Adaptive Coverage Policies in Conformal Prediction

Etienne Gauthier
Inria, Ecole Normale Supérieure,
PSL Research University

Francis Bach
Inria, Ecole Normale Supérieure,
PSL Research University

Michael I. Jordan
Inria, Ecole Normale Supérieure,
PSL Research University,
University of California, Berkeley

Abstract

Traditional conformal prediction methods construct prediction sets such that the true label falls within the set with a user-specified coverage level. However, poorly chosen coverage levels can result in uninformative predictions, either producing overly conservative sets when the coverage level is too high, or empty sets when it is too low. Moreover, the fixed coverage level cannot adapt to the specific characteristics of each individual example, limiting the flexibility and efficiency of these methods. In this work, we leverage recent advances in e-values and post-hoc conformal inference, which allow the use of data-dependent coverage levels while maintaining valid statistical guarantees. We propose to optimize an adaptive coverage policy by training a neural network using a leave-one-out procedure on the calibration set, allowing the coverage level and the resulting prediction set size to vary with the difficulty of each individual example. We support our approach with theoretical coverage guarantees and demonstrate its practical benefits through a series of experiments.

1 INTRODUCTION

Conformal prediction (Gammerman et al., 1998; Vovk et al., 2005; Shafer and Vovk, 2008) is a powerful framework for quantifying uncertainty that is particularly useful in machine-learning applications (Papadopoulos et al., 2002; Balasubramanian et al., 2014; Laxhammar and Falkman, 2015; Lei et al., 2018; Chernozhukov et al., 2018; Angelopoulos et al., 2021; Fisch et al., 2021; Cella and Martin, 2021; Johnstone and Cox, 2021; Bates et al., 2023; Su et al., 2024). It provides prediction sets that contain the true label with high probability, without relying on parametric assumptions about the data distribution. The only requirement is exchangeability, meaning the joint distribution of the data remains invariant under any permutation of the samples. This assumption is significantly weaker than the standard independent and identically distributed (i.i.d.) assumption, making conformal prediction broadly applicable while still offering strong, distribution-free guarantees.

In standard conformal prediction, the available data is typically split into a training set, used to fit a predictive model, and a calibration set, which is held out to assess the model's behavior on unseen data. Applying the model to the calibration set yields nonconformity scores, which quantify how unusual each true label is relative to the model's prediction. These scores are then used to construct prediction sets for a new test point that satisfy the desired coverage guarantee. Formally, letting $\mathcal{X} \times \mathcal{Y}$ denote the feature-label space, we assume that a calibration set $\{(X_i, Y_i)\}_{i=1}^n$ and a new input $(X_{\text{test}}, Y_{\text{test}})$ have been drawn exchangeably from some distribution $\mathbb{P}$ over $\mathcal{X} \times \mathcal{Y}$. Conformal prediction constructs a prediction set, $\hat{C}_n^\alpha(X_{\text{test}}) \subseteq \mathcal{Y}$, based on the calibration set such that

$$\mathbb{P}(Y_{\text{test}} \in \hat{C}_n^\alpha(X_{\text{test}})) \geq 1 - \alpha, \tag{1}$$

where $\alpha \in (0, 1)$ is a user-specified miscoverage level. This guarantee is marginal, holding over all sources of randomness including both the calibration samples and the test point.

Machine learning prediction methods often focus on producing a point prediction for a new input X, via a mapping $f\colon \mathcal{X} \to \mathcal{Y}$, without indicating uncertainty in the prediction. Conformal prediction addresses this limitation by providing a principled, model-agnostic approach to uncertainty quantification: it acts as a wrapper around f, transforming its outputs into a prediction set that contains the true label with high probability. To construct such sets, conformal prediction uses a score function, $S\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, that depends

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

on the model f, and quantifies how well a candidate label matches the model's prediction. The standard approach in conformal prediction relies on the observation that the calibration scores $S(X_i, Y_i)$ for $i = 1, \ldots, n$ and the test score $S(X_{\text{test}}, Y_{\text{test}})$ are exchangeable. As a result, their ranks follow a uniform distribution. In particular, the test score is unlikely to rank among the highest, which is the key insight used to construct valid prediction sets in conformal prediction. For a comprehensive treatment of conformal prediction and its applications in machine learning, we refer the reader to Angelopoulos and Bates (2023) and Angelopoulos et al. (2024).

A core feature of classical conformal prediction is the specification of a fixed coverage level $1 - \alpha$ prior to observing the data. While this provides a highly desirable and easily interpretable statistical guarantee, it imposes a rigid operational constraint. In practice, this fixed choice can lead to uninformative predictions: small values of α can yield overly conservative prediction sets, while large values may lead to empty sets. Moreover, classical conformal methods do not allow α to be chosen based on the observed data, because the marginal coverage guarantee (1) only holds when α is fixed before seeing the calibration or test points. In many operational settings, a practitioner might prefer to adjust α after inspecting the data. For instance, in a medical diagnosis task, a standard conformal method guaranteeing 99% coverage might output a broad set of seven diagnoses. While statistically valid, this set may be clinically unactionable. A physician might prefer to dynamically relax the coverage slightly (e.g., from $\alpha = 1\%$ to $\alpha = 2\%$) to yield a precise, actionable set of three diagnoses. Most practitioners would naturally opt for the latter, as it provides a more informative and usable prediction. Unfortunately, in traditional conformal prediction, choosing α after inspecting the data undermines the marginal coverage guarantee (1) and is closely related to the phenomenon of p-hacking in the statistical literature (Simmons et al., 2011; Head et al., 2015). This rigidity is limiting in settings where uncertainty varies across examples, or where we may want to tailor coverage to the difficulty of each instance.

The method proposed by Cherian et al. (2024) uses a neural network to fit a data-dependent miscoverage $\tilde{\alpha}$ in conformal prediction. However, their network is trained on separate training data and offers no guidance on how to optimize training to achieve a desired expected prediction set size. By contrast, our approach trains $\tilde{\alpha}$ directly on calibration data, without needing to reserve any training data for this purpose, and allows practitioners to adjust the training procedure to control the expected size of prediction sets at test time.

We tackle this challenge by leveraging recent advances in e-values (Shafer and Vovk, 2019; Vovk and Wang, 2021; Grünwald et al., 2024; Ramdas and Wang, 2025) and post-hoc inference (Wang and Ramdas, 2022; Xu et al., 2024; Grünwald, 2024; Koning, 2024; Gauthier et al., 2025b; Chugg et al., 2026), which provide valid coverage guarantees even when the miscoverage level α is selected adaptively. It is known that conformal sets constructed using e-values generally yield slightly larger prediction sets than those built with standard p-values for a given fixed α (Vovk, 2025). However, e-values are necessary to enable this adaptivity as they are the only way to construct post-hoc p-values, the statistical objects that allow for valid inference even when the miscoverage level is chosen after observing the data (Koning, 2024). We willingly trade a slight loss in fixed-α efficiency for this powerful operational advantage, as it allows the coverage level to be selected adaptively without invalidating the statistical guarantees.

Building on this, we propose to optimize an adaptive coverage policy by training a neural network on observed data. To construct training examples, we adopt a leave-one-out approach on the calibration set: for each held-out point, the remaining points serve as a pseudo calibration set, and the held-out point acts as a pseudo test sample. This generates a collection of labeled examples, enabling a model to predict a sample-specific coverage policy. Our aim is to design a mapping from each pseudo calibration-test pair to a coverage policy that maximizes the *informativeness* of the resulting prediction sets. We formalize this informativeness as a dual objective (detailed in Section 2.3): minimizing the prediction set size while simultaneously minimizing the adaptive miscoverage level $\tilde{\alpha}$. This trade-off is governed by a user-specified regularization parameter λ. Furthermore, in Section 2.4, we introduce a principled procedure to select λ such that the expected prediction set size at test time meets a desired target. Crucially, this approach allows the data-dependent miscoverage level $\tilde{\alpha}$ to adapt to individual test samples while maintaining valid marginal guarantees, offering a more flexible and data-driven alternative to choosing a fixed α in conventional conformal prediction.

By leveraging the post-hoc validity of e-values, this approach enables adaptive conformal prediction, modulating prediction set size according to instance difficulty. Our work is related to recent efforts that combine e-values with conformal prediction (Vovk, 2025; Gauthier et al., 2025b). In particular, we contrast our approach with recent prior work (Gauthier et al., 2025a; Liu et al., 2026), which also utilizes e-values and a leave-one-out strategy. However, our objective differs fundamentally: while that method inverts the conformal procedure to find the minimum α satisfying a *hard* size constraint

for a specific prediction set, our method learns a parametric coverage policy to optimize a *soft* constraint by targeting an expected prediction set size at test time. Furthermore, the two methods employ the leave-one-out procedure for entirely different purposes. Whereas the prior work uses it solely to estimate coverage guarantees, we utilize it as an integral generative step to construct the training data required to fit our neural coverage policy.

