	10,100	1.00×
	Native (ImageNet)	46.7	0.842	0.1	100.0	6,068	4,527	2.23×
	Zero-shot from CIFAR-10	46.7	0.837	0.2	100.0	6,116	4,564	2.21×
	Zero-shot from MNIST	46.7	0.816	0.2	100.0	6,663	4,947	2.04×

Table 15: **Influence of the meta-learner's design choices** (eval). For every configuration, architecture and ϵ: the median over 10 seeds, relative to KT under the same stopping rule; then averaged over everything else. N/N_{KT}: samples relative to KT; ACR and Acc relative to Cohen-10k; last column: seeds whose samples exceed KT's (%).

Parameter	Value	N/N_{KT}	ACR / $\text{ACR}_{\text{Cohen-10k}}$	ΔAcc (pp)	Worse than KT (%)
Components K	1	0.840	0.821	-0.26	3.2
	3	0.841	0.821	-0.25	0.0
	6	0.841	0.821	-0.25	0.0
	10	0.841	0.821	-0.25	0.0
Feature	Entropy	0.841	0.821	-0.26	0.9
	Margin	0.840	0.821	-0.25	0.7
Support	Dynamic	0.841	0.821	-0.25	2.4
	Full	0.840	0.821	-0.25	0.0
	Hybrid	0.840	0.821	-0.26	0.0
ϵ/σ	0.0126	0.936	0.975	-0.25	0.8
	0.0252	0.921	0.962	-0.25	0.8
	0.0504	0.764	0.916	-0.25	0.8
	0.1008	0.821	0.856	-0.25	0.8
	0.2016	0.766	0.718	-0.25	0.8
	0.4032	0.835	0.502	-0.25	0.8

Table 16: **Meta-learner inputs** (ImageNet, $\sigma = 0.5$, eval: 2,000 images; Meta-10-Margin-Full under the Global rule, $\epsilon = 0.0126\sigma$; median over 10 seeds). Softmax proxy: the margin of the Phase I mean softmax (Full, clean-image softmax: of the clean image's, as in the original method); ϕ of the Phase I draws: the embedding averaged over the 100 Phase I noisy draws instead of the clean image's. Avg All / Cert / Rej: mean samples over all, certified and abstained inputs (incl. Phase I); FR (%): inputs Cohen-10k certifies but the method does not.

Inputs	$\phi(x)$	Acc (%)	ACR	Avg All	Avg Cert	Avg Rej	FR (%)
Static KT prior ($K = 0$)	–	61.3	0.656	6,297	7,285	1,282	0.4
Full: $\phi(x)$, softmax proxy, $\hat{p}_{sel}$	✓	61.4	0.675	5,935	6,866	1,186	0.3
Full, ϕ of the Phase I draws	✓	61.4	0.675	5,943	6,876	1,185	0.3
Full, clean-image softmax	✓	61.4	0.673	5,950	6,884	1,194	0.3
Softmax proxy, $\hat{p}_{sel}$	×	61.4	0.676	5,897	6,823	1,176	0.3
Softmax proxy only	×	61.3	0.675	6,012	6,952	1,240	0.4

Table 17: **Meta-learner depth and safety anchor** (eval). Variants of the Global champion (Meta-10-Margin-Full, ch.), each trained for all 10 seeds in every configuration and run under the Global rule ($\epsilon = 0.0126\sigma$). N/N_{ch}: average samples relative to the champion of the same seed, median over seeds, range over the three σ; ACR likewise, over all nine configurations; last column: seeds that need at least as many samples as KT, of 90. The anchor weight is used in training and certification. Hidden layers: [Linear 512, LayerNorm, ReLU] blocks after the input projection.

Setting	Value	N/N_{ch} MNIST	N/N_{ch} CIFAR-10	N/N_{ch} ImageNet	ACR / ACR_{ch}	$\geq$ KT
Hidden layers	0	1.000–1.005	1.001–1.002	0.994–1.002	0.999–1.001	0/90
	1	1.000	1.000–1.001	0.997–1.002	1.000	0/90
	2 (champion)	1.000	1.000	1.000	1.000	0/90
	4	1.000	1.000	0.999–1.000	1.000	0/90
Anchor w_{safety}	0	0.999–1.000	0.999–1.000	0.998–1.001	1.000	0/90
	0.001	1.000	0.999–1.000	0.998–1.000	1.000	0/90
	0.01 (champion)	1.000	1.000	1.000	1.000	0/90
	0.1	1.000–1.004	1.003–1.005	1.003	0.999–1.000	0/90

