TOWARDS TRUSTWORTHY OBJECT DETECTION: ADAPTIVE BOUNDARY OPTIMIZATION WITH JOINT CLASSIFICATION AND LOCALIZATION RISK CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantifying image-level uncertainty in multi-object detection is paramount for safety-critical applications, such as autonomous driving. While Conformal Risk Control (CRC) provides marginal coverage guarantees, achieving rigorous risk control for individual images remains a significant challenge. To address this problem, we propose a novel approach that learns image-adaptive boundary extension thresholds to ensure consistent detection reliability. Specifically, we introduce the Conditional Adaptive Boundary Extension (CABE) framework, which leverages a differentiable surrogate loss based on soft Intersection-over-Ground-Truth (IoG) to directly optimize for improved conditional coverage. Experiments across three diverse safety-critical domains demonstrate that CABE not only achieves superior conditional coverage compared to standard methods but also significantly reduces the magnitude of boundary expansion. Our approach offers a robust pathway for reliable and interpretable uncertainty estimation in object detection, providing vital conditional guarantees required for high-stakes applications.

1 INTRODUCTION

Object Detection (OD) is pivotal in computer vision, underpinning safety-critical applications such as autonomous driving and medical imaging. While standard metrics like mean Average Precision measure average performance, they fail to provide the rigorous, statistical safety guarantees required for high-stakes deployment. This gap has driven the need for uncertainty quantification methods that can deliver verifiable assurances for OD predictions.

Conformal Prediction (CP) (Vovk et al., 2005) has emerged as a powerful, distribution-free framework for this purpose, offering valid coverage guarantees for any black-box model. Recently, Conformal Risk Control (CRC) (Angelopoulos et al., 2024) extended this paradigm to bound the expected risk for general loss functions below a user-specified level α, making it applicable to complex tasks like segmentation and detection.

However, a critical limitation of CRC in such vision tasks is its focus on marginal risk guarantees. While the average risk across the dataset is controlled, the conditional risk for individual test instances can vary significantly. In the context of OD, this implies that for some images, the predicted bounding boxes may fail to cover the target object completely. Such failures—leading to the loss of key features or the inclusion of extraneous background noise—can severely compromise the accuracy and reliability of downstream decisions.

To address this challenge, we propose the Conditional Adaptive Boundary Extension (CABE) framework. CABE ensures consistent reliability across diverse inputs by learning an image-adaptive policy to dynamically optimize boundary extensions. By formulating a differentiable surrogate for the coverage risk, we construct an end-to-end optimization method. This approach maximizes conditional coverage while strictly adhering to joint classification and localization safety budgets. Experimental results demonstrate that, compared to the CRC method, CABE not only improves conditional coverage but also reduces the set size, offering a principled path toward trustworthy object detection.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2 BACKGROUND

In this section, we formally define the object detection setup and the proposed risk control framework. Unlike standard metrics based on average Intersection-over-Union (IoU), we seek rigorous statistical guarantees for both classification recall and spatial containment. These guarantees are paramount for safety-critical applications, such as search and rescue, where failure to detect or accurately localize an object can have severe consequences.

2.1 OBJECT DETECTION

Let $X \in \mathbb{R}^{H\times W\times C}$ denote an input image. The corresponding ground truth, Y, consists of a set of $N(X)$ objects:

$$Y = \{Y_j\}_{j=1}^{N(X)}. \tag{1}$$

Each ground-truth object Y_j is characterized by a bounding box $B_j \in \mathbb{R}^4$ and a class label $k_j \in \{1,\dots,K\}$, such that $Y_j = (B_j, k_j)$. The bounding box is represented by coordinates $B_j = (x^j_{\min}, y^j_{\min}, x^j_{\max}, y^j_{\max})$.

An object detection model, $\hat{f}$, outputs a set of $\hat{N}(X)$ predicted objects, $\hat{Y} = \{\hat{Y}_p\}_{p=1}^{\hat{N}(X)}$. Each prediction $\hat{Y}_p$ is a tuple comprising a predicted bounding box $\hat{B}_p \in \mathbb{R}^4$ and a class probability vector $\hat{\pi}_p = (\hat{\pi}^p_1,\dots,\hat{\pi}^p_K)$, denoted as $\hat{Y}_p = (\hat{B}_p, \hat{\pi}_p)$. The predicted class label is given by $\hat{l}_p = \arg\max_c \hat{\pi}^p_c$.

To evaluate performance, we assume a standard matching protocol is applied. Let $\mathcal{M}(X)$ denote the set of indices of ground truth objects that are successfully matched. For any matched ground truth $j \in \mathcal{M}(X)$, let $\sigma(j)$ be the index of the corresponding prediction. Consequently, let $\hat{B}_{\sigma(j)}$ denote the predicted box assigned to ground truth Y_j by the detector's matching algorithm, and $\hat{\pi}_{\sigma(j)}$ denote the corresponding class scores.

2.2 CONFORMAL RISK CONTROL

CRC (Angelopoulos et al., 2024) generalizes conformal prediction to bound the expected value of any monotonic loss function ℓ at a user-specified level α. Ginven a calibration set $\{(X_i, Y_i)\}_{i=1}^n$, the objective is to identify a control parameter λ such that for a new test point (X_{n+1}, Y_{n+1}):

$$\mathbb{E}\left[\ell(\mathcal{C}_\lambda(X_{n+1}), Y_{n+1})\right] \le \alpha. \tag{2}$$

Assuming the loss is bounded by B and non-increasing in λ, a valid threshold $\hat{\lambda}$ can be computed empirically:

$$\hat{\lambda} = \inf\left\{\lambda : \frac{n}{n+1}\hat{R}_n(\lambda) + \frac{B}{n+1} \le \alpha\right\}, \tag{3}$$

where $\hat{R}_n(\lambda)$ represents the empirical risk calculated over the calibration set.

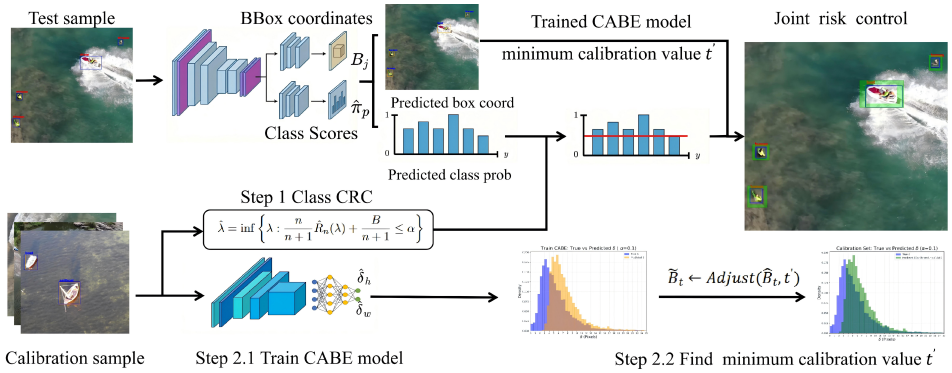


Figure 1: Schematic Overview of the CABE Framework Pipeline

2.3 CONFORMAL RISK CONTROL FOR OBJECT DETECTION

We formulate the safety requirement as a joint guarantee: ensuring ground truth objects are both correctly classified (high recall) and spatially contained.

Step 1: Image-Level Conformal Risk Control for Set-Valued Classification We use a threshold $\lambda \in [0, 1]$ to construct a *set-valued* class prediction for every detected box. A matched ground-truth object $Y_j = (B_j, k_j)$ with matched prediction index $\sigma(j)$ is deemed correctly classified if $\hat{\pi}_{\sigma(j),k_j} \geq \lambda$. We then define the **image-level classification False Negative Rate ($\mathrm{FNR}_{\mathrm{matched}}$)** over the matched objects. Note that this metric is computed conditionally on the subset of objects successfully recalled by the base detector. For brevity, we refer to this specific matched-conditional error simply as FNR throughout the remainder of this paper. We formulate it as:

$$\ell_{\mathrm{cls}}(X_i, Y_i; \lambda) = \begin{cases} 1 - \dfrac{\sum_{j \in \mathcal{M}(X_i)} \mathbb{I}\left(\hat{\pi}_{\sigma(j),k_j} \geq \lambda\right)}{|\mathcal{M}(X_i)|}, & \text{if } |\mathcal{M}(X_i)| > 0, \\ 0, & \text{if } |\mathcal{M}(X_i)| = 0. \end{cases} \tag{4}$$

By construction, $\ell_{\mathrm{cls}}(X_i, Y_i; \lambda) \in [0, 1]$ and it is *non-decreasing* in λ. We apply CRC to determine $\hat{\lambda}$ such that the expected FNR is bounded by α_1.

Step 2: Spatial Coverage Control via FNR Constraint We introduce a margin parameter $\delta \geq 0$ to expand predicted bounding boxes. For a matched pair, the expanded prediction is denoted as $\mathcal{B}_\delta(\hat{B}_{\sigma(j)})$. The coverage condition requires the ground truth box B_j to be strictly contained within the expanded prediction:

$$B_j \subseteq \mathcal{B}_\delta(\hat{B}_{\sigma(j)}) \iff B_j \cap \mathcal{B}_\delta(\hat{B}_{\sigma(j)}) = B_j. \tag{5}$$

Specifically, we consider two CRC-based boundary expansion strategies (Andéol et al., 2023). For the multiplicative case, the expanded box is obtained by adding offsets proportional to the box dimensions (where $\hat{w}$ is the width and $\hat{h}$ is the height of the original predicted bounding box):

$$\mathcal{B}^{\mathrm{mul}}_\delta(\hat{B}_{\sigma(j)}) = \left[x^j_{\min} - \delta\hat{w}, y^j_{\min} - \delta\hat{h}, x^j_{\max} + \delta\hat{w}, y^j_{\max} + \delta\hat{h}\right]. \tag{6}$$

Alternatively, for the additive case, as inspired by CRC additive methods, the expanded box is obtained by adding a constant scalar offset δ directly to the boundaries:

$$\mathcal{B}^{\mathrm{add}}_\delta(\hat{B}_{\sigma(j)}) = \left[x^j_{\min} - \delta, y^j_{\min} - \delta, x^j_{\max} + \delta, y^j_{\max} + \delta\right]. \tag{7}$$

We define the loss function $\ell_{box}(X_i, Y_i; \delta)$ as the spatial FNR for image X_i, calculated over the set of matched objects:

$$\ell_{box}(X_i, Y_i; \delta) = \begin{cases} 1 - \frac{1}{|\mathcal{M}(X_i)|} \sum_{j \in \mathcal{M}(X_i)} \mathbb{I}\left(B_j \subseteq \mathcal{B}_\delta(\hat{B}_{\sigma(j)})\right) & \text{if } |\mathcal{M}(X_i)| > 0, \\ 0 & \text{if } |\mathcal{M}(X_i)| = 0. \end{cases} \tag{8}$$

Our goal is to identify the minimal margin $\hat{\delta}$ such that the expected FNR is controlled at level α_2.

Joint Risk Analysis Since safety requires both accurate classification and localization, a ground truth object $Y_j = (B_j, k_j)$ is considered successfully retrieved only if it satisfies both conditions simultaneously: the class score of the matched prediction must exceed the threshold, i.e., $\hat{\pi}_{\sigma(j),k_j} \geq \hat{\lambda}$, and the ground truth box must be spatially contained within the expanded prediction, i.e., $B_j \subseteq \mathcal{B}_{\hat{\delta}}(\hat{B}_{\sigma(j)})$.