2 METHOD

Traditional conformal prediction methods compare the rank of the test score to those in the calibration set, a procedure that can be interpreted in terms of p-values. As we will discuss, an alternative is to base conformal inference on e-values. This approach has a wider range of applicability and in particular will permit us to obtain valid inference even when the miscoverage level α is selected in a data-dependent manner.

2.1 Conformal e-prediction

Conformal sets can be constructed using e-values, a method known as conformal e-prediction. E-values are simply the realizations of random variables known as e-variables:

Definition 2.1 (E-variable)**.** An *e-variable* E is a nonnegative random variable that satisfies

$$\mathbb{E}[E] \leq 1.$$

Thresholding an e-variable at level $1/\alpha$ yields a prediction set with marginal coverage at least $1-\alpha$. Indeed, we can apply Markov's inequality to obtain:

$$\mathbb{P}(E < 1/\alpha) \geq 1 - \alpha.$$

While conformal e-prediction is compatible with any valid e-variable, for concreteness we employ the *soft-rank e-variable*. This construction, which first appeared in Wang and Ramdas (2022) and Koning (2025), and was later applied in the context of conformal prediction by Balinsky and Balinsky (2024), takes the following form:

$$E = \frac{S(X_{\text{test}}, Y_{\text{test}})}{\frac{1}{n+1}\left(\sum_{i=1}^{n} S(X_i, Y_i) + S(X_{\text{test}}, Y_{\text{test}})\right)}. \quad (2)$$

This quantity defines a valid e-value as long as the scores are exchangeable and non-negative. Note that it is meaningful only when the score function S is negatively oriented; that is, lower scores indicate better predictions. In the remainder of this paper, we assume that these conditions hold.

Intuitively, the soft-rank e-variable construction mirrors the logic of traditional conformal prediction: the test score cannot be disproportionately large relative to the average calibration scores. However, unlike traditional conformal prediction that uses rank-based comparisons, the soft-rank e-variable directly compares the actual score values.

Using e-values goes beyond simply applying Markov's inequality to obtain a valid conformal set with a fixed miscoverage level α. They possess stronger properties, such as post-hoc guarantees, which allow for coverage guarantees even when the significance level α is chosen based on the observed data, something standard conformal prediction methods cannot provide.

2.2 Post-hoc Validity

We recall the key result on post-hoc validity with e-variables in conformal prediction, stated here for the specific case of the soft-rank e-variable.

Proposition 2.2 (Gauthier et al. (2025b))**.** *Consider a calibration set* $\{(X_i, Y_i)\}_{i=1}^n$ *and a test data point* $(X_{\text{test}}, Y_{\text{test}})$ *such that* $(X_1, Y_1), \ldots, (X_n, Y_n), (X_{\text{test}}, Y_{\text{test}})$ *are exchangeable. Let* $\tilde{\alpha} > 0$ *be any miscoverage level may depend on all of these data points. Then we have that:*

$$\mathbb{E}\left[\frac{\mathbb{P}(Y_{\text{test}} \notin \hat{C}_n^{\tilde{\alpha}}(X_{\text{test}}) \mid \tilde{\alpha})}{\tilde{\alpha}}\right] \leq 1, \quad (3)$$

where

$$\hat{C}_n^{\tilde{\alpha}}(x) := \left\{ y : \frac{S(x,y)}{\frac{1}{n+1}\left(\sum_{i=1}^{n} S(X_i, Y_i) + S(x,y)\right)} < \frac{1}{\tilde{\alpha}} \right\}. \quad (4)$$

When $\tilde{\alpha}$ *is a fixed constant independent of the data, the guarantee (3) reduces to the standard conformal guarantee (1).*

Proposition 2.2 enables us to obtain marginal guarantees for any coverage level, including those that depend on the data. This post-hoc validity holds because the definition of the e-value relies strictly on the nonconformity scores, which are exchangeable by assumption. Because an adaptive policy does not alter the computation of these underlying scores, selecting the miscoverage level $\tilde{\alpha}$ in a data-dependent manner does not retroactively compromise exchangeability. Building on this flexibility, we aim to design a coverage policy that adapts the miscoverage level in order to minimize the size of the resulting prediction sets. Informally, a coverage policy is simply a rule that maps the calibration scores and potential test scores to a (data-dependent) miscoverage level $\tilde{\alpha} \in (0, 1)$. We will provide a formal definition in Definition 2.5, but before doing so, we first examine how prediction set sizes behave in both the classification and regression settings.

This detour is useful since we want to design coverage policies that minimize these sizes, and the form they take in these two cases will guide us toward a simplified definition of coverage policies.

Remark 2.3 (Conformal set size in classification). Given a calibration set $(X_i, Y_i)_{i=1}^n$ and a test feature X_{test}, in classification problems the size of a conformal set $\text{Size}\big(\hat{C}_n^{\tilde{\alpha}}(X_{\text{test}})\big)$ at miscoverage level $\tilde{\alpha} \in (0,1)$ is given by

$$\#\left\{y \in \mathcal{Y} : \frac{S(X_{\text{test}}, y)}{\frac{1}{n+1}(\sum_{i=1}^n S(X_i, Y_i) + S(X_{\text{test}}, y))} < \frac{1}{\tilde{\alpha}}\right\}.$$

Because the cardinality operator is non-differentiable, it cannot be used directly in gradient-based optimization methods typical in machine learning. Since our ultimate goal is to train coverage policies via gradient-based methods, we replace the indicator inside the cardinality with a smooth surrogate. Specifically, we leverage the approximation:

$$\begin{aligned}&\sum_{y\in\mathcal{Y}} \mathbb{1}\left\{\frac{S(X_{\text{test}}, y)}{\frac{1}{n+1}\left(\sum_{i=1}^n S(X_i, Y_i) + S(X_{\text{test}}, y)\right)} < \frac{1}{\tilde{\alpha}}\right\} \\ &\approx \sum_{y\in\mathcal{Y}} \sigma\left(k\left(\frac{1}{\tilde{\alpha}} - \frac{S(X_{\text{test}}, y)}{\frac{1}{n+1}\left(\sum_{i=1}^n S(X_i, Y_i) + S(X_{\text{test}}, y)\right)}\right)\right),\end{aligned} \tag{5}$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function, and $k > 0$ is a parameter controlling the sharpness of the approximation. As $k \to \infty$, the sigmoid approaches a step function, and the approximation becomes exact. This smooth approximation enables efficient end-to-end training using standard neural network toolkits. It is worth noting, however, that alternative approaches to sigmoid smoothing, such as randomized smoothing proposed by Berthet et al. (2020), may lead to more robust optimization.

Remark 2.4 (Conformal set size in regression). The conformal set built using the soft-rank e-variable can be rewritten as:

$$\hat{C}_n^{\tilde{\alpha}}(X_{\text{test}}) = \left\{y \in \mathcal{Y} : S(X_{\text{test}}, y) < \frac{\sum_{i=1}^n S(X_i, Y_i)}{(n+1)\tilde{\alpha} - 1}\right\},$$

using basic algebraic simplifications. Now, consider a standard choice of score in regression, namely the mean absolute error (MAE):

$$S(x, y) = |f(x) - y|.$$

In this case, one can directly express the size of the conformal set,

$$\text{Size}\big(\hat{C}_n^{\tilde{\alpha}}(X_{\text{test}})\big) = 2\frac{\sum_{i=1}^n S(X_i, Y_i)}{(n+1)\tilde{\alpha} - 1}, \tag{6}$$

provided that $\tilde{\alpha} > 1/(n+1)$. This expression is differentiable almost everywhere with respect to $\tilde{\alpha}$, enabling gradient-based optimization.

We observe that the prediction sets considered here, based on the soft-rank e-value conformal set (4), depend on the calibration scores in a very simple way: they depend only on the sum of all calibration scores. In addition, in classification, the size also depends on the vector of potential test scores $(S(X_{\text{test}}, y))_{y\in\mathcal{Y}}$, while in regression the size does not depend on the test feature at all.

This observation motivates a natural simplification: we can define a coverage policy directly as a function of the sum of the calibration scores and an appropriate test summary statistic (which may be empty in the regression setting) as follows.