Table 18: **Meta-training data and its cost** (eval, Global rule). Meta-training needs a Phase I glimpse (100 draws) and the Phase II stream (10,000 draws, truncated in the "draws" rows) for each meta-train and meta-val image, all by default (5,000 + 1,000 for MNIST and CIFAR-10, 10,000 + 2,000 for ImageNet); the image rows subsample them in the same proportion. Pred.: noisy predictions (millions). % clf.: their GPU time relative to training the base classifier (0.04, 0.10 and 116.04 h of compute for MNIST, CIFAR-10 and ImageNet: 30, 100 and 90 epochs of 60,000, 50,000 and 1,281,167 images), both measured on an RTX 2080 Ti in fp32 with synthetic inputs. N/N_{ch}: average samples relative to the champion of the same seed, median over 10 seeds, range over the three σ. Training the meta-learner itself takes at most 190 s on one CPU core.

Training data	MNIST			CIFAR-10			ImageNet		
	Pred.	% clf.	N/N_{ch}	Pred.	% clf.	N/N_{ch}	Pred.	% clf.	N/N_{ch}
All (champion)	60.6	414.4	1.000	60.6	164.3	1.000	121.2	32.6	1.000
2k images	24.2	165.8	0.999–1.006	24.2	65.7	1.004–1.005	24.2	6.5	1.012–1.015
1k images	12.1	82.9	1.001–1.014	12.1	32.9	1.009–1.010	12.1	3.3	1.028–1.031
500 images	6.1	41.4	1.001–1.023	6.1	16.4	1.013–1.019	6.1	1.6	1.029–1.055
3k draws	18.6	127.2	0.997–1.000	18.6	50.4	0.999–1.000	37.2	10.0	0.999–1.001
1k draws	6.6	45.1	0.995–1.000	6.6	17.9	0.998–1.000	13.2	3.5	0.999–1.000
1k images, 1k draws	1.3	9.0	0.998–1.014	1.3	3.6	1.007–1.011	1.3	0.3	1.027–1.032

Table 19: **The safety anchor under a misspecified prior** (target ImageNet, eval: 2,000 images per σ, Global rule). The zero-shot prior (Meta-3-Margin-Dynamic without $\phi(x)$, Table 14) trained on the source dataset with the anchor ($w_{safety} = 0.01$, in training and certification) and without it (0). ACR and Avg All: median over 10 seeds and the worst seed (lowest ACR, most samples).

σ	Prior	w_{safety}	Acc (%)	ACR	ACR, worst seed	Avg All	Avg All, worst seed
	KT	–	69.3	0.425	–	7,354	–
	Native (ImageNet)	0.01	69.3	0.440	0.440	7,011	7,014
		0	69.3	0.441	0.440	7,010	7,031
0.25	From CIFAR-10	0.01	69.3	0.436	0.436	7,177	7,186
		0	69.3	0.436	0.436	7,179	7,209
	From MNIST	0.01	69.3	0.437	0.425	7,459	7,596
		0	69.3	0.418	0.000	7,443	9,348
	KT	–	61.3	0.656	–	6,297	–
	Native (ImageNet)	0.01	61.4	0.676	0.675	5,899	5,905
		0	61.4	0.676	0.675	5,896	5,915
0.5	From CIFAR-10	0.01	61.4	0.671	0.671	6,003	6,008
		0	61.4	0.671	0.670	6,002	6,006
	From MNIST	0.01	61.4	0.656	0.651	6,297	6,365
		0	61.4	0.642	0.607	6,413	6,452
	KT	–	46.7	0.823	–	4,909	–
	Native (ImageNet)	0.01	46.7	0.842	0.842	4,527	4,534
		0	46.7	0.842	0.842	4,527	4,539
1	From CIFAR-10	0.01	46.7	0.837	0.836	4,564	4,572
		0	46.7	0.837	0.836	4,564	4,572
	From MNIST	0.01	46.7	0.816	0.814	4,947	4,956
		0	46.7	0.801	0.766	5,029	5,098

Table 20: **The same total budget as Cohen-B, spent differently across inputs** (eval split). Each policy spends on average the same B Phase II draws per input as Cohen-B (plus the $N_{sel} = 100$ Phase I draws), choosing per input how many (up to 10^5). All columns but the two ACR columns: the samples uniform Cohen needs to reach the policy's ACR, as a multiple of the policy's own (> 1: the policy is more sample-efficient). Hindsight is an upper bound that knows every input's realised radius. ΔACR: meta-learned allocation minus Cohen-B, median over 10 seeds; every seed's paired-bootstrap 95% CI lies above 0 in every row.