To evaluate the system at the image level, we analyze the expected joint risk over the matched objects (Following Timans et al. (2024), we typically set $\alpha_1 = 0.01$ and $\alpha_2 = 0.1$ or 0.05). The joint risk is bounded as follows:

$$\mathbb{E}\left[\mathcal{R}(X, Y; \hat{\lambda}, \hat{\delta})\right] = \mathbb{E}\left[1 - \frac{\sum_{j \in \mathcal{M}(X)} \mathbb{I}\left(\hat{\pi}_{\sigma(j),k_j} \geq \hat{\lambda} \wedge B_j \subseteq \mathcal{B}_{\hat{\delta}}(\hat{B}_{\sigma(j)})\right)}{|\mathcal{M}(X)|}\right] \leq \alpha_1 + \alpha_2. \tag{9}$$

Eq. 9 demonstrates that controlling the component risks at α_1 and α_2 results in a bounded aggregate system risk. Importantly, the denominator $|\mathcal{M}(X)|$ reflects that our risk analysis is conditioned on the subset of objects successfully recalled by the base detector; objects missed by the detector (unmatched) represent a separate failure mode not addressable by post-hoc calibration.

2.4 Problem Setup

While the joint risk guarantee derived in Eq. 9 ensures that the expected error rate is bounded, it remains a *marginal* guarantee. This global statistic does not preclude high variance in safety across individual instances. In practice, a fixed global margin $\hat{\delta}$ often leads to inconsistent behavior: it may be overly conservative for simple scenes while failing to sufficiently cover objects in complex scenarios.

Like Luo et al. (2026), to address this heterogeneity, we formulate a constrained optimization problem to determine an image-specific margin $\delta(X)$. Our objective is to minimize the variance of the coverage error, thereby encouraging the system to achieve the target coverage rate $(1-\alpha_2)$ consistently for every image. We seek the optimal $\delta(X)$ that minimizes the mean squared deviation from the target coverage over the matched objects, while strictly satisfying the joint safety budget:

$$\begin{aligned} \min \quad & \mathop{\mathbb{E}}_{(X,Y)\sim\mathcal{D}_{\text{test}}} \left[\frac{\sum_{j\in\mathcal{M}(X)} \mathbb{I}\left(B_j \subseteq \mathcal{B}_{\delta(X)}(\hat{B}_{\sigma(j)})\right)}{|\mathcal{M}(X)|} - (1-\alpha_2) \right]^2 \\ \text{s.t.} \quad & \mathop{\mathbb{E}}_{(X,Y)\sim\mathcal{D}_{\text{test}}} \left[\mathcal{R}(X,Y;\hat{\lambda},\delta(X)) \right] \leq \alpha_1 + \alpha_2. \end{aligned} \tag{10}$$

3 METHODOLOGY

In this section, we propose the **CABE** (Conditional Adaptive Boundary Extension) model. This is a novel end-to-end differentiable framework that adaptively predicts boundary expansion parameters based on image features, aiming to ensure that the proportion of uncovered ground-truth objects within each individual image is strictly controlled below a pre-defined risk threshold.

3.1 Prediction

The adaptive predictor, denoted as f_θ, is the core component of **image-level** adaptation. As shown in the framework, f_θ takes the input image X directly and predicts an image-specific expansion vector $\hat{\boldsymbol{\delta}}(X)$:

$$\hat{\boldsymbol{\delta}}(X) = f_\theta(X). \tag{11}$$

Here, $\hat{\boldsymbol{\delta}}(X) = (\delta_w, \delta_h)$ represents the anisotropic expansion ratios for the width and height directions, respectively, enabling more flexible coverage compared to a fixed scalar margin.

Remark 1 *The output dimension of $\hat{\boldsymbol{\delta}}(X)$ constitutes a flexible design choice. Theoretically, it can be formulated as a scalar ($\mathbb{R}^1$), a 2-dimensional vector ($\mathbb{R}^2$), or a 4-dimensional vector ($\mathbb{R}^4$). In Appendix 10, we conduct an ablation study evaluating these configurations. Our empirical results demonstrate that the 2-dimensional parameterization achieves a significantly lower coverage gap compared to the scalar approach. Furthermore, the 4-dimensional setting yields a coverage gap comparable to the 2-dimensional version but at the cost of higher computational latency. Therefore, we adopt the 2-dimensional output as the optimal trade-off between minimizing coverage deviation and maintaining inference speed. Furthermore, after training, we use $\mathcal{D}_2$ for calibration to maintain marginal coverage. The calibration value t is consistent with that in Equation 6 or 7, which can be either additive or multiplicative.*

3.2 Differentiable Training Objective

As illustrated in Figure 2, the magnitude of boundary expansion required to satisfy risk constraints varies significantly across images of different complexities. Consequently, methods relying on a

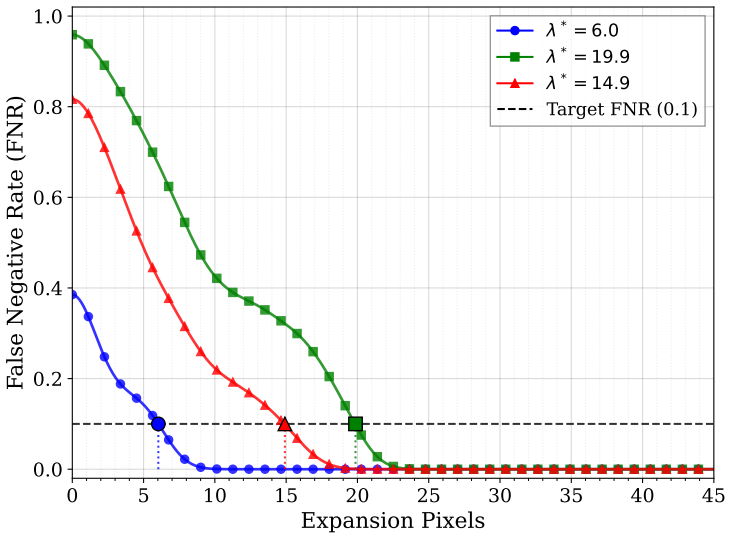


Figure 2: The relationship between Expansion pixels and the empirical FNR across different images.

single global threshold fail to guarantee conditional risk control. Furthermore, the mapping from expansion pixels to empirical risk is inherently discrete and non-differentiable. To enable end-to-end optimization, we construct a smooth, differentiable surrogate for the empirical coverage risk.

Differentiable Coverage Approximation Given a predicted box $\hat{B}$ and a ground truth box B, we apply the predicted expansion $\hat{\boldsymbol{\delta}}$ to generate an expanded box $\mathcal{B}_\delta(\hat{B})$.

To make this differentiable, we define the *Intersection-over-Ground-Truth* (IoG) ratio. Let $\hat{B}_{\sigma(j)}$ denote the predicted box assigned to ground truth Y_j by the detector's matching algorithm. The continuous containment ratio ρ_j is calculated as:

$$\rho_j(X;\hat{\boldsymbol{\delta}}) = \frac{\text{Area}(B_j \cap \mathcal{B}_{\hat{\boldsymbol{\delta}}}(\hat{B}_{\sigma(j)})) + \epsilon}{\text{Area}(B_j) + \epsilon}, \tag{12}$$

where ϵ is a smoothing term. Note that $\rho_j \in [0,1]$, where $\rho_j = 1$ implies complete spatial containment ($B_j \subseteq \mathcal{B}_{\hat{\boldsymbol{\delta}}}(\hat{B}_{\sigma(j)})$).

To approximate the binary indicator function $\mathbb{I}(\cdot)$ used in the risk calculation, we apply a power transformation with a temperature parameter $\beta \geq 1$. This yields a soft coverage score for the entire image, averaged over the matched objects:

$$\hat{S}(X;\hat{\boldsymbol{\delta}}) = \frac{1}{|\mathcal{M}(X)|} \sum_{j \in \mathcal{M}(X)} \Big(\rho_j(X;\hat{\boldsymbol{\delta}})\Big)^\beta. \tag{13}$$

As β increases, this soft score provides a tighter upper bound on the true coverage count, penalizing even minor boundary violations. Note that if $|\mathcal{M}(X)| = 0$, the score is undefined (or 0), but such images do not contribute to the gradient for expansion learning.

Training Objective Our goal is to learn the image-specific expansion $\hat{\boldsymbol{\delta}}(X)$ such that the expected coverage rate matches the target safety level $(1-\alpha_2)$ while keeping the expansion minimal. We formulate the training objective as a regression problem against the target risk budget, augmented with a regularization term:

$$\mathcal{L}_{\text{CABE}} = \mathop{\mathbb{E}}_{(X,Y)\sim\mathcal{D}_{\text{train}}} \left[\Big(\hat{S}(X;\hat{\boldsymbol{\delta}}) - (1-\alpha_2)\Big)^2\right] + \lambda_{\text{reg}} \mathop{\mathbb{E}}_{X\sim\mathcal{D}_{\text{train}}} \left[\|\hat{\boldsymbol{\delta}}(X)\|_2^2\right]. \tag{14}$$

The first term minimizes the deviation of the soft coverage score from the required safety guarantee $(1-\alpha_2)$. The second term, weighted by λ_{reg}, directly penalizes the magnitude of the predicted expansion $\hat{\boldsymbol{\delta}}$. This regularization encourages the model to predict the smallest necessary boundary extension required to satisfy the safety constraint, thereby preventing overly conservative predictions.

Algorithm 1 Conditional Adaptive Boundary Extension

Input: Calibration set $\mathcal{D}_{\text{cal}}$, Test set $\mathcal{D}_{\text{test}}$, Object Detector $\hat{f}$, Classification risk α_1, Spatial risk α_2, Regularization weight λ_{reg}.
Randomly split $\mathcal{D}_{\text{cal}}$ into $\mathcal{D}_1$, $\mathcal{D}_2$.
▷ ***Phase 1: Classification Risk Control***
Compute the maximal threshold $\hat{\lambda}$ on $\mathcal{D}_{\text{cal}}$ satisfying $\frac{|\mathcal{D}_{\text{cal}}|}{|\mathcal{D}_{\text{cal}}|+1}\hat{R}^{cls}_{|\mathcal{D}_{\text{cal}}|}(\hat{\lambda}) + \frac{1}{|\mathcal{D}_{\text{cal}}|+1} \leq \alpha_1$.
▷ ***Phase 2: Train Adaptive Predictor on*** $\mathcal{D}_1$
for epoch in training epochs **do**
  for each $(X_i, \mathcal{M}_k(X_i)) \in \mathcal{D}_1$ **do**
    Predict expansion vector: $\hat{\boldsymbol{\delta}}_i \leftarrow f_\theta(X_i)$.
    Generate expanded boxes: $\mathcal{B}_{\hat{\boldsymbol{\delta}}}(\hat{B})$.
    Compute $\hat{S}(X_i; \hat{\boldsymbol{\delta}}_i)$ (see Eq. 13).
    Compute $\mathcal{L}_{\text{CABE}}$ loss (see Eq. 14).
    Update parameters θ by descending gradient $\nabla\mathcal{L}_{\text{CABE}}$.
  end for
end for
▷ ***Phase 3: Spatial Offset Calibration on*** $\mathcal{D}_2$
Compute base expansion $\hat{\boldsymbol{\delta}}_j = f_\theta(X_j)$.
Define risk function: $L^{box}_j(t) = \ell_{box}(X_j, Y_j; \hat{\boldsymbol{\delta}}_j(t))$.
Find the minimal scalar $\hat{t}$ such that: $\frac{|\mathcal{D}_2|}{|\mathcal{D}_2|+1}\hat{R}^{box}_{|\mathcal{D}_2|}(\hat{t}) + \frac{1}{|\mathcal{D}_2|+1} \leq \alpha_2$.
▷ ***Phase 4: Inference on*** $\mathcal{D}_{\text{test}}$
for each test image $X_i \in \mathcal{D}_{\text{test}}$ **do**
  Classification: Remove labels where score $< \hat{\lambda}$.
  Adaptive Expansion: Predict $\hat{\boldsymbol{\delta}}_i \leftarrow f_\theta(X_i)$.
  Calibrated Adjustment: Apply final expansion parameters $\hat{\boldsymbol{\delta}}_i(\hat{t}) = \hat{\boldsymbol{\delta}}_i + \hat{t}\mathbf{1}_d$.
  Construct final boxes: $\widetilde{B}_i \leftarrow \mathcal{B}_{\hat{\boldsymbol{\delta}}_i(\hat{t})}(\hat{B})$.
end for