Definition 2.5 (Coverage policy)**.** A *coverage policy* is a function

$$\pi : \mathbb{R}_+ \times \mathcal{T} \to (0,1),$$

that maps the sum of calibration scores and a test summary statistic to a miscoverage level.[1] Here, $\mathcal{T}$ denotes the space of possible test statistics $t(X_{\text{test}})$: in classification, $t(X_{\text{test}})$ is the vector of potential scores $(S(X_{\text{test}}, y))_{y\in\mathcal{Y}}$, whereas in regression it is empty, reflecting that the conformal set size (6) does not depend on the test feature. The output

$$\pi\big(\textstyle\sum_{i=1}^n S(X_i, Y_i), t(X_{\text{test}})\big)$$

specifies the miscoverage level $\tilde{\alpha}$ to be used for constructing the conformal set $\hat{C}_n^{\tilde{\alpha}}(X_{\text{test}})$.

Definition 2.5 formalizes the idea that the miscoverage level can adapt based on summary information from the calibration set and the new test point, while remaining compatible with the guarantees of Proposition 2.2. Our goal is to design a coverage policy that selects the miscoverage level to produce prediction sets that are as informative as possible.

2.3 Training a Coverage Policy via a Leave-One-Out Procedure

For concreteness, we model the coverage policy using a neural network

$$\tilde{\alpha}_\theta : \mathbb{R}_+ \times \mathcal{T} \to (0,1),$$

parameterized by $\theta \in \Theta$ (though any model could be used). The goal of training is to select parameters θ that

[1] While we refer to it as a coverage policy, the function actually outputs a miscoverage level $\tilde{\alpha}$. This convention simplifies our notation, since the conformal set is defined directly using the inverse $1/\tilde{\alpha}$.

produce informative prediction sets while maintaining appropriate coverage.

To generate training samples, we employ a leave-one-out procedure: for a given calibration set of size n and $j \in \{1, \dots, n\}$, we treat the j-th sample as a pseudo test point and the remaining $n-1$ samples as a pseudo calibration set. Repeating this for all n points in the set produces n pseudo calibration-test pairs, each of which serves as a labeled example for training. The intuition behind this procedure is that, when leaving out a single data point, the aggregated information from the remaining $n-1$ points changes only slightly. As a result, each pseudo calibration-test pair provides nearly the same perspective as if we were observing a fresh sample with a fresh calibration set together with a new test point. This allows the network to infer a meaningful mapping from calibration sets to coverage levels without needing multiple independent sets.

For the j-th training example of the leave-one-out procedure, we denote the network's predicted miscoverage by

$$\tilde{\alpha}_\theta^j := \tilde{\alpha}_\theta\big(\textstyle\sum_{i \neq j} S(X_i, Y_i), t(X_j)\big), \tag{7}$$

and for the actual test point we write

$$\tilde{\alpha}_\theta^{\text{test}} := \tilde{\alpha}_\theta\big(\textstyle\sum_{i=1}^{n} S(X_i, Y_i), t(X_{\text{test}})\big). \tag{8}$$

Note that at training time the network is fed sums of $n-1$ scores, while at test time it receives a sum over n scores. The difference is negligible: leaving out one observation only slightly perturbs the aggregate, so the pseudo examples are essentially indistinguishable from the true test-time scenario.

We aim to tune θ so as to minimize the size of the prediction sets produced across the n training samples. However, naively minimizing prediction set size leads to a degenerate solution with coverage equal to zero. To avoid this, we introduce a regularization term that penalizes overly large miscoverage, thereby stabilizing the training process and discouraging degenerate solutions. The strength of this penalty is controlled by a user-specified parameter $\lambda > 0$, which allows practitioners to balance between the compactness of the prediction sets and the conservativeness of the coverage level. A smaller λ places more weight on obtaining smaller prediction sets, while a larger λ emphasizes achieving better coverage.

Specifically, we train $\tilde{\alpha}_\theta$ by minimizing the following objective:

$$\mathcal{L}_\lambda(\theta) = \frac{1}{n} \sum_{j=1}^{n} \text{Size}\big(\hat{\text{C}}_{\text{n-1}}^{\tilde{\alpha}_\theta^{\text{j}}}(\text{X}_\text{j})\big) + \lambda \cdot \tilde{\alpha}_\theta^{\text{j}}, \tag{9}$$

where $\hat{C}_{n-1}^{\tilde{\alpha}_\theta^j}(X_j)$ is the prediction set built from the pseudo calibration set obtained by leaving out the j-th sample, applied to the pseudo test feature X_j, with miscoverage level $\tilde{\alpha}_\theta^j$ predicted by the network. The first term encourages informative prediction sets, while the second term discourages the network from selecting excessively high miscoverage levels.

Algorithm 1: Training a Coverage Policy via Leave-One-Out

Input: Calibration set $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$, score function S, untrained neural network $\tilde{\alpha}_\theta$, regularization parameter λ, batch size B, optimizer

Output: Trained neural network $\tilde{\alpha}_\theta$

Step 1: Construct pseudo episodes via leave-one-out

for $j = 1, \dots, n$ **do**

  Define pseudo test point X_j and pseudo calibration set $\{(X_i, Y_i) : i \neq j\}$

Step 2: Update network parameters

Initialize θ randomly;

while *not converged* **do**

  Sample a minibatch $\mathcal{B} \subset \{1, \dots, n\}$ of size B;

  foreach $j \in \mathcal{B}$ **do**

    Compute coverage level $\tilde{\alpha}_\theta^j$ using (7);

    Construct conformal set $\hat{C}_{n-1}^{\tilde{\alpha}_\theta^j}(X_j)$ defined in (4) using score function S;

    Compute $\text{Size}\big(\hat{\text{C}}_{\text{n-1}}^{\tilde{\alpha}_\theta^{\text{j}}}(\text{X}_\text{j})\big)$;

  Compute loss defined in (9):

$$\mathcal{L}_\lambda(\theta) = \frac{1}{B} \sum_{j \in \mathcal{B}} \text{Size}\big(\hat{\text{C}}_{\text{n-1}}^{\tilde{\alpha}_\theta^{\text{j}}}(\text{X}_\text{j})\big) + \lambda \cdot \tilde{\alpha}_\theta^{\text{j}}$$

  Update θ by minimizing $\mathcal{L}_\lambda(\theta)$ using the optimizer;

return $\tilde{\alpha}_\theta$

By minimizing the loss (9) in the leave-one-out protocol, we fit a coverage policy that adaptively selects a miscoverage level $\tilde{\alpha}$ at test time. We summarize the training procedure of $\tilde{\alpha}_\theta$ in Algorithm 1. Using the neural network output by Algorithm 1, we can compute the test-time miscoverage $\tilde{\alpha}_\theta^{\text{test}}$ as defined in (8). The associated conformal set $\hat{C}_n^{\tilde{\alpha}_\theta^{\text{test}}}(X_{\text{test}})$ is optimized for informative prediction sets while still satisfying the marginal coverage guarantee (3), ensuring both practical efficiency and rigorous statistical reliability.

2.4 Selecting λ: Insights From the Constant-α Setting

The choice of the regularization term λ is critical in practice, as it directly impacts, among other things, the expected size of the prediction set at test time. One

practical strategy to select λ is to monitor the behavior of the network $\tilde{\alpha}_\theta$ during training and track the final average sizes of the training conformal sets.

To build intuition for this strategy, we first analyze an idealized setting in which the network output is constant, i.e., $\tilde{\alpha}_\theta \equiv \alpha$. In this case, we can show that the average size under the leave-one-out protocol provides an accurate estimate of the expected test-time prediction set size. More precisely, the estimation error is of order $O_P\left(1/\sqrt{n}\right)$ as the calibration size n increases (using the O_P notation from van der Vaart (1998)).

Theorem 2.6 (Leave-one-out proxy under constant α)**.** *Assume that the calibration samples (X_i, Y_i) are i.i.d., and let $\alpha \in (0,1)$ be a given target miscoverage level.*

Assume one of the following two cases holds:

(i) ***(Classification, sigmoid smoothing)*** *The size is given by the smooth sigmoid approximation defined in (5) with some parameter $k > 0$. The score function S is bounded and takes values in $[S_{\min}, S_{\max}]$ with $0 < S_{\min} \leq S_{\max} < \infty$, and $n > S_{\max}/S_{\min}$.*

(ii) ***(Regression, MAE score)*** *The size is defined in (6). The score function S is bounded and takes values in $[0, S_{\max}]$ with $S_{\max} < \infty$, and the miscoverage level satisfies $\alpha > 1/n$.*

Let $\overline{\mathrm{Size}}_n := \frac{1}{n}\sum_{j=1}^{n} \mathrm{Size}(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n}-1}(\mathrm{X_j}))$ denote the average size under the leave-one-out protocol. Then, under either (i) or (ii),

$$\left|\overline{\mathrm{Size}}_n - \mathbb{E}\left[\mathrm{Size}\big(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n}}(\mathrm{X_{test}})\big)\right]\right| = O_P\left(\frac{1}{\sqrt{n}}\right),$$

i.e., the average size consistently estimates the expected test-time size at rate $1/\sqrt{n}$ in probability.