		Cohen-B	Hindsight	Allocation + Cohen certificate			Meta alloc.	Meta-RS
σ	B	ACR	(bound)	Phase I	Meta	ΔACR	+ E-value	(sequential)
MNIST								
0.25	1,000	0.602	1.03	1.01	**1.01**	+0.0009	0.98	0.87
0.25	10,000	0.775	1.05	1.02	**1.02**	+0.0014	0.95	0.93
0.5	1,000	1.126	1.11	1.03	**1.04**	+0.0060	0.93	0.80
0.5	10,000	1.390	1.30	1.07	**1.12**	+0.0094	0.88	0.80
1	1,000	1.070	1.67	1.13	**1.15**	+0.0129	0.76	0.61
1	10,000	1.198	1.91	1.12	**1.14**	+0.0036	0.63	0.53
CIFAR-10								
0.25	1,000	0.330	1.75	1.34	**1.36**	+0.0121	1.08	0.67
0.25	10,000	0.396	2.29	1.49	**1.62**	+0.0098	1.13	0.74
0.5	1,000	0.435	2.37	1.46	**1.51**	+0.0185	1.05	0.64
0.5	10,000	0.507	3.29	1.64	**1.80**	+0.0108	1.04	0.62
1	1,000	0.448	3.41	1.62	**1.62**	+0.0199	1.02	0.54
1	10,000	0.511	3.98	1.62	**1.64**	+0.0066	0.85	0.51
ImageNet								
0.25	1,000	0.358	1.66	1.28	**1.30**	+0.0127	1.14	0.82
0.25	10,000	0.448	1.91	1.38	**1.46**	+0.0120	1.22	0.85
0.5	1,000	0.563	2.09	1.46	**1.52**	+0.0294	1.25	0.76
0.5	10,000	0.690	2.62	1.62	**1.78**	+0.0243	1.35	0.79
1	1,000	0.720	3.12	1.90	**1.98**	+0.0550	1.46	0.78
1	10,000	0.864	4.40	2.21	**2.43**	+0.0372	1.60	0.71

Table 21: **Where the budget goes** (eval, $B = 10{,}000$). Share of all Phase II draws, and in parentheses the mean draws per input, by the input's p_A (its 100,000-draw estimate). Meta-learned: median over 10 seeds.

p_A	≤ 0.5	0.5–0.7	0.7–0.9	0.9–0.99	0.99–0.999	> 0.999
CIFAR-10, $\sigma = 0.5$						
inputs	405 (14%)	668 (22%)	700 (23%)	658 (22%)	297 (10%)	272 (9%)
Uniform (Cohen-B)	14% (10,000)	22% (10,000)	23% (10,000)	22% (10,000)	10% (10,000)	9% (10,000)
Hindsight (bound)	0% (0)	7% (3,313)	11% (4,898)	20% (8,975)	16% (16,440)	45% (49,884)
True p (oracle)	0% (0)	11% (5,154)	16% (6,879)	21% (9,416)	16% (15,809)	36% (39,886)
Phase I votes	2% (1,120)	13% (5,957)	18% (7,600)	25% (11,589)	20% (20,323)	22% (24,208)
Meta-learned	2% (1,162)	12% (5,234)	16% (6,926)	22% (10,075)	19% (18,702)	30% (33,073)
Pilot, $\gamma = -4$	1% (825)	48% (21,470)	32% (13,561)	14% (6,185)	3% (3,502)	2% (2,651)
Pilot, $\gamma = +4$	1% (476)	7% (2,937)	10% (4,491)	20% (9,184)	21% (21,219)	41% (45,428)
ImageNet, $\sigma = 0.5$						
inputs	287 (14%)	279 (14%)	330 (16%)	310 (16%)	178 (9%)	616 (31%)
Uniform (Cohen-B)	14% (10,000)	14% (10,000)	16% (10,000)	16% (10,000)	9% (10,000)	31% (10,000)
Hindsight (bound)	0% (0)	1% (1,052)	3% (2,005)	5% (3,465)	7% (7,703)	83% (26,930)
True p (oracle)	0% (0)	4% (2,839)	6% (3,780)	8% (5,179)	8% (8,747)	74% (24,019)
Phase I votes	1% (534)	6% (4,626)	10% (6,106)	14% (9,269)	14% (15,397)	55% (17,737)
Meta-learned	1% (423)	4% (2,851)	7% (3,956)	11% (7,255)	12% (14,039)	65% (21,188)
Pilot, $\gamma = -4$	1% (1,004)	42% (29,840)	32% (19,270)	13% (8,076)	4% (4,171)	9% (2,892)
Pilot, $\gamma = +4$	1% (394)	3% (2,226)	5% (3,159)	9% (5,492)	10% (11,362)	72% (23,537)