3.3 Finite-Sample Marginal Joint Risk Control

We now establish a finite-sample *marginal* joint risk control for CABE. Let f_θ be an image-to-expansion predictor that outputs a base expansion $\hat{\boldsymbol{\delta}} = f_\theta(X) \in \mathbb{R}^d_+$and $t \in \mathbb{R}_+$ be a scalar offset, yielding the adjusted expansion $\hat{\boldsymbol{\delta}}(t) := \hat{\boldsymbol{\delta}} + t\mathbf{1}_d$. We denote the two losses as $L^{cls}_i(\lambda) := \ell_{cls}(X_i, Y_i; \lambda)$ and $L^{box}_i(t) := \ell_{box}(X_i, Y_i; \hat{\boldsymbol{\delta}}_i(t))$, with their empirical risks

$$\hat{R}^{cls}_m(\lambda) := \frac{1}{m}\sum_{i=1}^{m} L^{cls}_i(\lambda),\ \hat{R}^{box}_m(t) := \frac{1}{m}\sum_{i=1}^{m} L^{box}_i(t). \tag{15}$$

Following Algorithm 1, we first calibrate the threshold λ on the calibration set $\mathcal{D}_{\text{cal}}$, then randomly split $\mathcal{D}_{\text{cal}}$ into $\mathcal{D}_1$ and $\mathcal{D}_2$ to train f_θ and calibrate the offset t, respectively. The theoretical guarantee is formalized as follows:

Theorem 3.1 (Marginal joint risk control)**.** *Let $\mathcal{D}_{\text{cal}} = \{(X_i, Y_i)\}_{i=1}^n$ and a test sample (X_{n+1}, Y_{n+1}) be exchangeable. Given a fixed detector $\hat{f}$ and matching protocol $(\mathcal{M}, \sigma)$, define the classification threshold as*

$$\hat{\lambda} := \sup\left\{\lambda \in [0,1] : \frac{n}{n+1}\hat{R}^{cls}_{\mathcal{D}_{\text{cal}}}(\lambda) + \frac{1}{n+1} \leq \alpha_1\right\}. \tag{16}$$

Let $\mathcal{D}_{\text{cal}}$ be randomly partitioned into $\mathcal{D}_1$ and $\mathcal{D}_2$. After training f_θ on $\mathcal{D}_1$, define the expansion offset as

$$\hat{t} := \inf\left\{t \in \mathbb{R}_+ : \frac{|\mathcal{D}_2|}{|\mathcal{D}_2|+1}\hat{R}^{box}_{\mathcal{D}_2}(t) + \frac{1}{|\mathcal{D}_2|+1} \leq \alpha_2\right\}. \tag{17}$$

The CABE predictor then satisfies the joint risk bound:

$$\mathbb{E}\Big[\mathcal{R}(X_{n+1}, Y_{n+1}; \hat{\lambda}, \hat{\boldsymbol{\delta}}_{n+1}(\hat{t}))\Big] \leq \alpha_1 + \alpha_2. \tag{18}$$

Proofs are provided in the appendix A.2, where we further discuss conditions under which CABE yields image-conditional localization risk control.

4 EXPERIMENTS

This section presents the empirical evaluation of our proposed CABE method. We assess its performance in providing joint risk control for classification and localization across object detection tasks in various safety-critical domains. To validate the robustness of our approach, we evaluate it across three distinct datasets: Rescue Object Detection Meine (2023), Aerial Floating Objects (AFO) Ugorji (2024), and DroneVehicle Sun et al. (2022). Our experiments employ YOLOv8 (Jocher et al., 2023) and YOLOv10 Wang et al. (2024) as the base object detection models. For the CABE boundary expansion predictor $f_\theta(X)$, we utilize a ResNet-50 He et al. (2016) backbone and set the power parameter $\beta = 100$ for the soft mask calculation.

We systematically examined the performance of CABE and standard CRC (Angelopoulos et al., 2024) under two distinct calibration strategies: additive adjustment ($\mathrm{CABE}_{\mathrm{add}}$ / $\mathrm{CRC}_{\mathrm{add}}$) and multiplicative adjustment ($\mathrm{CABE}_{\mathrm{mul}}$ / $\mathrm{CRC}_{\mathrm{mul}}$). By comparing these method-strategy combinations, we quantified their specific impacts on efficiency and coverage. To ensure rigorous evaluation, data was strictly partitioned into disjoint sets for base model training, conformal calibration, and testing. Detailed dataset descriptions, data partitioning protocols, and further implementation hyperparameters can be found in Appendix A.4 and A.5.

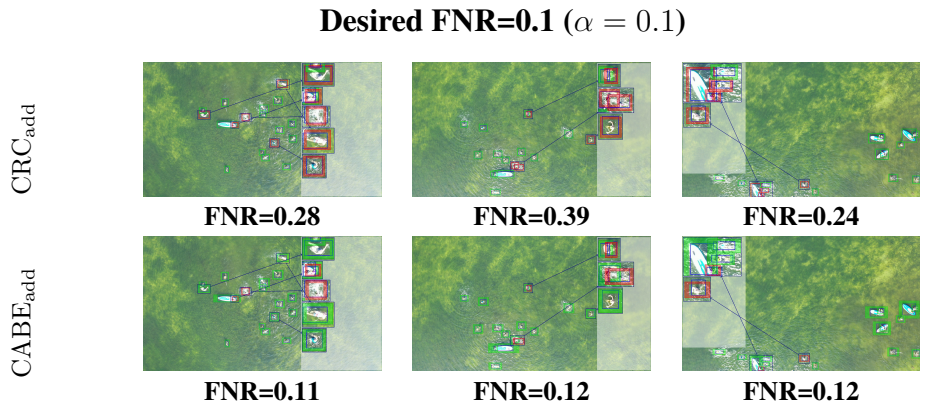


Figure 3: Qualitative comparison of $\mathrm{CRC}_{\mathrm{add}}$ (top) and $\mathrm{CABE}_{\mathrm{add}}$ (bottom) on the AFO dataset at $\alpha = 0.1$. Blue borders indicate the ground truth. Shaded regions represent the prediction ranges before and after expansion, where green and red denote successful and failed coverage, respectively; targets with failed coverage are additionally magnified in the blank spaces, and annotated with their corresponding FNR (False Negative Rate) below the images. Compared with $\mathrm{CRC}_{\mathrm{add}}$, $\mathrm{CABE}_{\mathrm{add}}$ demonstrates more consistent coverage and lower false negative rates.

As shown in Figures 3 and 4, we conducted qualitative analysis on the AFO and DroneVehicle datasets based on the YOLOv8 model. The results show that, compared to $\mathrm{CRC}_{\mathrm{add}}$, $\mathrm{CABE}_{\mathrm{add}}$ consistently approaches the target FNR across images of varying difficulty. Similar consistent improvements in maintaining target coverage are also observed in the Rescue dataset, with extended qualitative visualizations provided in Appendix A.6.

As illustrated in Table 1 and Figure 5, CABE consistently outperforms the corresponding CRC in terms of coverage gap across all datasets, regardless of the base object detection model or the choice between additive and multiplicative methods. Furthermore, CABE significantly surpasses heuristic-based stratified CRC baselines by effectively capturing continuous instance-level variations rather than relying on discrete binning (detailed in Appendix A.7). To address whether this adaptive boundary expansion excessively degrades localization precision, we additionally evaluated the IoU metrics (with detailed box plots available in Appendix A.8), demonstrating that CABE effectively generates tighter, instance-adaptive prediction sets. Furthermore, to rigorously validate these empirical improvements, we conducted paired t-tests (detailed in Appendix A.9), which statistically confirm that CABE significantly reduces the coverage gap and improves localization quality ($p < 0.05$)

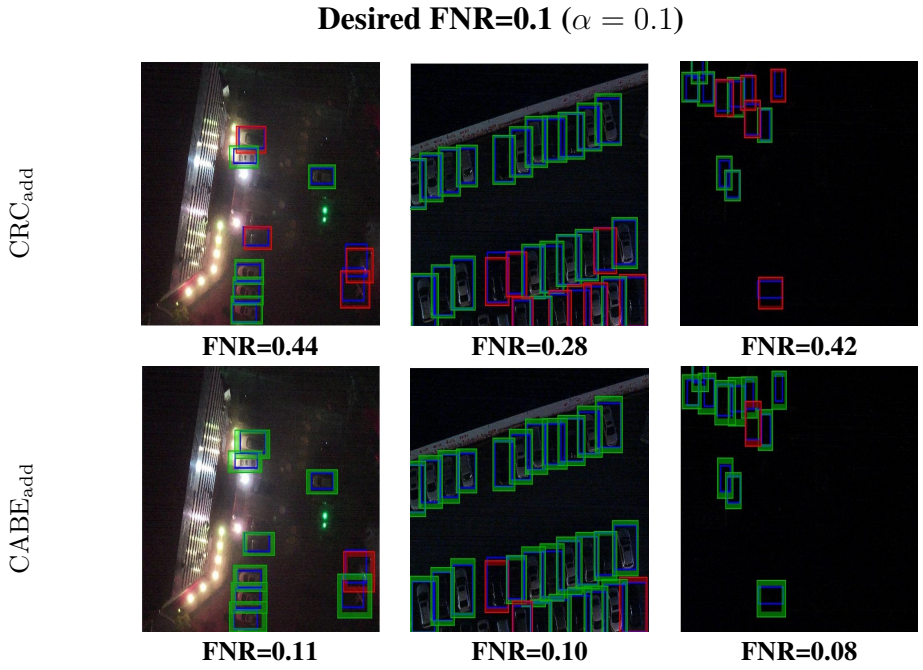


Figure 4: Qualitative comparison of $\mathrm{CRC}_{\mathrm{add}}$ (top) and $\mathrm{CABE}_{\mathrm{add}}$ (bottom) on the DroneVehicle dataset at $\alpha = 0.1$

Table 1: Joint Coverage, Box Coverage, Coverage Gap, Size, and Mean IoU Results at $\alpha = 0.1$ Across Different Models and Methods. Each dataset result is the mean and standard deviation of 20 random splits.