Theorem 2.6, proved in Appendix A, shows that in the idealized constant-outpute case, the average size of the conformal sets provides a reliable estimate of the expected size at test time. This result provides a theoretical foundation for our approach: monitoring the leave-one-out sizes during training to approximate the expected test-time behavior of the conformal predictor.

In practice, the network output is generally a trained, input-dependent function rather than a constant. While Theorem 2.6 does not directly extend to this more realistic case, we observe empirically that the same approximation remains accurate: when the network varies smoothly with the input distribution, the leave-one-out average continues to track the expected test-time size. Our experiments in the next section substantiate this observation.[2]

[2]Our code is publicly available at https://github.com/GauthierE/adaptive-coverage-policies.

Algorithm 2: Two-Stage λ-Selection: Bracketing then Bisection

Input: Calibration set $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, score function S, target size $M > 0$, tolerance $\varepsilon > 0$, untrained neural network $\tilde{\alpha}_\theta$, initial $\lambda > 0$

Output: λ_M such that $\mathbb{E}\left[\mathrm{Size}\big(\hat{\mathrm{C}}^{\tilde{\alpha}_\theta^{\mathrm{test}}}_{\mathrm{n}}(\mathrm{X_{test}})\big)\right] \approx \mathrm{M}$

Step 1: Expansion phase to bracket M
repeat
  Train $\tilde{\alpha}_\theta$ with parameter λ using Algorithm 1;
  Compute $\overline{\mathrm{Size}}_n(\lambda) := \frac{1}{n}\sum_{j=1}^{n} \mathrm{Size}(\hat{\mathrm{C}}^{\tilde{\alpha}^{\mathrm{j}}_\theta}_{\mathrm{n}-1}(\mathrm{X_j}))$;
  if $\overline{\mathrm{Size}}_n(\lambda) < M$ **then**
    $\lambda \leftarrow 2\lambda$;
  else
    $\lambda \leftarrow \lambda/2$;
until $\overline{\mathrm{Size}}_n(\lambda)$ *crosses* M;
Set λ_{low} and λ_{high} as the two most recent values of λ bracketing M;

Step 2: Bisection refinement
repeat
  $\lambda \leftarrow (\lambda_{\mathrm{low}} + \lambda_{\mathrm{high}})/2$;
  Train $\tilde{\alpha}_\theta$ with parameter λ using Algorithm 1;
  Compute $\overline{\mathrm{Size}}_n(\lambda)$;
  if $\overline{\mathrm{Size}}_n(\lambda) < M$ **then**
    $\lambda_{\mathrm{low}} \leftarrow \lambda$;
  else
    $\lambda_{\mathrm{high}} \leftarrow \lambda$;
until $\left|\overline{\mathrm{Size}}_n(\lambda) - M\right| \leq \varepsilon$;
return $\lambda_M \leftarrow \lambda$

Building on this intuition, Algorithm 2 describes our procedure for selecting the regularization parameter λ given a desired target average prediction set size M. The goal is to find a λ such that the empirical average size under the leave-one-out protocol is close to M, which, by Theorem 2.6, implies that the expected test-time size will also be close to M.

A key ingredient in this procedure is the monotonicity of the leave-one-out size with respect to λ. In the constant-α setting, Proposition 2.7, proved in Appendix B, establishes this property formally. Monotonicity ensures that we can safely use a bracketing-and-bisection strategy: once we identify two values of λ such that the leave-one-out average lies below and above M, repeated halving of the interval guarantees convergence to the desired value. Empirically, we observe that this monotonicity approximately holds in the trained, input-dependent setting, ensuring the reliability of the bracketing-and-bisection procedure in practice.

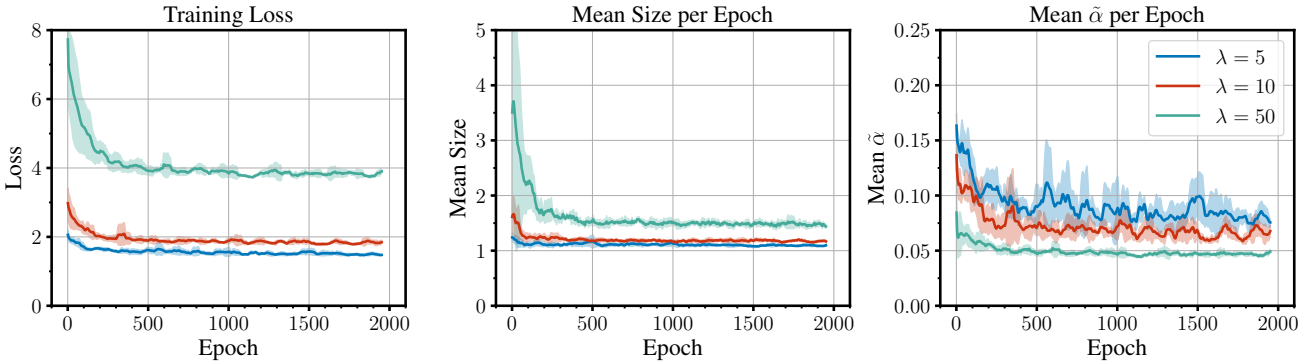

Figure 1: Training curves for $\lambda \in \{5, 10, 50\}$, averaged over 5 runs and smoothed with a moving average of size 50 for clarity. Shaded regions show ± 1 standard deviation across runs. **Left:** training loss. **Center:** mean set size. **Right:** mean adaptive miscoverage $\tilde{\alpha}$.

Proposition 2.7 (Monotonicity of leave-one-out size under constant α)**.** *Define*

$$\overline{\mathrm{Size}}_n(\lambda) := \frac{1}{n}\sum_{j=1}^{n} \mathrm{Size}\big(\hat{\mathrm{C}}_{\mathrm{n-1}}^{\alpha^*(\lambda)}(\mathrm{X_j})\big),$$

where

$$\alpha^*(\lambda) := \operatorname*{argmin}_{\alpha\in(0,1)} \frac{1}{n}\sum_{j=1}^{n} \mathrm{Size}(\hat{\mathrm{C}}_{\mathrm{n-1}}^{\alpha}(\mathrm{X_j})) + \lambda\alpha,$$

and assume that this minimizer exists.

Assume moreover that the size function $\mathrm{Size}(\hat{\mathrm{C}}_{\mathrm{n-1}}^{\alpha}(\mathrm{X_j}))$ *is non-increasing in* α*, which holds for any conformal set constructed using an e-value and threshold* $1/\alpha$*, in particular for the conformal set defined in (4).*

Then $\overline{\mathrm{Size}}_n(\lambda)$ *is non-decreasing in* λ*.*

3 EXPERIMENTAL EVALUATION

To demonstrate the effectiveness of our approach, we conduct experiments on the CIFAR-10 dataset (Krizhevsky, 2009), a standard benchmark in computer vision consisting of 60,000 32×32 color images evenly distributed across 10 object classes (such as airplanes, cats, and trucks). The dataset is split into 50,000 training examples and 10,000 test examples.

We train a deep neural network, denoted by f, on the full CIFAR-10 training set and treat it as a black-box predictor throughout our experiments. Specifically, we use an EfficientNet-B0 model (Tan and Le, 2019) trained to minimize the cross-entropy loss using stochastic gradient descent (SGD) with momentum 0.9, a learning rate of 0.1, weight decay of 5×10^{-4}, and cosine annealing over 100 epochs. We use a batch size of 512 and apply standard data augmentation techniques during training. At the end of training, the model f achieves a training accuracy of 98.6% and a test accuracy of 91.1%.

For our experiments, we choose the cross-entropy as the score function:

$$S(x, y) = -\log p_f(y|x),$$

where $p_f(y|x)$ is the probability that the pretrained model f assigns to label y for a given input image x.

To construct the data needed for training an adaptive coverage policy, we randomly split the CIFAR-10 test set into a calibration set and a remaining set from which we randomly sample a test point. In our experiments, we fix the calibration set size to $n = 100$.

For the neural network $\tilde{\alpha}_\theta$, we use a simple architecture consisting of a fully connected feedforward network with one hidden layer of 32 units and ReLU activation. The output layer has a single neuron followed by a sigmoid activation to produce outputs between 0 and 1.

We optimize the loss function (9) using sigmoid smoothing (see Remark 2.3) with $k = 100$, and the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 1×10^{-3}. Training is performed with a batch size of 64 over 2000 epochs.

For a calibration set sampled uniformly at random, Figure 1 illustrates the training dynamics of the coverage policy produced by Algorithm 1 under different choices of the regularization strength $\lambda \in \{5, 10, 50\}$. The training curves decrease and converge, indicating that the model is optimizing effectively. Moreover, both the prediction set size and the expected miscoverage stabilize asymptotically as training progresses. This effective convergence highlights the value of the leave-one-out procedure, which provides sufficient signal for successfully training a coverage policy.