Table 22: **Where to spend extra samples: pilot + tilt** ($B = 10,000$, eval). A pilot of $0.2B$ draws with futility exits, then the rest shared $\propto e^{\gamma u}$, $u = \Phi^{-1}(\hat{p})$ scaled to $[0, 1]$: $\gamma < 0$ favours inputs near $p = \frac{1}{2}$, $\gamma > 0$ near $p = 1$. Certified with the E-value over all draws. Cells as in Table 20: the samples uniform Cohen needs for the same ACR, as a multiple of the policy's (> 1: better than uniform Cohen). Last column: the meta-learned allocation for comparison.

		γ							Meta
Dataset	σ	-4	-2	-1	$+0$	$+1$	$+2$	$+4$	alloc.
MNIST	0.25	0.74	0.88	0.91	0.94	0.95	0.95	**0.96**	0.95
MNIST	0.5	0.48	0.68	0.75	0.81	0.85	**0.87**	**0.87**	0.88
MNIST	1	0.40	0.50	0.56	0.61	**0.63**	**0.63**	0.59	0.63
CIFAR-10	0.25	0.29	0.47	0.63	0.82	0.98	1.10	**1.16**	1.13
CIFAR-10	0.5	0.29	0.45	0.58	0.75	0.92	1.03	**1.04**	1.04
CIFAR-10	1	0.39	0.53	0.63	0.72	0.83	**0.88**	0.81	0.85
ImageNet	0.25	0.31	0.55	0.74	0.93	1.08	1.18	**1.25**	1.22
ImageNet	0.5	0.29	0.50	0.70	0.94	1.17	1.32	**1.43**	1.35
ImageNet	1	0.29	0.50	0.71	1.02	1.34	1.60	**1.74**	1.60

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 key claims made in the paper are: a novel meta learner (described in Section 4) significant reductions in computational cost (described in Section 6), the ability to dynamically allocate compute and perform resource triage (both also described in Section 6). We are confident that the abstract and claims within the paper are well aligned.

   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: We discuss in Section 6 that anytime-valid certification bounds will always, inherently, be looser than those associated with a fixed frequentist N. These insights are contextualised by the utility provided by anytime valid certifications. Besides this, the paper is predicated upon the standard assumptions of this field—in that it measures robustness against ℓ_2 adversarial examples.

   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 statements build upon the foundations of prior work, and when a proof is not included in the document, it is available within the referenced materials.

   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 paper comprehensively documents all required steps to recreate this work, including algorithms, experimental settings, and computational environments employed. Our associated github also provides the information required to replicate this work.

   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: All associated code is hosted on our github, as linked within the paper.

   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: All details are covered in both the main body and appendices.

   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: [No]

   Justification: Experiments were conducted over 1000 validation points, with confidence intervals for the underlying quantities set to $\alpha = 0.001$, corresponding to a 99.9% confidence interval. These levels of tightness left us to remove the confidence intervals, as they were uniformly tiny while simultaneously creating visual noise in the underlying presentations.

   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: These results were discussed within the Appendices.

   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: We have consulted the NeurIPS Ethics Guidelines, and are confident that our work conforms with them. Our Impact Statement discusses the broader ethical positioning of this work.

   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: [Yes]

Justification: As per the prior answer, we have included a detailed Impact Statement that we believe covers the societal impacts associated with this work.

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: We do not believe any of this works associated content aligns with a need for responsible release restrictions.

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: Licenses for the tested datasets are discussed in the Appendices.

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: [N/A]

    Justification: No additional assets are introduced within this work.

    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: No human participants were involved within this work.

    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: No IRB approval was required, due to both the lack of human subjects and the broad alignment of this work with institutional policies.

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: No non-standard uses of LLMs were required for this research.

    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.