Dataset	Model	Method	Joint Coverage	Box Coverage	Coverage Gap	Size	IoU
AFO	YOLOv8	$\mathrm{CRC}_{\mathrm{add}}$	0.895 (0.009)	0.902 (0.010)	0.121 (0.006)	2.028 (0.032)	0.546 (0.007)
		$\mathrm{CRC}_{\mathrm{mul}}$	0.900 (0.012)	0.906 (0.012)	0.119 (0.008)	1.942 (0.017)	0.535 (0.005)
		$\mathrm{CABE}_{\mathrm{add}}$	0.895 (0.011)	0.902 (0.013)	0.111 (0.009)	**1.930 (0.030)**	**0.578 (0.010)**
		$\mathrm{CABE}_{\mathrm{mul}}$	0.903 (0.009)	0.909 (0.010)	**0.107 (0.007)**	1.959 (0.040)	0.571 (0.007)
	YOLOv10	$\mathrm{CRC}_{\mathrm{add}}$	0.895 (0.011)	0.902 (0.011)	0.122 (0.008)	1.916 (0.032)	0.554 (0.008)
		$\mathrm{CRC}_{\mathrm{mul}}$	0.896 (0.011)	0.902 (0.010)	0.120 (0.008)	1.805 (0.012)	0.556 (0.004)
		$\mathrm{CABE}_{\mathrm{add}}$	0.896 (0.011)	0.903 (0.011)	0.110 (0.005)	**1.800 (0.028)**	**0.582 (0.011)**
		$\mathrm{CABE}_{\mathrm{mul}}$	0.899 (0.010)	0.906 (0.010)	**0.109 (0.005)**	1.811 (0.030)	0.579 (0.010)
DroneVehicle	YOLOv8	$\mathrm{CRC}_{\mathrm{add}}$	0.896 (0.004)	0.901 (0.004)	0.117 (0.002)	1.677 (0.005)	0.605 (0.002)
		$\mathrm{CRC}_{\mathrm{mul}}$	0.896 (0.004)	0.902 (0.004)	0.112 (0.002)	1.621 (0.003)	0.625 (0.001)
		$\mathrm{CABE}_{\mathrm{add}}$	0.894 (0.004)	0.900 (0.004)	0.099 (0.002)	**1.605 (0.011)**	**0.653 (0.004)**
		$\mathrm{CABE}_{\mathrm{mul}}$	0.894 (0.004)	0.900 (0.004)	**0.099 (0.002)**	1.612 (0.011)	0.650 (0.004)
	YOLOv10	$\mathrm{CRC}_{\mathrm{add}}$	0.895 (0.004)	0.900 (0.004)	0.116 (0.003)	1.671 (0.008)	0.609 (0.003)
		$\mathrm{CRC}_{\mathrm{mul}}$	0.895 (0.003)	0.900 (0.004)	0.112 (0.002)	1.615 (0.003)	0.629 (0.002)
		$\mathrm{CABE}_{\mathrm{add}}$	0.894 (0.004)	0.899 (0.004)	0.099 (0.003)	**1.597 (0.011)**	**0.654 (0.004)**
		$\mathrm{CABE}_{\mathrm{mul}}$	0.894 (0.004)	0.899 (0.004)	**0.098 (0.003)**	1.603 (0.010)	0.652 (0.004)
Rescue	YOLOv8	$\mathrm{CRC}_{\mathrm{add}}$	0.899 (0.003)	0.900 (0.003)	0.140 (0.002)	2.309 (0.020)	0.507 (0.003)
		$\mathrm{CRC}_{\mathrm{mul}}$	0.898 (0.004)	0.900 (0.004)	0.128 (0.003)	1.567 (0.004)	**0.657 (0.002)**
		$\mathrm{CABE}_{\mathrm{add}}$	0.899 (0.004)	0.901 (0.004)	0.119 (0.002)	**1.555 (0.014)**	0.650 (0.005)
		$\mathrm{CABE}_{\mathrm{mul}}$	0.907 (0.006)	0.910 (0.006)	**0.115 (0.003)**	1.575 (0.029)	0.642 (0.010)
	YOLOv10	$\mathrm{CRC}_{\mathrm{add}}$	0.897 (0.004)	0.899 (0.004)	0.141 (0.003)	2.096 (0.021)	0.526 (0.004)
		$\mathrm{CRC}_{\mathrm{mul}}$	0.898 (0.002)	0.899 (0.003)	0.126 (0.002)	**1.484 (0.003)**	**0.668 (0.001)**
		$\mathrm{CABE}_{\mathrm{add}}$	0.897 (0.004)	0.899 (0.004)	0.120 (0.003)	1.485 (0.013)	0.666 (0.003)
		$\mathrm{CABE}_{\mathrm{mul}}$	0.906 (0.007)	0.907 (0.007)	**0.116 (0.004)**	1.503 (0.023)	0.658 (0.008)

across the vast majority of evaluated scenarios. Finally, to rigorously evaluate the reliability and efficiency of the proposed method in safety-critical scenarios, we extended our evaluations to a wider range of target risk levels ($\alpha \in \{0.05, 0.15, 0.20\}$) and varying IoG thresholds, with comprehensive quantitative analyses detailed in Appendix A.10 and A.11.

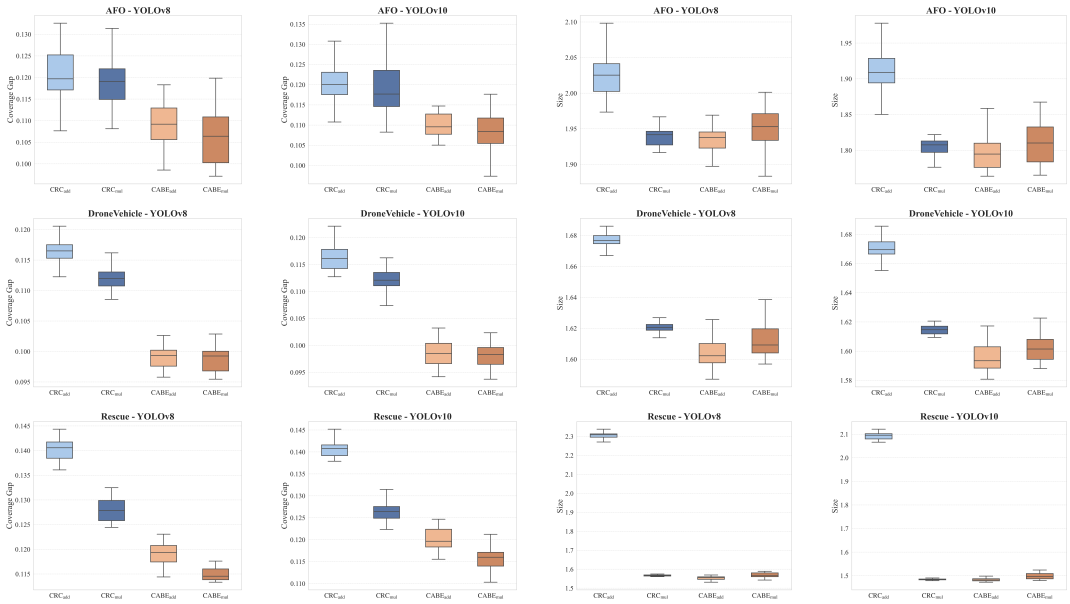


Figure 5: The box plot results for the Coverage Gap and Size obtained from different datasets and various base object detection models are presented. The target risk level is set to $\alpha = 0.1$. For each experimental setup, 20 random splits were conducted.

Furthermore, visualizations of image-level coverage density distributions confirm that CABE effectively concentrates sample density at the target coverage $(1 - \alpha)$ and avoids severe under-coverage failures. Detailed density plots and accompanying analyses are provided in Appendix A.12.

In addition, we illustrate the training progress of CABE in Appendix A.13. Regardless of the detection model or dataset, the loss function demonstrates high stability, rapidly converging. To verify the specific contributions of the neural network optimization and the post-hoc calibration step, we analyze the uncalibrated model efficiency versus the global calibration impact in Appendix A.14. Finally, the computational efficiency and optimal configuration of our framework are thoroughly validated; detailed time metrics, ablation studies on expansion dimensionality, and critical sensitivity analyses concerning hyperparameters and calibration set ratios are discussed in Appendix A.15, A.16, and A.17.

Remark 2 (Coverage Gap Limitations.) *Owing to the impacts of multiple factors, it is difficult to reduce the coverage gap to zero in practical applications, including finite sample effects and model capacity constraints. Despite these theoretical limitations, the CABE method proposed in this paper achieves the smallest coverage gap across all experimental scenarios, which verifies its superiority in learning effective image-adaptive feature patterns. The end-to-end differentiable optimization strategy adopted by CABE is underpinned by rigorous theoretical foundations, and it can directly optimize the target coverage, thereby providing more reliable conditional guarantees. This novel end-to-end paradigm is of great value for safety-critical object detection tasks.*

5 CONCLUSION

In this paper, we presented a principled framework for achieving conditional risk control, addressing the inherent limitations of marginal guarantees in traditional conformal prediction. Theoretical analysis and extensive empirical evaluations demonstrate that our method not only satisfies the prescribed risk budgets but also significantly enhances the efficiency of the predicted sets compared to existing baselines. This work provides a robust foundation for deploying uncertainty-aware models in high-stakes environments, offering a scalable path toward more trustworthy and interpretable machine learning systems. While acknowledging that our guarantees remain contingent on the base detector's recall and cannot fully recover inherently missed objects, future work will explore extending this framework to non-monotonic risk measures, such as IoU, as well as addressing these base-level recall limitations.

AI USE STATEMENT

Generative AI tools were used solely for language editing and LaTeX formatting assistance. They were not used for research ideation, methodology design, experiment design, data analysis, result interpretation, code generation, or the development of the scientific claims in this work. All AI-assisted material was reviewed and verified by the authors, who take full responsibility for the final content of this paper.

ETHICS STATEMENT

This work studies uncertainty-aware object detection for safety-critical application domains. The datasets used in our experiments are publicly available research datasets and are used in accordance with their respective licenses and intended research purposes. Our method provides statistical risk-control guarantees under the stated assumptions, but it does not eliminate errors caused by missed detections, distribution shifts, annotation errors, or failures of the underlying detector. Therefore, CABE should not be used as the sole basis for high-stakes or life-critical decisions; deployment should include appropriate human oversight, domain-specific validation, and safeguards against misuse. We do not claim that the proposed method is suitable for surveillance practices that violate privacy, legal, or ethical requirements.

REPRODUCIBILITY STATEMENT

We provide a detailed description of the problem formulation, CABE framework, differentiable training objective, and finite-sample marginal joint risk-control guarantee in the main paper. The appendix includes complete theoretical proofs, dataset partitioning protocols, implementation details, evaluation metrics, hyperparameter settings, sensitivity analyses, additional experimental results, and pseudocode for the core training and calibration procedures. To further support reproducibility, we will release the complete source code, trained model weights, and scripts required to reproduce the reported experiments upon acceptance.

REFERENCES

Léo Andéol, Thomas Fel, Florence De Grancey, and Luca Mossina. Confident object detection via conformal prediction and conformal risk control: an application to railway signaling. In *Conformal and Probabilistic Prediction with Applications*, pp. 36–55. PMLR, 2023.

Léo Andéol, Thomas Fel, Florence de Grancey, and Luca Mossina. Conformal prediction for trustworthy detection of railway signals. *AI and Ethics*, 4(1):157–161, 2024.

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

Anastasios Nikolas Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=33XGfHLtZg.

Konstantina Bairaktari, Jiayun Wu, and Steven Wu. Kandinsky conformal prediction: Beyond class-and covariate-conditional coverage. In *Forty-second International Conference on Machine Learning*, 2025.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

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

Vincent Blot, Alexandra Lorenzo de Brionne, Ines Sellami, Olivier Trassard, Isabelle Beau, Charlotte Sonigo, and Nicolas J-B Brunel. Efficient precision control in object detection models for enhanced and reliable ovarian follicle counting. In *International Workshop on Uncertainty for Safe Utilization of Machine Learning in Medical Imaging*, pp. 183–193. Springer, 2024.

Vincent Blot, Anastasios Nikolas Angelopoulos, Michael Jordan, and Nicolas JB Brunel. Automatically adaptive conformal risk control. In *International Conference on Artificial Intelligence and Statistics*, pp. 19–27. PMLR, 2025.

Joren Brunekreef, Eric Marcus, Ray Sheombarsing, Jan-Jakob Sonke, and Jonas Teuwen. Kandinsky conformal prediction: efficient calibration of image segmentation algorithms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4135–4143, 2024.

John Cherian, Isaac Gibbs, and Emmanuel Candes. Large language model validity via enhanced conformal prediction methods. *Advances in Neural Information Processing Systems*, 37:114812–114842, 2024.

Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. Distributional conformal prediction. *Proceedings of the National Academy of Sciences*, 118(48):e2107794118, 2021.

Vicky Copley, Greg Finlay, and Ben Hiett. The uncertain object: Application of conformal prediction to aerial and satellite images. In *The 13th Symposium on Conformal and Probabilistic Prediction with Applications*, pp. 73–89. PMLR, 2024.

Florence De Grancey, Jean-Luc Adam, Lucian Alecu, Sébastien Gerchinovitz, Franck Mamalet, and David Vigouroux. Object detection with probabilistic guarantees: A conformal prediction approach. In *International Conference on Computer Safety, Reliability, and Security*, pp. 316–329. Springer, 2022.