Once training is complete, we can evaluate the model at test time. We begin by illustrating the results with a model $\tilde{\alpha}_\theta$ trained using $\lambda = 50$. The model is evaluated on 100 test points sampled uniformly at random from

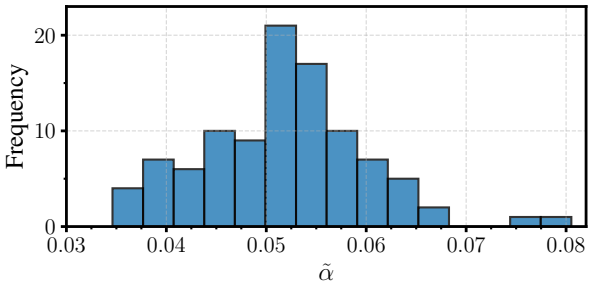

Figure 2: Distribution of adaptive miscoverage $\tilde{\alpha}$ across 100 randomly sampled test points.

the portion of the test set that is not used for calibration. Figure 2 displays the resulting distribution of adaptive miscoverage levels $\tilde{\alpha}$. The $\tilde{\alpha}$ values are reasonably spread, as would be expected if the coverage policy adapts to the varying difficulty of different predictions.

Figure 3 shows the conformal sets produced by our method, henceforth referred to as e-adaptive, with an average set size of 1.30. We compare this to two baselines. The first, e-fixed, also uses e-values but constructs conformal sets (4) with a fixed α equal to the empirical mean $\mathbb{E}[\tilde{\alpha}]$ computed from the 100 test points under our adaptive method. This yields a larger average set size of 1.42, highlighting the efficiency gains of adapting α to individual samples. The second baseline, p-fixed, employs standard conformal prediction with p-values and the same fixed α. While it achieves the smallest average set size of 1.01, it provides no principled way to select α in practice, making the resulting set sizes unpredictable. By contrast, both e-adaptive and e-fixed naturally support post-hoc selection of α, combining flexibility with valid coverage guarantees.

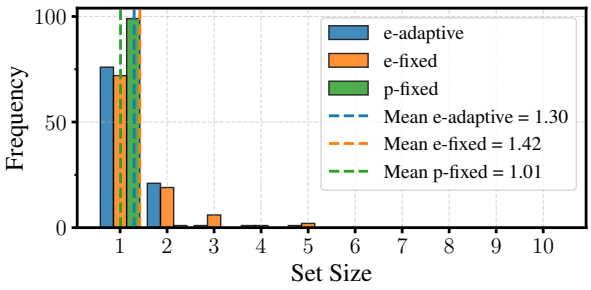

Figure 3: Distribution of conformal set sizes across 100 test points for the three methods.

To provide a clearer picture of the benefits of our e-adaptive method, we aggregate results over five independent calibration sets. For each calibration set, we apply Algorithm 1 to train a network $\tilde{\alpha}_\theta$ and construct the associated conformal sets for different values of λ. We then compute the average set size over 100 randomly chosen test points. We report the mean and standard deviation of these average set sizes across the five runs, and compare against the two baselines introduced above (e-fixed and p-fixed) in Table 1.

To select the regularization strength in practice, we can apply Algorithm 2, which iteratively adjusts λ to achieve some target mean prediction set size M. Figure 4 illustrates this process for an initial $\lambda = 40$, target $M = 2$, and tolerance $\varepsilon = 0.1$. The bracketing phase completes in a single iteration, as the mean size is below M for $\lambda = 40$ and above M for $\lambda = 80$. The subsequent bisection phase then refines λ through 60, 70, and 65, converging smoothly to a value that meets the target within the prescribed tolerance.

In this example, Algorithm 2 converges to a final mean set size of 2.00. We compare this value to the expected test-time prediction set, computed over 100 randomly sampled test points, yielding 2.07, which is very close to the final mean. This suggests that Theorem 2.6, proven in the idealized constant-output case, extends in practice to neural networks with input-dependent outputs. Moreover, the middle plot from Figure 1 shows that the monotonicity of the set size with respect to λ, established in Proposition 2.7 for the constant case, also holds for these more complex networks. Intuitively, this is expected because the network's $\tilde{\alpha}$ outputs vary smoothly with the inputs in practice. Moreover, the calibration sum changes little across data points, as it concentrates around n times the mean expected score. Figure 2 confirms that the distribution of network outputs remains smooth as test features vary. Together, these observations indicate that the theoretical properties derived for constant-output networks provide reliable guidance in the input-dependent setting.

To further support the theoretical guarantees in Theorem 2.6, we present an additional regression experiment in Appendix C.

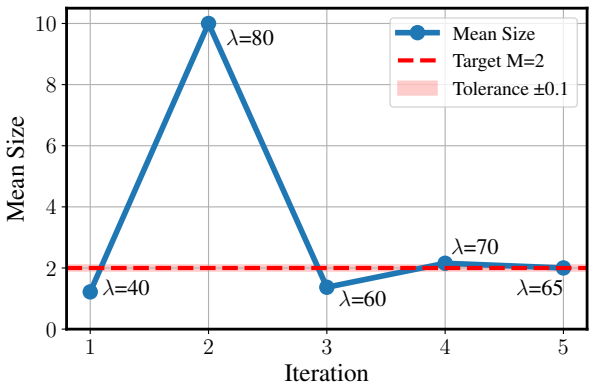

Figure 4: Evolution of λ and mean set size during Algorithm 2, showing the bracketing and bisection phases converging to the target $M = 2$.

Table 1: Prediction set size comparison across methods and regularization values.

Method	$\lambda = 5$	$\lambda = 10$	$\lambda = 50$	Post-hoc selection of α?
e-adaptive	1.21±0.10	1.23±0.12	1.63±0.19	✓
e-fixed	1.26±0.12	1.33±0.19	1.92±0.29	✓
p-fixed	0.96±0.05	0.98±0.05	1.18±0.12	✗

4 CONCLUSION

We have introduced an extension of conformal prediction that allows the miscoverage level to be set adaptively. By leveraging e-values and their post-hoc validity, our approach tailors the miscoverage level to the difficulty of each test sample via a coverage policy trained on the calibration set using a leave-one-out procedure. The method is theoretically grounded, providing adaptive miscoverage at deployment while maintaining valid marginal coverage. Unlike standard conformal methods, which offer no principled way to choose α and provide no insight into expected set size, our approach both adapts coverage to the data and enables optimizing the coverage policy for a desired expected test-time set size.

Note that when a history of conformal prediction episodes is available, the coverage policy can be trained directly on them, making leave-one-out unnecessary. However, in such cases, it is also possible to anticipate the optimal coverage level statistically and apply standard conformal prediction methods with p-values, which may be preferable. The strength of our method lies in the fact that the leave-one-out approach allows us to emulate pseudo conformal prediction episodes to train a coverage policy, without requiring access to any prior history of conformal predictions.

Acknowledgements

The authors thank the anonymous reviewers for their helpful feedback that improved this work.

Funded by the European Union (ERC-2022-SYG-OCEAN-101071601). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

This publication is part of the Chair "Markets and Learning," supported by Air Liquide, BNP PARIBAS ASSET MANAGEMENT Europe, EDF, Orange and SNCF, sponsors of the Inria Foundation.

This work has also received support from the French government, managed by the National Research Agency, under the France 2030 program with the reference "PR[AI]RIE-PSAI" (ANR-23-IACL-0008).

References

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

Anastasios N. Angelopoulos, Stephen Bates, Michael I. Jordan, and Jitendra Malik. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*, 2021.

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

Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications*. 2014.

Alexander A. Balinsky and Alexander David Balinsky. Enhancing conformal prediction using e-test statistics. In *Symposium on Conformal and Probabilistic Prediction with Applications*, 2024.

Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p-values. *The Annals of Statistics*, 51(1), 2023.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. In *Advances in Neural Information Processing Systems*, 2020.

Leonardo Cella and Ryan Martin. Valid inferential models for prediction in supervised learning problems. In *Proceedings of the International Symposium on Imprecise Probability: Theories and Applications*, volume 147, pages 72–82, 2021.

John Cherian, Isaac Gibbs, and Emmanuel Candès. Large language model validity via enhanced conformal prediction methods. In *Advances in Neural Information Processing Systems*, 2024.

Victor Chernozhukov, Kaspar Wüthrich, and Zhu Yinchu. Exact and robust conformal inference methods for predictive machine learning with dependent data. In *Conference On Learning Theory*, 2018.