Tiffany Ding, Anastasios Angelopoulos, Stephen Bates, Michael Jordan, and Ryan J Tibshirani. Class-conditional conformal prediction with many classes. *Advances in neural information processing systems*, 36:64555–64576, 2023.

Di Feng, Ali Harakeh, Steven L Waslander, and Klaus Dietmayer. A review and comparative study on probabilistic object detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):9961–9980, 2021.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021.

Chao Gao, Liren Shan, Vaidehi Srinivas, and Aravindan Vijayaraghavan. Volume optimality in conformal prediction with structured prediction sets. In *Forty-second International Conference on Machine Learning*, 2025.

Isaac Gibbs, John J Cherian, and Emmanuel J Candès. Conformal prediction with conditional guarantees. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, pp. qkaf008, 2025.

Leying Guan. Localized conformal prediction: A generalized inference framework for conformal prediction. *Biometrika*, 110(1):33–50, 2023.

Chirag Gupta, Aleksandr Podkopaev, and Aaditya Ramdas. Distribution-free binary classification: prediction sets, confidence intervals and calibration. *Advances in Neural Information Processing Systems*, 33:3711–3723, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Rohan Hore and Rina Foygel Barber. Conformal prediction with local weights: randomization enables local guarantees. *arXiv preprint arXiv:2310.07850*, 2023.

Rafael Izbicki, Gilson Shimizu, and Rafael B Stern. Cd-split and hpd-split: Efficient conformal regions in high dimensions. *Journal of Machine Learning Research*, 23(87):1–32, 2022.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. URL `https://github.com/ultralytics/ultralytics`.

Christopher Jung, Georgy Noarov, Ramya Ramalingam, and Aaron Roth. Batch multivalid conformal prediction. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Dk7QQp8jHEo.

Jivat Neet Kaur, Michael I Jordan, and Ahmed Alaa. Conformal prediction sets with improved conditional coverage using trust scores. *arXiv preprint arXiv:2501.10139*, 2025.

Shayan Kiyani, George J Pappas, and Hamed Hassani. Conformal prediction with learned features. In *International Conference on Machine Learning*, pp. 24749–24769. PMLR, 2024.

Florian Kraus and Klaus Dietmayer. Uncertainty estimation in one-stage object detection. In *2019 ieee intelligent transportation systems conference (itsc)*, pp. 53–60. IEEE, 2019.

Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(1):71–96, 2014.

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.

Shuo Li, Sangdon Park, Xiayan Ji, Insup Lee, and Osbert Bastani. Towards pac multi-object detection and tracking. *arXiv preprint arXiv:2204.07482*, 2022.

Yangyi Li and Mengdi Huai. Quantifying uncertainty in natural language explanations of large language models for question answering. *arXiv preprint arXiv:2509.15403*, 2025.

Rui Luo and Zhixin Zhou. Conditional conformal risk adaptation. *arXiv preprint arXiv:2504.07611*, 2025a.

Rui Luo and Zhixin Zhou. Conformal thresholded intervals for efficient regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19216–19223, 2025b.

Rui Luo, Jie Bao, Xiaoyi Su, Wen Jung Li, and Suqun Cao. Enhancing image-conditional coverage in segmentation: Adaptive thresholding via differentiable miscoverage loss. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Gd2AiWes1J.

Julien Meine. Rescue object detection dataset. https://www.kaggle.com/datasets/julienmeine/rescue-object-detection, 2023. Accessed: 2025-12-22.

Luca Mossina, Adrien Mazoyer, Sébastien Gerchinovitz, et al. Conformal object detection by sequential risk control. *arXiv preprint arXiv:2505.24038*, 2025.

Sangdon Park, Osbert Bastani, Nikolai Matni, and Insup Lee. Pac confidence sets for deep neural networks via calibrated prediction. In *International Conference on Learning Representations*, 2020.

Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

Matteo Sesia and Yaniv Romano. Conformal prediction using conditional histograms. *Advances in Neural Information Processing Systems*, 34:6304–6315, 2021.

Yiming Sun, Jiliang Cao, Pengfei Zhu, and Kun Lan. Drone-based rgb-infrared cross-modality vehicle detection via transfer learning. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. doi: 10.1109/LGRS.2020.3042565.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Alexander Timans, Christoph-Nikolas Straehle, Kaspar Sakmann, and Eric Nalisnick. Adaptive bounding box uncertainties via two-step conformal prediction. In *European Conference on Computer Vision*, pp. 363–398. Springer, 2024.

Paolo Toccaceli and Alexander Gammerman. Combination of inductive mondrian conformal predictors. *Machine Learning*, 108:489–510, 2019.

I. R. Ugorji. Rescuefobj: Aerial floating objects variation. `https://www.kaggle.com/code/ugorjiir/rescuefobj`, 2024. Accessed: 2025-12-22.

Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pp. 475–490. PMLR, 2012.

Vladimir Vovk, David Lindsay, Ilia Nouretdinov, and Alex Gammerman. Mondrian confidence machine. *Technical Report*, 2003.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, et al. Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, 37:107984–108011, 2024.

Yunpeng Xu, Mufang Ying, Wenge Guo, and Zhi Wei. Two-stage risk control with application to ranked retrieval. *arXiv preprint arXiv:2404.17769*, 2024.

Mert Yuksekgonul, Linjun Zhang, James Y Zou, and Carlos Guestrin. Beyond confidence: Reliable models should also consider atypicality. *Advances in Neural Information Processing Systems*, 36: 38420–38453, 2023.

A APPENDIX

A.1 Related Work

A.1.1 Conformal prediction for object detection.

In contrast to many prevailing methods for quantifying model uncertainty in object detection Feng et al. (2021)—such as Deep Ensembles, Monte Carlo Dropout (MC-Dropout), or estimating variance from bounding boxes or logits Kraus & Dietmayer (2019), which often lack reliability in capturing epistemic uncertainty—conformal prediction offers a more attractive alternative. It does not require prior knowledge of the data distribution and is independent of the base object detection model, as we will discuss further.

Conformal prediction Vovk et al. (2005); Lei et al. (2018); Angelopoulos et al. (2024) and its variants Park et al. (2020); Bates et al. (2021); Angelopoulos et al. (2025) serve as natural post-processing methods that can provide statistically valid uncertainty estimates. Recent applications to object detection have emerged, primarily through approaches based on conformal prediction De Grancey et al. (2022); Andéol et al. (2023; 2024); Timans et al. (2024) and tolerance regions Li et al. (2022); Blot et al. (2024). While both frameworks construct and calibrate post-hoc prediction sets with statistical guarantees—albeit of slightly different natures—another line of work builds upon the Learn Then Test (LTT) framework Angelopoulos et al. (2025). This method Xu et al. (2024); Mossina et al. (2025) employs a multiple testing strategy and can handle more than one parameter.

Among these, the work in Timans et al. (2024) and Li et al. (2022) offers notable contributions. The former provides guarantees for the classification task and performs conformal corrections for localization conditioned on the predicted object class. The latter designs comprehensive prediction sets that encompass confidence scores, classification, and localization, specifically tailored for Faster R-CNN models. However, none of these methods can achieve fine-grained instance-level conditional coverage. In high-stakes applications—particularly personnel search and rescue, autonomous driving, and scenarios involving small objects or diverse environmental conditions—more robust safety guarantees are imperative.

A.1.2 Conditional Conformal Prediction

Achieving the desired coverage level for prediction sets across all possible covariate values is known to be theoretically impossible without distributional assumptions (Vovk, 2012; Foygel Barber et al., 2021). As a result, recent scholarship has pivoted toward methods that adapt to local feature properties, aiming to better approximate conditional coverage.

Significant efforts have been directed at enhancing conditional validity, either by refining the calibration procedure (Lei & Wasserman, 2014; Guan, 2023; Hore & Barber, 2023; Barber et al., 2023) or by modifying initial prediction rules (Romano et al., 2019; Sesia & Romano, 2021; Chernozhukov et al., 2021; Luo & Zhou, 2025b). Furthermore, attention has been given to coverage under covariate shift (Lei & Wasserman, 2014; Tibshirani et al., 2019; Izbicki et al., 2022; Gao et al., 2025); notably, Gibbs et al. (2025) introduced a framework designed to achieve exact finite-sample coverage under shift conditions.

Group conditional guarantees have also been explored (Toccaceli & Gammerman, 2019; Gupta et al., 2020; Ding et al., 2023; Cherian et al., 2024; Li & Huai, 2025). Mondrian conformal prediction (Vovk et al., 2003) provides exact coverage for disjoint groups, while Kandinsky conformal prediction extends this (Brunekreef et al., 2024; Bairaktari et al., 2025). More flexible methods for overlapping groups also exist (Foygel Barber et al., 2021; Jung et al., 2023), though they can be computationally intensive or rely on distributional assumptions. Other methods improve conditional coverage by learning representations or risk adaptation, such as density-based atypicality (Yuksekgonul et al., 2023), learning partitions of the covariate space (Kiyani et al., 2024), automatically adaptive risk control (Blot et al., 2024; 2025), and conditional risk adaptation (Luo & Zhou, 2025a).

Our method makes a significant contribution to this field by proposing a novel instance-level conditional coverage framework, advancing research on coverage mechanisms in object detection. It moves beyond traditional group-based processing approaches, adopting a new mechanism that adaptively adjusts the expansion scale for each predicted instance. This is achieved by training the end-to-end predictive model CABE based on a differentiable loss function architecture, which explicitly optimizes and reduces the instance-level coverage gap while strictly adhering to joint risk constraints.

A.2 Theoretical Analysis

A.2.1 Proof of Theorem 3.1

Proof. For each sample (X_i, Y_i), by definition, the image-level classification FNR $L_i^{cls}(\lambda) = \ell_{\text{cls}}(X_i, Y_i; \lambda) \in [0,1]$ is *non-decreasing* in $\lambda \in [0,1]$ (so *non-increasing* in $1-\lambda$), and spatial FNR $L_i^{box}(t) = \ell_{box}(X_i, Y_i; \hat{\boldsymbol{\delta}}_i(t)) \in [0,1]$ is *non-increasing* in $t \in [0, t_{\max}]$, where $t_{\max} := \max(H, W)$ is the maximum pixel offset. Both $L_i^{cls}(\lambda)$ and $L_i^{box}(t)$ are right-continuous and satisfy

$$L_i^{cls}(0) \le \alpha_1, \quad L_i^{box}(t_{\max}) \le \alpha_2. \tag{19}$$

By Theorem 1 in Angelopoulos et al. (2024), we have

$$\mathbb{E}\left[L_{n+1}^{cls}(\hat{\lambda})\right] \le \alpha_1, \tag{20}$$

$$\mathbb{E}\left[L_{n+1}^{box}(\hat{t})\right] \le \alpha_2. \tag{21}$$

For a matched object $j \in \mathcal{M}(X_{n+1})$, define:

$$A_j := \mathbb{I}\left(\hat{\pi}_{\sigma(j),k_j} \ge \hat{\lambda}\right), \quad C_j := \mathbb{I}\left(B_j \subseteq \mathcal{B}_{\hat{\boldsymbol{\delta}}(\hat{t})}(\hat{B}_{\sigma(j)})\right), \tag{22}$$

where $\hat{\boldsymbol{\delta}}_{n+1}(\hat{t}) = \hat{\boldsymbol{\delta}}_{n+1} + \hat{t}\mathbf{1}_d$. Then

$$\mathbb{I}\left(\neg(A_j \wedge C_j)\right) \le \mathbb{I}(\neg A_j) + \mathbb{I}(\neg C_j). \tag{23}$$

Averaging over $j \in \mathcal{M}(X_{n+1})$ yields the *image-level* inequality

$$\mathcal{R}(X_{n+1}, Y_{n+1}; \hat{\lambda}, \hat{\boldsymbol{\delta}}_{n+1}(\hat{t})) \le L_{n+1}^{cls}(\hat{\lambda}) + L_{n+1}^{box}(\hat{t}). \tag{24}$$

Taking expectations we have:

$$\mathbb{E}\left[\mathcal{R}(X_{n+1}, Y_{n+1}; \hat{\lambda}, \hat{\boldsymbol{\delta}}_{n+1}(\hat{t}))\right] \le \mathbb{E}\left[L_{n+1}^{cls}(\hat{\lambda})\right] + \mathbb{E}\left[L_{n+1}^{box}(\hat{t})\right] \le \alpha_1 + \alpha_2. \tag{25}$$

□

A.2.2 A REMARK ON IMAGE-CONDITIONAL BOX RISK CONTROL

Achieving distribution-free *image-conditional* control is known to be impossible in general without additional assumptions. Nonetheless, CABE is designed to *reduce heterogeneity across images* by learning an image-adaptive base expansion $\hat{\boldsymbol{\delta}}(\cdot)$, after which a global CRC offset calibration $\hat{t} \geq 0$ is applied to ensure marginal risk control. To clarify the relationship between the learned image-adaptive expansion and the subsequent CRC offset, we record a simple sufficient condition under which the resulting (offset-adjusted) rule satisfies image-conditional control of the localization (box) risk.

Let $(X, Y) \sim P$ denote a test pair drawn from an unknown distribution on $\mathcal{X} \times \mathcal{Y}$, and let P_X be the marginal distribution of X. Fix the detector $\hat{f}$ and the matching protocol $(\mathcal{M}, \sigma)$. For an expansion parameter $\boldsymbol{\delta} \in \mathbb{R}^d_+$, let $\mathcal{B}_{\boldsymbol{\delta}}(\hat{B})$ denote the expanded box (possibly anisotropic). By construction, the expansion operator is nested coordinatewise: for all predicted boxes $\hat{B}$,

$$\boldsymbol{\delta} \preceq \boldsymbol{\delta}' \quad \Longrightarrow \quad \mathcal{B}_{\boldsymbol{\delta}}(\hat{B}) \subseteq \mathcal{B}_{\boldsymbol{\delta}'}(\hat{B}), \tag{26}$$

where $\preceq$ denotes the componentwise order on $\mathbb{R}^d$. Consequently, for all (x, y) and all $\boldsymbol{\delta}, \boldsymbol{\delta}' \in \mathbb{R}^d_+$, if $\boldsymbol{\delta} \preceq \boldsymbol{\delta}'$, then enlarging the boxes cannot increase the localization loss, i.e.,

$$\ell_{box}(x, y; \boldsymbol{\delta}') \leq \ell_{box}(x, y; \boldsymbol{\delta}). \tag{27}$$

Define the *image-conditional box risk* at image x as

$$r_{box}(x, \boldsymbol{\delta}) := \mathbb{E}\left[\ell_{box}(X, Y; \boldsymbol{\delta}) \mid X = x\right]. \tag{28}$$

By the monotonicity property above, for P_X-almost every x, the map $\boldsymbol{\delta} \mapsto r_{box}(x, \boldsymbol{\delta})$ is non-increasing in each coordinate: if $\boldsymbol{\delta} \preceq \boldsymbol{\delta}'$, then $r_{box}(x, \boldsymbol{\delta}') \leq r_{box}(x, \boldsymbol{\delta})$.

Assumption A.1 (Near-conditional safety of the learned expansion)**.** *Fix $\alpha_2 \in (0, 1)$. Let $\hat{\boldsymbol{\delta}} : \mathcal{X} \to \mathbb{R}^d_+$ be the learned (image-adaptive) expansion rule. Assume that for P_X-almost every x,*

$$r_{box}\big(x, \hat{\boldsymbol{\delta}}(x)\big) \leq \alpha_2. \tag{29}$$

Proposition A.2 (Image-conditional box-risk control)**.** *Under Assumption A.1, for any global correction parameter $\hat{t} \geq 0$ and the adjusted expansion $\hat{\boldsymbol{\delta}}_{\hat{t}}(x) = \hat{\boldsymbol{\delta}}(x) + \hat{t}\,\mathbf{1}_d$, we have for P_X-almost every x,*

$$r_{box}\big(x, \hat{\boldsymbol{\delta}}_{\hat{t}}(x)\big) \leq \alpha_2. \tag{30}$$

Proof. Fix x in the full-measure set where Assumption A.1 holds. Since $\hat{t} \geq 0$, we have $\hat{\boldsymbol{\delta}}(x) \preceq \hat{\boldsymbol{\delta}}_{\hat{t}}(x)$. By the componentwise monotonicity of $\boldsymbol{\delta} \mapsto r_{box}(x, \boldsymbol{\delta})$,

$$r_{box}\big(x, \hat{\boldsymbol{\delta}}_{\hat{t}}(x)\big) \leq r_{box}\big(x, \hat{\boldsymbol{\delta}}(x)\big) \leq \alpha_2. \tag{31}$$

□

Remark 3 *Assumption A.1 is strong and not directly verifiable from finite calibration data because it concerns the unobservable conditional risk $r_{box}(x, \cdot)$. We do* not *claim distribution-free image-conditional risk control for CABE. Instead, CABE empirically reduces the variability of image-level risk, and Theorem 3.1 provides a rigorous marginal guarantee. Proposition A.2 simply shows that if the learned base expansion is already close to conditionally safe, then the final CRC global calibration step (with $\hat{t} \geq 0$) cannot worsen conditional box risk.*

A.3 METRICS

1. Coverage. We evaluate the spatial coverage rate by averaging the proportion of covered matched ground-truth objects across all test images. A ground-truth object B_j is considered covered if it is strictly contained within its corresponding expanded prediction $\mathcal{B}_\delta(\hat{B}_{\sigma(j)})$:

$$\text{Coverage} = \frac{1}{M} \sum_{X_i \in \mathcal{D}_{\text{test}}} \left(\frac{1}{|\mathcal{M}(X_i)|} \sum_{j \in \mathcal{M}(X_i)} \mathbb{I}\left(B_j \subseteq \mathcal{B}_\delta(\hat{B}_{\sigma(j)})\right) \right). \tag{32}$$

The theoretical target value for this metric is $1 - \alpha_2$.

2. Coverage Gap.(Kaur et al., 2025) To assess the stability of risk control across individual image samples, we define the coverage gap as the average absolute deviation of each sample's actual coverage from the target coverage.:

$$\text{Coverage Gap} = \frac{1}{M} \sum_{X_i \in \mathcal{D}_{\text{test}}} \left| \left(\frac{1}{|\mathcal{M}(X_i)|} \sum_{j \in \mathcal{M}(X_i)} \mathbb{I}\left(B_j \subseteq \mathcal{B}_\delta(\hat{B}_{\sigma(j)}) \right) \right) - (1 - \alpha_2) \right|. \quad (33)$$

3. Size. To quantify the cost of safety, we measure the relative increase in bounding box area. This is calculated as the macro-average of the expansion ratios over matched objects:

$$\text{Size} = \frac{1}{M} \sum_{X_i \in \mathcal{D}_{\text{test}}} \left(\frac{1}{|\mathcal{M}(X_i)|} \sum_{j \in \mathcal{M}(X_i)} \frac{\text{Area}(\mathcal{B}_\delta(\hat{B}_{\sigma(j)}))}{\text{Area}(\hat{B}_{\sigma(j)})} \right). \quad (34)$$

4. Mean IoU. To ensure that boundary expansion does not excessively degrade localization precision, we compute the IoU as a global (micro) average across all matched objects in the test set:

$$\text{mIoU} = \frac{\sum_{X_i \in \mathcal{D}_{\text{test}}} \sum_{j \in \mathcal{M}(X_i)} \text{IoU}(B_j, \mathcal{B}_\delta(\hat{B}_{\sigma(j)}))}{\sum_{X_i \in \mathcal{D}_{\text{test}}} |\mathcal{M}(X_i)|}. \quad (35)$$

A.4 Datasets and Partitioning Protocol

For each dataset, we adhered to a unified partitioning protocol. First, the entire dataset was randomly split into a training set (85%) and a test set (15%), with the test set strictly reserved for the final performance evaluation. The training set was then further divided: 60% of it was utilized exclusively to train the base object detection models, while the remaining 40% was reserved for uncertainty quantification. For the standard CRC method, the entire aforementioned 40% of the data served as the calibration set. In contrast, for the CABE method, the aforementioned 40% of the data was further subdivided: 70% of these data were used to train the boundary expansion prediction model f_θ, and the remaining 30% served as the calibration set for the final parameter adjustment of the prediction model.

The specific safety-critical datasets evaluated in this study are detailed below:

- **Rescue Object Detection Dataset:** Sourced from the repository maintained by Meine Meine (2023), this dataset is designed to simulate disaster response scenarios via aerial imagery. It contains annotations for three critical classes: Person, Car, and Fire.
- **AFO (Aerial Floating Objects) Dataset:** We utilized the RescueFObj variation Ugorji (2024) of the AFO dataset. This dataset has also been previously investigated in the context of conformal prediction for aerial imagery Copley et al. (2024). It focuses on maritime objects and includes classes such as Human, Wind/Sup-board, Boat, Buoy, Sailboat, and Kayak. The primary challenge lies in distinguishing small floating targets from wave textures and sun glint.
- **DroneVehicle Dataset:** We employed the DroneVehicle dataset, originally proposed by Sun et al. Sun et al. (2022), to assess object detection performance in traffic surveillance scenarios. This dataset includes a high volume of images captured under varying illumination conditions (day and night). While the original dataset supports oriented bounding boxes (OBB), we focus on the risk control of their spatial coverage for small-vehicle and large-vehicle categories using horizontal bounding boxes.

A.5 Experimental Details

Reproducibility Statement: To ensure the transparency and reproducibility of our results, the complete source code and trained model weights will be made publicly available upon acceptance.

Implementation Details: All experiments were conducted on a server equipped with an NVIDIA A100 GPU, implemented using PyTorch 2.3.0. The CABE network was trained for 60 epochs to

ensure convergence, utilizing a fixed learning rate of 1×10^{-5} and a batch size of 32. The base object detection models (YOLOv8, YOLOv10) were trained following standard protocols with a learning rate of 1×10^{-3} and a batch size of 128. To guarantee deterministic results, a fixed random seed of 42 was applied to all experiments.

A.6 Rescue Results

Due to space constraints in the main text, we provide additional qualitative analysis of the Rescue dataset in this appendix. As illustrated in Figure 6, $\mathrm{CRC}_{\mathrm{add}}$ often fails to sufficiently cover ground truth objects (e.g., missing parts of fire or persons), leading to a higher False Negative Rate (FNR). In contrast, $\mathrm{CABE}_{\mathrm{add}}$ leverages image-adaptive boundary optimization to predict image-specific expansion vectors. This allows $\mathrm{CABE}_{\mathrm{add}}$ to consistently maintain the target coverage and achieve a significantly lower and more stable FNR, which is paramount for safety-critical rescue applications.