Ben Chugg, Tyron Lardy, Aaditya Ramdas, and Peter Grünwald. On admissibility in post-hoc hypothesis testing. *International Journal of Approximate Reasoning*, 191:109634, 2026.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Few-shot conformal prediction with auxiliary tasks. In *International Conference on Machine Learning*, 2021.

Alex Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. In *Conference on Uncertainty in Artificial Intelligence*, 1998.

Etienne Gauthier, Francis Bach, and Michael I. Jordan. Backward conformal prediction. In *Advances in Neural Information Processing Systems*, 2025a.

Etienne Gauthier, Francis Bach, and Michael I. Jordan. E-values expand the scope of conformal prediction. *arXiv preprint arXiv:2503.13050*, 2025b.

Peter Grünwald, Rianne de Heide, and Wouter Koolen. Safe testing. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(5):1091–1128, 2024.

Peter Grünwald. Beyond Neyman-Pearson: E-values enable hypothesis testing with a data-driven alpha. *Proceedings of the National Academy of Sciences*, 121 (39):e2302098121, 2024.

Megan L. Head, Luke Holman, Rob Lanfear, Andrew T. Kahn, and Michael D. Jennions. The extent and consequences of p-hacking in science. *PLOS Biology*, 13(3):1–15, 03 2015.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Chancellor Johnstone and Bruce Cox. Conformal uncertainty sets for robust optimization. In *Proceedings of the Symposium on Conformal and Probabilistic Prediction and Applications*, volume 152, pages 72–90, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Nick W. Koning. Post-hoc α hypothesis testing and the post-hoc p-value. *arXiv preprint arXiv:2312.08040*, 2024.

Nick W. Koning. Measuring evidence against exchangeability and group invariance with e-values. *arXiv preprint arXiv:2310.01153*, 2025.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Rikard Laxhammar and Göran Falkman. Inductive conformal anomaly detection for sequential detection of anomalous sub-trajectories. *Annals of Mathematics and Artificial Intelligence*, 74(1–2):67–94, 2015.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

Junxian Liu, Hao Zeng, and Hongxin Wei. St-bcp: Tightening coverage bound for backward conformal prediction via non-conformity score transformation. *arXiv preprint arXiv:2602.01733*, 2026.

Harris Papadopoulos, Kostas Proedrou, Vladimir Vovk, and Alexander Gammerman. Inductive confidence machines for regression. In *European Conference on Machine Learning*, 2002.

Aaditya Ramdas and Ruodu Wang. Hypothesis testing with e-values. *Foundations and Trends® in Statistics*, 1(1-2):1–390, 2025.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421, 2008.

Glenn Shafer and Vladimir Vovk. *Game-Theoretic Foundations for Probability and Finance*. Wiley, 2019.

Joseph P. Simmons, Leif D. Nelson, and Uri Simonsohn. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11): 1359–1366, 2011.

Jiayuan Su, Jing Luo, Hongwei Wang, and Lu Cheng. API is enough: Conformal prediction for large language models without logit-access. In *Conference on Empirical Methods in Natural Language Processing*, 2024.

Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019.

Aad W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.

Vladimir Vovk. Conformal e-prediction. *Pattern Recognition*, 166:111674, 2025.

Vladimir Vovk and Ruodu Wang. E-values: Calibration, combination and applications. *The Annals of Statistics*, 49(3):1736–1754, 2021.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer-Verlag, 2005.

Ruodu Wang and Aaditya Ramdas. False discovery rate control with e-values. *Journal of the Royal Statistical Society Series B*, 84(3):822–852, 2022.

Ziyu Xu, Ruodu Wang, and Aaditya Ramdas. Post-selection inference for e-value based confidence intervals. *Electronic Journal of Statistics*, 18(1):2292 – 2338, 2024.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No] We will publicly release the code.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

A PROOF OF THEOREM 2.6

Theorem A.1 (Consistency of leave-one-out size)**.** *Assume that the calibration samples (X_i, Y_i) are i.i.d., and let $\alpha \in (0,1)$ be a given target miscoverage level.*

Assume one of the following two cases holds:

(i) ***(Classification, sigmoid smoothing)*** *The size is given by the smooth sigmoid approximation defined in (5) with some parameter $k > 0$. The score function S is bounded and takes values in $[S_{\min}, S_{\max}]$ with $0 < S_{\min} \le S_{\max} < \infty$, and $n > S_{\max}/S_{\min}$.*

(ii) ***(Regression, MAE score)*** *The size is defined in (6). The score function S is bounded and takes values in $[0, S_{\max}]$ with $S_{\max} < \infty$, and the miscoverage level satisfies $\alpha > 1/n$.*

Let $\overline{\mathrm{Size}}_n := \frac{1}{n}\sum_{j=1}^{n} \mathrm{Size}(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n}-1}(\mathrm{X_j}))$ denote average size under the leave-one-out protocol. Then, under either (i) or (ii),

$$\left|\overline{\mathrm{Size}}_n - \mathbb{E}\left[\mathrm{Size}\left(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n}}(\mathrm{X_{test}})\right)\right]\right| = O_P\left(\frac{1}{\sqrt{n}}\right),$$

i.e., the average size consistently estimates the expected test-time size at rate $1/\sqrt{n}$ in probability.

Proof. Let $\mu := \mathbb{E}[S(X,Y)]$ denote the expected score. We focus first on case (ii) (regression with MAE score). We want to compare the leave-one-out average size $\overline{\mathrm{Size}}_n := \frac{1}{n}\sum_{j=1}^{n} \mathrm{Size}(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n}-1}(\mathrm{X_j}))$ with the expected test-time size $\mathbb{E}[\mathrm{Size}(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n}}(\mathrm{X_{test}}))]$. For the MAE score, these quantities have the explicit forms

$$\overline{\mathrm{Size}}_n = \frac{1}{n}\sum_{j=1}^{n} \frac{2\sum_{i\neq j} S(X_i, Y_i)}{n\alpha - 1}, \qquad \mathbb{E}[\mathrm{Size}(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n}}(\mathrm{X_{test}}))] = \frac{2\mathrm{n}\mu}{(\mathrm{n}+1)\alpha - 1}.$$

Subtracting and rearranging terms gives

$$\begin{aligned}\left|\overline{\mathrm{Size}}_n - \mathbb{E}[\mathrm{Size}(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n}}(\mathrm{X_{test}}))]\right| &= \left|\frac{2(n-1)}{n\alpha - 1}\cdot\frac{1}{n}\sum_{i=1}^{n} S(X_i,Y_i) - \frac{2n\mu}{(n+1)\alpha - 1}\right| \\ &\le \underbrace{\frac{2(n-1)}{n\alpha-1}\left|\frac{1}{n}\sum_{i=1}^{n} S(X_i,Y_i) - \mu\right|}_{\text{fluctuation term}} + \underbrace{\left|\frac{2(n-1)\mu}{n\alpha-1} - \frac{2n\mu}{(n+1)\alpha-1}\right|}_{\text{bias term}}.\end{aligned}$$

The first term is a standard concentration term. Since the $S(X_i, Y_i)$ are i.i.d. and bounded by assumption, Hoeffding's inequality (Hoeffding, 1963) gives, for any $\delta > 0$,

$$\left|\frac{1}{n}\sum_{i=1}^{n} S(X_i,Y_i) - \mu\right| \le S_{\max}\sqrt{\frac{\log(2/\delta)}{2n}} \quad \text{with probability at least } 1-\delta.$$

The second term is purely deterministic and can be verified to satisfy

$$\left|\frac{2(n-1)\mu}{n\alpha-1} - \frac{2n\mu}{(n+1)\alpha-1}\right| = O\left(\frac{1}{n}\right).$$

Combining these two bounds, we see that

$$\left|\overline{\mathrm{Size}}_n - \mathbb{E}[\mathrm{Size}(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n}}(\mathrm{X_{test}}))]\right| = \mathrm{O_P}\left(\frac{1}{\sqrt{\mathrm{n}}}\right),$$

as claimed.

We now turn to case (i). The argument follows the strategy of Gauthier et al. (2025a, Theorem 3.1). We use the same notations: for each calibration index $j = 1, \dots, n$, define the random vector

$$\mathbf{E}^j := \left(\frac{S(X_j, y)}{\frac{1}{n}\left(\sum_{i \neq j} S(X_i, Y_i) + S(X_j, y) \right)} \right)_{y \in \mathcal{Y}},$$

and its population counterpart

$$\tilde{\mathbf{E}}^j := \left(\frac{S(X_j, y)}{\mu} \right)_{y \in \mathcal{Y}}.$$

Similarly, for the test point we write

$$\mathbf{E}^{\text{test}} := \left(\frac{S(X_{\text{test}}, y)}{\frac{1}{n+1}\left(\sum_{i=1}^{n} S(X_i, Y_i) + S(X_{\text{test}}, y) \right)} \right)_{y \in \mathcal{Y}}, \qquad \tilde{\mathbf{E}}^{\text{test}} := \left(\frac{S(X_{\text{test}}, y)}{\mu} \right)_{y \in \mathcal{Y}}.$$

Define the function

$$f : \mathbb{R}_+^{|\mathcal{Y}|} \to \mathbb{R}, \qquad (\mathbf{E}_y)_{y \in \mathcal{Y}} \mapsto \sum_{y \in \mathcal{Y}} \sigma\left(k \left(\frac{1}{\alpha} - \mathbf{E}_y \right) \right),$$

so that the size can be written as $f(\mathbf{E}^j)$ in the leave-one-out case, and as $f(\mathbf{E}^{\text{test}})$ at test time. First, observe that f is Lipschitz-continuous for the ℓ_∞ norm. Indeed, for $\mathbf{E}^1, \mathbf{E}^2 \in \mathbb{R}_+^{|\mathcal{Y}|}$ we have

$$\left| f(\mathbf{E}^1) - f(\mathbf{E}^2) \right| = \left| \sum_{y \in \mathcal{Y}} \left(\sigma(k(1/\alpha - \mathbf{E}^1_y)) - \sigma(k(1/\alpha - \mathbf{E}^2_y)) \right) \right| \leq \sum_{y \in \mathcal{Y}} \left| \sigma(k(1/\alpha - \mathbf{E}^1_y)) - \sigma(k(1/\alpha - \mathbf{E}^1_y)) \right|.$$

The slope of the sigmoid is uniformly bounded by $1/4$, hence σ is $1/4$-Lipschitz. Therefore:

$$|f(\mathbf{E}^1) - f(\mathbf{E}^2)| \leq \frac{k}{4} \sum_{y \in \mathcal{Y}} |\mathbf{E}^1_y - \mathbf{E}^2_y| \leq \frac{k|\mathcal{Y}|}{4} \|\mathbf{E}^1 - \mathbf{E}^2\|_\infty.$$

Thus f is Lipschitz-continuous for the ℓ_∞-norm with Lipschitz constant $L := k|\mathcal{Y}|/4$.

Now, we write

$$\begin{aligned}
\overline{\text{Size}}_n - \mathbb{E}[\text{Size}(\hat{\text{C}}^\alpha_{\text{n}}(\text{X}_{\text{test}}))] &= \frac{1}{n} \sum_{j=1}^{n} f(\mathbf{E}^j) - \mathbb{E}[f(\mathbf{E}^{\text{test}})] \\
&= \underbrace{\frac{1}{n} \sum_{j=1}^{n} \left(f(\mathbf{E}^j) - f(\tilde{\mathbf{E}}^j) \right)}_{=: T_1} + \underbrace{\frac{1}{n} \sum_{j=1}^{n} \left(f(\tilde{\mathbf{E}}^j) - \mathbb{E}[f(\tilde{\mathbf{E}}^{\text{test}})] \right)}_{=: T_2} + \underbrace{\mathbb{E}[f(\tilde{\mathbf{E}}^{\text{test}})] - \mathbb{E}[f(\mathbf{E}^{\text{test}})]}_{=: T_3}.
\end{aligned}$$

We now bound each term separately. Using the Lipschitz continuity of f with constant L, we have

$$|T_1| \leq \frac{1}{n} \sum_{j=1}^{n} \left| f(\mathbf{E}^j) - f(\tilde{\mathbf{E}}^j) \right| \leq \frac{L}{n} \sum_{j=1}^{n} \left\| \mathbf{E}^j - \tilde{\mathbf{E}}^j \right\|_\infty \leq L S_{\max} \frac{\sqrt{\frac{\log(2/\delta)}{2n}} + \frac{2}{n}}{\mu \left(S_{\min}/S_{\max} - \frac{1}{n} \right)} \quad \text{with probability} \geq 1 - \delta,$$

for all $\delta > 0$, where the last inequality follows from Gauthier et al. (2025a). By Hoeffding's inequality,

$$|T_2| \leq \sqrt{\frac{\log(2/\delta)}{2n}} \quad \text{with probability} \geq 1 - \delta.$$

Finally, for the bias term T_3, we have

$$|T_3| \leq \mathbb{E}\left[\left| f(\tilde{\mathbf{E}}^{\text{test}}) - f(\mathbf{E}^{\text{test}}) \right| \right] \leq L\, \mathbb{E}\left[\left\| \tilde{\mathbf{E}}^{\text{test}} - \mathbf{E}^{\text{test}} \right\|_\infty \right] \leq \frac{2LS^2_{\max}}{\mu S_{\min}} \frac{n+1}{n} \sqrt{\frac{\pi}{2(n+1)}},$$

again using Gauthier et al. (2025a). Combining the three terms with an union bound, we conclude that with probability $\geq 1-\delta$,

$$\left|\overline{\mathrm{Size}}_n - \mathbb{E}[\mathrm{Size}(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n}}(\mathrm{X}_{\mathrm{test}}))]\right| \leq \mathrm{LS}_{\max}\frac{\sqrt{\frac{\log(4/\delta)}{2\mathrm{n}}}+\frac{2}{\mathrm{n}}}{\mu\left(\mathrm{S}_{\min}/\mathrm{S}_{\max}-\frac{1}{\mathrm{n}}\right)}+\sqrt{\frac{\log(4/\delta)}{2\mathrm{n}}}+\frac{2\mathrm{LS}^2_{\max}}{\mu \mathrm{S}_{\min}}\frac{\mathrm{n}+1}{\mathrm{n}}\sqrt{\frac{\pi}{2(\mathrm{n}+1)}}.$$

This shows that

$$\left|\overline{\mathrm{Size}}_n - \mathbb{E}[\mathrm{Size}(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n}}(\mathrm{X}_{\mathrm{test}}))]\right| = \mathrm{O_P}\left(\frac{1}{\sqrt{\mathrm{n}}}\right).$$

□

Our proof relies on standard concentration inequalities to control deviations of the leave-one-out scores from their expectation; in particular, we employ Hoeffding's inequality for its simplicity. In the regression case, we explicitly assume that $\alpha > 1/n$ so that the conformal set sizes are well-defined, and that the scores are bounded above by $S_{\max}$ to ensure bounded scores and enable the application of Hoeffding's inequality. In the classification case, we additionally assume that the score is bounded below by a strictly positive value $S_{\min} > 0$ to avoid division by zero when concentration inequalities are applied to denominators. Finally, the condition $n > S_{\max}/S_{\min}$ also ensures that denominators are strictly positive.

B PROOF OF PROPOSITION 2.7

Proposition B.1 (Monotonicity of leave-one-out size under constant α)**.** *Define*

$$\overline{\mathrm{Size}}_n(\lambda) := \frac{1}{n}\sum_{j=1}^{n}\mathrm{Size}\big(\hat{\mathrm{C}}^{\alpha^*(\lambda)}_{\mathrm{n-1}}(\mathrm{X_j})\big),$$

where

$$\alpha^*(\lambda) := \operatorname*{argmin}_{\alpha\in(0,1)}\ \frac{1}{n}\sum_{j=1}^{n}\mathrm{Size}(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n-1}}(\mathrm{X_j})) + \lambda\alpha,$$

and assume that this minimizer exists.

Assume moreover that the size function $\mathrm{Size}(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n-1}}(\mathrm{X_j}))$ *is non-increasing in* α*, which holds for any conformal set constructed using an e-value and threshold* $1/\alpha$*, in particular for the conformal set defined in (4).*

Then $\overline{\mathrm{Size}}_n(\lambda)$ *is non-decreasing in* λ*.*

Proof. Let $0 < \lambda_1 < \lambda_2$, and denote the corresponding minimizers by $\alpha_1 := \alpha^*(\lambda_1)$ and $\alpha_2 := \alpha^*(\lambda_2)$. By optimality:

$$\frac{1}{n}\sum_{j=1}^{n}\mathrm{Size}(\hat{C}^{\alpha_1}_{n-1}(X_j)) + \lambda_1\alpha_1 \leq \frac{1}{n}\sum_{j=1}^{n}\mathrm{Size}(\hat{C}^{\alpha_2}_{n-1}(X_j)) + \lambda_1\alpha_2,$$

$$\frac{1}{n}\sum_{j=1}^{n}\mathrm{Size}(\hat{C}^{\alpha_2}_{n-1}(X_j)) + \lambda_2\alpha_2 \leq \frac{1}{n}\sum_{j=1}^{n}\mathrm{Size}(\hat{C}^{\alpha_1}_{n-1}(X_j)) + \lambda_2\alpha_1.$$

Adding these two inequalities yields

$$(\lambda_2-\lambda_1)(\alpha_2-\alpha_1) \leq 0 \quad \Rightarrow \quad \alpha_2 \leq \alpha_1.$$