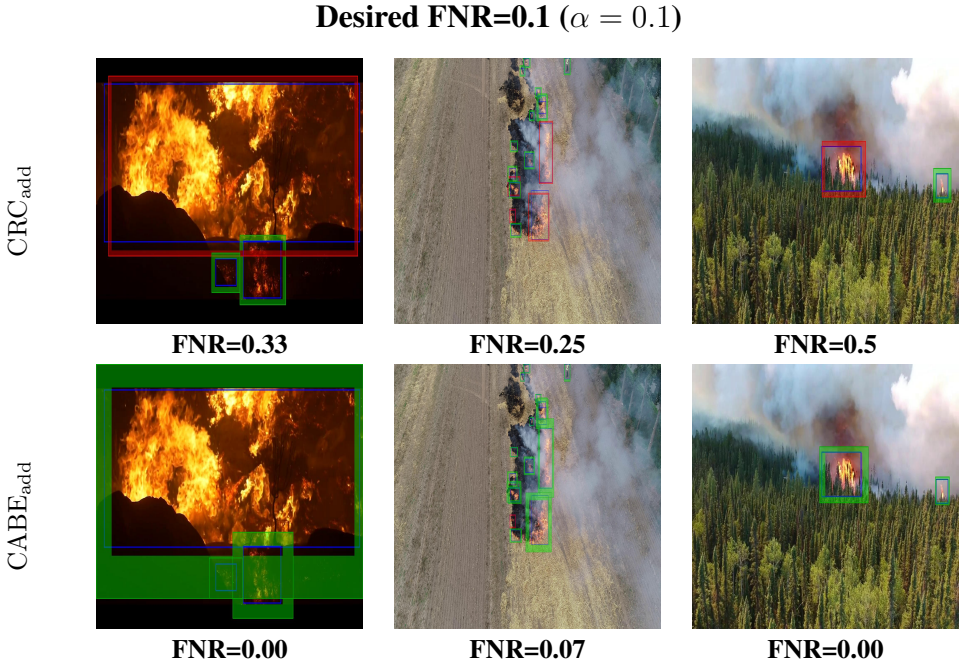


Figure 6: Qualitative comparison of $\mathrm{CRC}_{\mathrm{add}}$ (top) and $\mathrm{CABE}_{\mathrm{add}}$ (bottom) on the Rescue dataset at $\alpha = 0.1$.

A.7 Comparison with Stratified CRC Baselines

A common approach to improving conditional coverage in traditional CRC is to partition the calibration and test data into discrete groups (bins) based on a heuristic score, and then apply CRC independently within each group. To rigorously evaluate the effectiveness of our proposed continuous adaptive boundary optimization, we established two strong Stratified CRC baselines. Specifically, we stratified the data based on two distinct heuristics: bounding box pixel density ($\mathrm{CRC}_{\mathrm{Density}}$) and predictive entropy ($\mathrm{CRC}_{\mathrm{Entropy}}$).

As shown in Table 2, while the stratified CRC methods ($\mathrm{CRC}_{\mathrm{Density}}$ and $\mathrm{CRC}_{\mathrm{Entropy}}$) achieve marginal improvements in the Coverage Gap compared to the standard global CRC baseline, they still fall significantly short of the performance achieved by CABE. Across all three datasets, the stratified baselines exhibit a relative degradation (increase) in the Coverage Gap ranging from 9.4% to 12.1% when compared to CABE.

These results underscore a fundamental limitation of discrete stratification: binning relies on manually selected heuristics and discrete grouping, which inherently lacks the flexibility to capture the fine-grained, continuous instance-level variations present in complex safety-critical scenes. In contrast,

CABE directly learns an image-adaptive, end-to-end differentiable mapping that effectively minimizes instance-level coverage variance, resulting in a substantially tighter Coverage Gap and more reliable conditional safety guarantees.

Table 2: Comparison of $\mathrm{CABE}_{\mathrm{mul}}$ with Global and Spatial Density and Entropy Stratified $\mathrm{CRC}_{\mathrm{mul}}$ Baselines across three datasets (AFO, DroneVehicle, and Rescue) based on the YOLOv8 model at $\alpha = 0.1$. The values are presented as mean (standard deviation) across 20 independent runs. The percentages in parentheses with upward arrows within the Coverage Gap column indicate the relative increase (degradation) in coverage gap compared to $\mathrm{CABE}_{\mathrm{mul}}$.

Dataset	Method	Joint Coverage	Coverage	Coverage Gap	Size	IoU
AFO	$\mathrm{CRC}_{\mathrm{mul}}$	0.900 (0.012)	0.906 (0.012)	0.119 (0.008) (↑ 11.2%)	1.942 (0.017)	0.535 (0.005)
	$\mathrm{CRC}_{\mathrm{Density}}$	0.898 (0.009)	0.905 (0.010)	0.117 (0.007) (↑ 9.4%)	1.943 (0.019)	0.543 (0.003)
	$\mathrm{CRC}_{\mathrm{Entropy}}$	0.902 (0.012)	0.908 (0.012)	0.117 (0.008) (↑ 9.4%)	1.957 (0.015)	0.528 (0.005)
	$\mathrm{CABE}_{\mathrm{mul}}$	0.903 (0.009)	0.909 (0.010)	**0.107 (0.007)**	1.959 (0.040)	0.571 (0.007)
DroneVehicle	$\mathrm{CRC}_{\mathrm{mul}}$	0.896 (0.004)	0.902 (0.004)	0.112 (0.002) (↑ 13.1%)	1.621 (0.003)	0.625 (0.001)
	$\mathrm{CRC}_{\mathrm{Density}}$	0.896 (0.003)	0.902 (0.004)	0.111 (0.002) (↑ 12.1%)	1.622 (0.003)	0.631 (0.001)
	$\mathrm{CRC}_{\mathrm{Entropy}}$	0.896 (0.004)	0.902 (0.004)	0.110 (0.002) (↑ 11.1%)	1.612 (0.004)	0.629 (0.002)
	$\mathrm{CABE}_{\mathrm{mul}}$	0.894 (0.004)	0.900 (0.004)	**0.099 (0.002)**	1.612 (0.011)	0.650 (0.004)
Rescue	$\mathrm{CRC}_{\mathrm{mul}}$	0.898 (0.004)	0.900 (0.004)	0.128 (0.003) (↑ 11.3%)	1.567 (0.004)	0.657 (0.002)
	$\mathrm{CRC}_{\mathrm{Density}}$	0.899 (0.003)	0.900 (0.003)	0.126 (0.002) (↑ 9.6%)	1.565 (0.003)	0.654 (0.001)
	$\mathrm{CRC}_{\mathrm{Entropy}}$	0.899 (0.004)	0.900 (0.004)	0.127 (0.003) (↑ 10.4%)	1.566 (0.004)	0.648 (0.002)
	$\mathrm{CABE}_{\mathrm{mul}}$	0.907 (0.006)	0.910 (0.006)	**0.115 (0.003)**	1.575 (0.029)	0.642 (0.010)

A.8 Intersection Over Union(IoU)

To satisfy risk constraints, boundary expansion may cause damage to localization precision, typically manifesting as a decrease in IoU. To evaluate this critical trade-off, Figures 7, 8, and 9 illustrate the IoU distributions of different methods. The results indicate that, while strictly maintaining the target coverage, our proposed $\mathrm{CABE}_{\mathrm{add/mul}}$ method achieves consistently higher IoU than $\mathrm{CRC}_{\mathrm{add/mul}}$ in most cases. This demonstrates that $\mathrm{CABE}_{\mathrm{add/mul}}$ effectively mitigates the over-conservatism observed in $\mathrm{CRC}_{\mathrm{add/mul}}$ by generating tighter, instance-adaptive prediction sets.

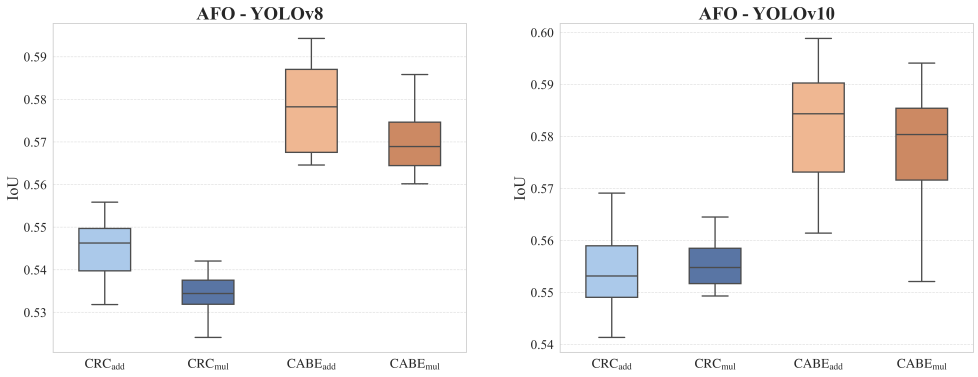


Figure 7: IoU distributions of $\mathrm{CRC}_{\mathrm{add}}$,$\mathrm{CRC}_{\mathrm{mul}}$, $\mathrm{CABE}_{\mathrm{add}}$ and $\mathrm{CABE}_{\mathrm{mul}}$ methods on the AFO dataset using different base detection models with risk level is set to $\alpha = 0.1$.

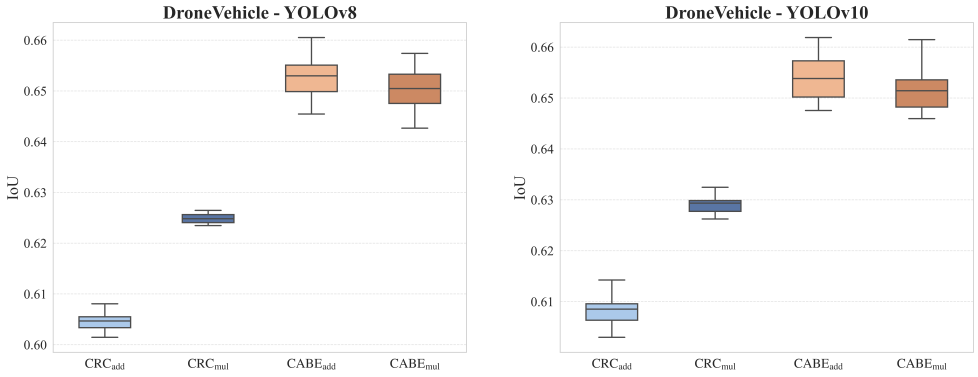


Figure 8: IoU distributions of $\mathrm{CRC}_{\mathrm{add}}$,$\mathrm{CRC}_{\mathrm{mul}}$, $\mathrm{CABE}_{\mathrm{add}}$ and $\mathrm{CABE}_{\mathrm{mul}}$ methods on the DroneVehicle dataset using different base detection models with risk level is set to $\alpha = 0.1$.

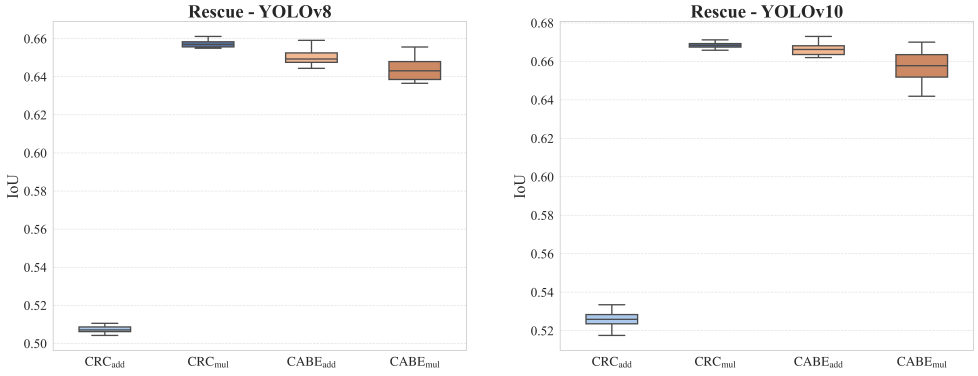


Figure 9: IoU distributions of $\mathrm{CRC}_{\mathrm{add}}$,$\mathrm{CRC}_{\mathrm{mul}}$, $\mathrm{CABE}_{\mathrm{add}}$ and $\mathrm{CABE}_{\mathrm{mul}}$ methods on the Rescue dataset using different base detection models with risk level is set to $\alpha = 0.1$.

A.9 Statistical Significance Testing

To rigorously validate the performance improvements observed in the main experiments (Table 1), we conducted paired t-tests to evaluate the statistical significance of the differences between our proposed CABE method and the CRC baseline.