Since $\mathrm{Size}(\hat{\mathrm{C}}^{\alpha}_{\mathrm{n-1}}(\mathrm{X_j}))$ is non-increasing in α, it follows that

$$\overline{\mathrm{Size}}_n(\lambda_2) = \frac{1}{n}\sum_{j=1}^{n}\mathrm{Size}(\hat{\mathrm{C}}^{\alpha_2}_{\mathrm{n-1}}(\mathrm{X_j})) \geq \frac{1}{\mathrm{n}}\sum_{\mathrm{j}=1}^{\mathrm{n}}\mathrm{Size}(\hat{\mathrm{C}}^{\alpha_1}_{\mathrm{n-1}}(\mathrm{X_j})) = \overline{\mathrm{Size}}_{\mathrm{n}}(\lambda_1),$$

which proves the claim. □

C ADDITIONAL EXPERIMENTS

We provide additional experiments in the regression setting to further demonstrate the empirical validity of our method and of Theorem 2.6. In this setup, we generate a synthetic regression dataset consisting of 100 training samples and 100 calibration samples. Each feature X_i is drawn independently from a uniform distribution on $[-5, 5]$, and the corresponding label is generated as

$$Y_i = 2X_i + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, 1).$$

The predictor f is taken to be a standard linear regression fit on the training data.

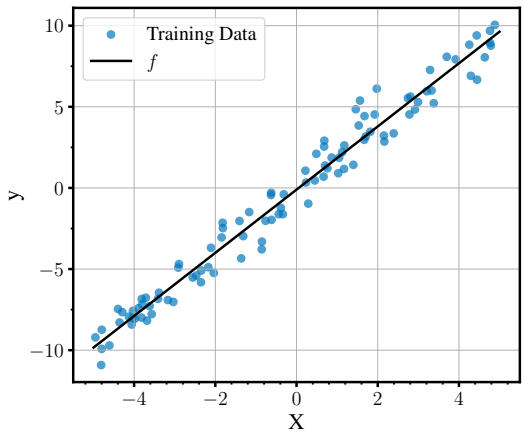

Figure 5: Visualization of the synthetic regression dataset.

For this experiment, we reuse essentially the same neural network architecture for $\tilde{\alpha}_\theta$ as in the main experiment described in the paper: a fully connected feedforward network with one hidden layer of 32 units and ReLU activation, followed by a single output neuron with sigmoid activation. The main difference is that here we initialize the network deliberately so that the output is initially close to 1. This ensures that $\tilde{\alpha}_\theta$ does not fall below $1/n$, which could otherwise lead to ill-defined conformal set sizes.

Training is performed with the Adam optimizer at a learning rate of 10^{-3} and batch size 32 over 200 epochs. The network inputs are the leave-one-out score for the left-out calibration point and the sum of the remaining scores. We plot the training dynamics for $\lambda \in \{10, 20, 50\}$ in Figure 6. The curves converge smoothly across all runs, demonstrating that the leave-one-out procedure allows the network to effectively train.

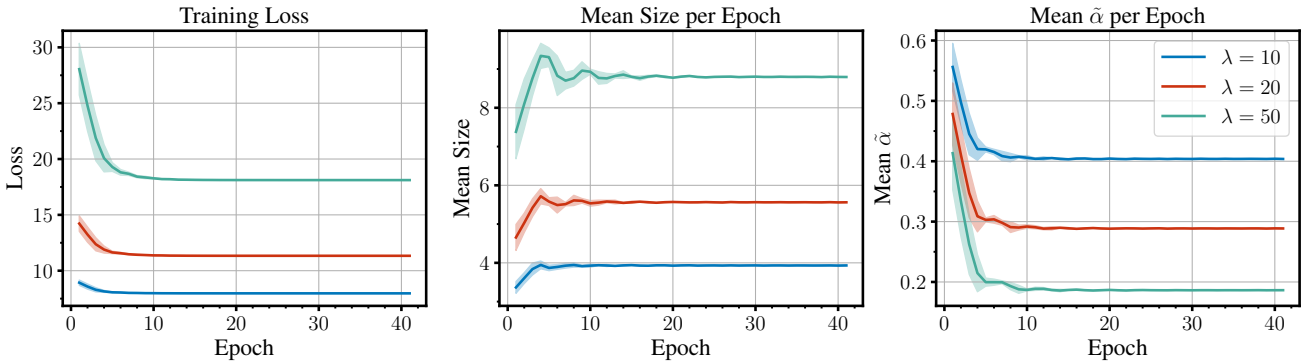

Figure 6: Training curves for $\lambda \in \{5, 10, 50\}$, averaged over 5 runs and smoothed with a moving average of size 10 for clarity. Shaded regions show ± 1 standard deviation across runs. **Left:** training loss. **Center:** mean set size. **Right:** mean adaptive $\tilde{\alpha}$.

Once the model is trained, we can use it to produce conformal sets. As in the main experiment of the paper, we sample 100 test points. Here, we focus on the validity of Theorem 2.6 in the regression setting. We consider a model trained with $\lambda = 50$. The leave-one-out estimator $\overline{\text{Size}}_n$ is equal to 8.82, while the expected size of the test

conformal set, computed from the 100 test conformal sets, is 9.31. The estimator is therefore relatively close to the true expected size. The gap between the two stems from the relatively small value of n in our experiment. Empirically, we observe that this difference shrinks as n increases, illustrating the effectiveness of the estimator.

Finally, we note an important practical limitation of this specific regression setup. As derived in Remark 2.4, the conformal set size for the MAE score using the soft-rank e-variable does not depend on the test feature X_{test}. Consequently, the resulting prediction sets maintain a constant width across the feature space and cannot adapt to varying levels of uncertainty or noise in the data. While this makes the sets overly large and rigid for practical, real-world regression tasks, this setup serves as a clean, differentiable testbed to empirically validate our procedure. For practical regression applications, combining our adaptive coverage framework with locally adaptive non-conformity scores or alternative e-variable constructions would be more appropriate.

D COMPLEXITY ANALYSIS AND RESOURCE DETAILS

In our complexity analysis, we assume that the base model f is a frozen black box. Therefore, the non-conformity scores $S(X_i, y)$ for all calibration points and all classes can be pre-computed prior to training. Once these scores are cached, querying S inside the training loop becomes a constant-time memory lookup. Under this pre-computation, in the classification setting with the loss (5), the time complexity of the optimization loop in Algorithm 1 is

$$O(T \cdot B \cdot (C_{\text{NN}} + n + |\mathcal{Y}|)),$$

where C_{NN} is the cost of a forward pass through the network $\tilde{\alpha}_\theta$, B is the batch size, T is the number of training epochs, and $|\mathcal{Y}|$ is the number of classes. In the regression setting with the loss (6), the time complexity reduces to

$$O(T \cdot B \cdot (C_{\text{NN}} + n)),$$

since conformal set sizes can be computed in constant time.

Algorithm 2 consists of a bracketing phase followed by bisection. In the bracketing phase, λ is repeatedly doubled or halved until a bracket around the target mean set size M is found, requiring $O(\log R)$ iterations if the initial λ differs by a factor of R from the solution. In the bisection phase, the interval is halved until the mean size is within a tolerance ε, yielding $O(\log(1/\varepsilon))$ iterations. Each iteration involves training the network for T epochs on minibatches of size B using Algorithm 1, giving an overall complexity of:

$$O((\log R + \log(1/\varepsilon)) \cdot T \cdot B \cdot (C_{\text{NN}} + n + |\mathcal{Y}|)),$$

All experiments were run on a machine with a 13th Gen Intel® Core™ i7-13700H CPU. Training times ranged from a few seconds in the regression setting to a few minutes in the classification setting on CIFAR-10.

To address the scalability of our approach to domains substantially larger than CIFAR-10 (such as ImageNet), we note that Algorithm 1 operates exclusively on the pre-computed non-conformity scores. It is entirely independent of the raw input dimensionality (e.g., image resolution). As established above, the training complexity scales linearly with the calibration set size n and the number of classes $|\mathcal{Y}|$. Even for massive datasets like ImageNet, the required calibration set size n typically remains small and manageable (e.g., $n \approx 1000$). Thus, for a large-scale task ($n = 1000$, $|\mathcal{Y}| = 1000$), the neural coverage policy $\tilde{\alpha}_\theta$ is simply trained over low-dimensional score vectors (of size 1001). The linear scaling in n and $|\mathcal{Y}|$: adds negligible computational overhead. The total wall-clock time is therefore overwhelmingly dominated by the initial forward passes of the base predictor f needed to cache the scores, ensuring our adaptive conformal wrapper remains highly scalable to large real-world tasks on standard hardware.