As detailed in Table 3, we analyzed the Coverage Gap, Size, and Mean IoU metrics across the AFO, DroneVehicle, and Rescue datasets for both additive (*add*) and multiplicative (*mul*) expansion schemes. The statistical analysis confirms that the reductions in the Coverage Gap achieved by

CABE are highly significant ($p < 0.05$) in all evaluated settings. In addition, the CABE framework demonstrates statistically significant improvements in bounding box efficiency (Size) and localization precision (Mean IoU) across the majority of configurations. These results underscore the robustness and reliability of the adaptive boundary optimization framework, proving that it can deliver tighter instance-level safety guarantees without arbitrarily degrading detection quality.

Table 3: Statistical significance testing (Paired t-test) comparing the performance of the CABE method against the CRC baseline corresponding to the results in Table 1. The analysis evaluates the Coverage Gap, Size, and Mean IoU across three datasets and two expansion methods (add/mul).

Dataset	Model	Method	Metric	t-statistic	p-value	Significant ($p < 0.05$)
AFO	YOLOv8	add	Coverage Gap	-4.481	2.56×10^{-4}	✓
			Size	-12.189	1.99×10^{-10}	✓
			IoU	11.691	4.02×10^{-10}	✓
		mul	Coverage Gap	-8.809	3.89×10^{-8}	✓
			Size	1.746	0.097	×
			IoU	16.582	9.32×10^{-13}	✓
	YOLOv10	add	Coverage Gap	-6.219	5.66×10^{-6}	✓
			Size	-15.573	2.84×10^{-12}	✓
			IoU	10.146	4.16×10^{-9}	✓
		mul	Coverage Gap	-5.738	1.57×10^{-5}	✓
			Size	0.779	0.446	×
			IoU	8.434	7.58×10^{-8}	✓
DroneVehicle	YOLOv8	add	Coverage Gap	-29.648	2.25×10^{-17}	✓
			Size	-28.761	3.97×10^{-17}	✓
			IoU	57.357	9.32×10^{-23}	✓
		mul	Coverage Gap	-30.817	1.10×10^{-17}	✓
			Size	-3.344	0.003	✓
			IoU	27.776	7.58×10^{-17}	✓
	YOLOv10	add	Coverage Gap	-35.004	1.01×10^{-18}	✓
			Size	-26.002	2.58×10^{-16}	✓
			IoU	41.143	4.89×10^{-20}	✓
		mul	Coverage Gap	-39.922	8.62×10^{-20}	✓
			Size	-5.235	4.71×10^{-5}	✓
			IoU	23.673	1.46×10^{-15}	✓
Rescue	YOLOv8	add	Coverage Gap	-32.107	5.10×10^{-18}	✓
			Size	-130.514	1.60×10^{-29}	✓
			IoU	119.296	8.79×10^{-29}	✓
		mul	Coverage Gap	-32.107	5.10×10^{-18}	✓
			Size	17.410	3.90×10^{-13}	✓
			IoU	119.296	8.79×10^{-29}	✓
	YOLOv10	add	Coverage Gap	-29.292	2.82×10^{-17}	✓
			Size	-118.578	9.85×10^{-29}	✓
			IoU	212.688	1.50×10^{-33}	✓
		mul	Coverage Gap	-14.308	1.26×10^{-11}	✓
			Size	3.766	0.001	✓
			IoU	-6.213	5.73×10^{-6}	✓

A.10 Quantitative Analysis

To rigorously evaluate the reliability and efficiency of the proposed method in safety-critical scenarios, we extend our experiments to quantitative settings with varying FNR targets, specifically $\alpha \in \{0.05, 0.15, 0.20\}$. As presented in Table 4, a comprehensive comparison of joint coverage, bounding-box coverage, coverage gap, predicted set size, and mean IoU is conducted on three datasets using YOLOv8 and YOLOv10 as base models. The results show that, while all methods achieve satisfactory proximity to their respective target coverages, the $\mathrm{CABE}_{\mathrm{add}}$ framework demonstrates notably better efficiency and robustness compared to the baseline $\mathrm{CRC}_{\mathrm{add}}$ method. Specifically, without compromising coverage, $\mathrm{CABE}_{\mathrm{add}}$ consistently produces more compact prediction sets. Furthermore, it maintains a lower coverage gap throughout, indicating that its actual coverage across different samples stays closer to the target value. This effectively reduces conservatism and enhances the credibility of the method for high-risk applications.

Table 4: Sensitivity analysis of the proposed $\mathrm{CABE}_{\mathrm{add}}$ and the baseline $\mathrm{CRC}_{\mathrm{add}}$ on the AFO, DroneVehicle, and Rescue datasets across varying target risk levels ($\alpha \in \{0.05, 0.15, 0.20\}$) using both YOLOv8 and YOLOv10 architectures. The reported values represent the mean (standard deviation) evaluated over 20 independent runs. The percentages in parentheses with downward arrows indicate the relative reduction (improvement) in the condition coverage gap achieved by CABE compared to the CRC baseline.

Dataset	Model	Target Risk (α)	Method	Dataset Metrics				
				Joint Coverage	Coverage	Coverage Gap	Size	IoU
AFO	YOLOv8	$\alpha = 0.05$	$\mathrm{CRC}_{\mathrm{add}}$	0.943 (0.008)	0.950 (0.008)	0.071 (0.006) (–)	2.375 (0.037)	0.480 (0.006)
			$\mathrm{CABE}_{\mathrm{add}}$	0.946 (0.006)	0.953 (0.007)	**0.064** (0.004) (↓ 9.9%)	**2.199** (0.050)	**0.503** (0.012)
		$\alpha = 0.15$	$\mathrm{CRC}_{\mathrm{add}}$	0.849 (0.013)	0.855 (0.014)	0.163 (0.008) (–)	1.829 (0.030)	0.595 (0.007)
			$\mathrm{CABE}_{\mathrm{add}}$	0.842 (0.015)	0.849 (0.016)	**0.152** (0.008) (↓ 7.1%)	**1.796** (0.040)	**0.629** (0.009)
		$\alpha = 0.20$	$\mathrm{CRC}_{\mathrm{add}}$	0.801 (0.013)	0.807 (0.015)	0.195 (0.007) (–)	1.700 (0.025)	0.629 (0.007)
			$\mathrm{CABE}_{\mathrm{add}}$	0.793 (0.012)	0.799 (0.014)	**0.178** (0.010) (↓ 8.9%)	**1.710** (0.028)	**0.660** (0.015)
	YOLOv10	$\alpha = 0.05$	$\mathrm{CRC}_{\mathrm{add}}$	0.941 (0.010)	0.949 (0.009)	0.072 (0.007) (–)	2.230 (0.050)	0.488 (0.009)
			$\mathrm{CABE}_{\mathrm{add}}$	0.945 (0.008)	0.952 (0.006)	**0.064** (0.004) (↓ 11.1%)	**2.032** (0.035)	**0.512** (0.011)
		$\alpha = 0.15$	$\mathrm{CRC}_{\mathrm{add}}$	0.846 (0.011)	0.851 (0.012)	0.164 (0.008) (–)	1.712 (0.026)	0.608 (0.008)
			$\mathrm{CABE}_{\mathrm{add}}$	0.851 (0.017)	0.857 (0.016)	**0.146** (0.007) (↓ 11.0%)	**1.671** (0.023)	**0.627** (0.013)
		$\alpha = 0.20$	$\mathrm{CRC}_{\mathrm{add}}$	0.798 (0.012)	0.803 (0.013)	0.195 (0.008) (–)	1.587 (0.019)	0.645 (0.006)
			$\mathrm{CABE}_{\mathrm{add}}$	0.802 (0.019)	0.807 (0.018)	**0.173** (0.011) (↓ 11.3%)	**1.582** (0.030)	**0.659** (0.009)
DroneVehicle	YOLOv8	$\alpha = 0.05$	$\mathrm{CRC}_{\mathrm{add}}$	0.945 (0.003)	0.951 (0.002)	0.071 (0.002) (–)	1.963 (0.009)	0.522 (0.002)
			$\mathrm{CABE}_{\mathrm{add}}$	0.943 (0.003)	0.949 (0.003)	**0.064** (0.002) (↓ 9.9%)	**1.773** (0.011)	**0.590** (0.004)
		$\alpha = 0.15$	$\mathrm{CRC}_{\mathrm{add}}$	0.845 (0.004)	0.851 (0.005)	0.153 (0.002) (–)	1.534 (0.004)	0.655 (0.001)
			$\mathrm{CABE}_{\mathrm{add}}$	0.844 (0.004)	0.850 (0.004)	**0.126** (0.002) (↓ 17.5%)	**1.500** (0.004)	**0.690** (0.003)
		$\alpha = 0.20$	$\mathrm{CRC}_{\mathrm{add}}$	0.796 (0.005)	0.801 (0.005)	0.181 (0.002) (–)	1.443 (0.003)	0.692 (0.001)
			$\mathrm{CABE}_{\mathrm{add}}$	0.793 (0.006)	0.799 (0.006)	**0.148** (0.003) (↓ 18.3%)	**1.427** (0.008)	**0.719** (0.003)
	YOLOv10	$\alpha = 0.05$	$\mathrm{CRC}_{\mathrm{add}}$	0.945 (0.003)	0.950 (0.004)	0.071 (0.002) (–)	1.954 (0.012)	0.526 (0.004)
			$\mathrm{CABE}_{\mathrm{add}}$	0.944 (0.003)	0.949 (0.003)	**0.063** (0.002) (↓ 11.3%)	**1.768** (0.015)	**0.592** (0.004)
		$\alpha = 0.15$	$\mathrm{CRC}_{\mathrm{add}}$	0.846 (0.006)	0.850 (0.006)	0.152 (0.003) (–)	1.531 (0.006)	0.659 (0.003)
			$\mathrm{CABE}_{\mathrm{add}}$	0.845 (0.005)	0.850 (0.006)	**0.125** (0.004) (↓ 17.8%)	**1.495** (0.007)	**0.694** (0.003)
		$\alpha = 0.20$	$\mathrm{CRC}_{\mathrm{add}}$	0.796 (0.007)	0.801 (0.007)	0.181 (0.003) (–)	1.443 (0.006)	0.695 (0.003)
			$\mathrm{CABE}_{\mathrm{add}}$	0.797 (0.005)	0.801 (0.005)	**0.147** (0.003) (↓ 18.8%)	**1.427** (0.007)	**0.721** (0.003)
Rescue	YOLOv8	$\alpha = 0.05$	$\mathrm{CRC}_{\mathrm{add}}$	0.949 (0.002)	0.951 (0.002)	0.084 (0.002) (–)	3.764 (0.062)	0.364 (0.004)
			$\mathrm{CABE}_{\mathrm{add}}$	0.949 (0.002)	0.950 (0.002)	**0.073** (0.002) (↓ 13.1%)	**1.754** (0.014)	**0.572** (0.005)
		$\alpha = 0.15$	$\mathrm{CRC}_{\mathrm{add}}$	0.849 (0.003)	0.851 (0.004)	0.181 (0.002) (–)	1.842 (0.011)	0.597 (0.003)
			$\mathrm{CABE}_{\mathrm{add}}$	0.850 (0.004)	0.852 (0.004)	**0.153** (0.003) (↓ 15.6%)	**1.454** (0.009)	**0.696** (0.002)
		$\alpha = 0.20$	$\mathrm{CRC}_{\mathrm{add}}$	0.799 (0.005)	0.801 (0.005)	0.215 (0.003) (–)	1.623 (0.008)	0.654 (0.003)
			$\mathrm{CABE}_{\mathrm{add}}$	0.800 (0.005)	0.802 (0.005)	**0.179** (0.003) (↓ 16.6%)	**1.370** (0.006)	**0.726** (0.003)
	YOLOv10	$\alpha = 0.05$	$\mathrm{CRC}_{\mathrm{add}}$	0.948 (0.003)	0.950 (0.003)	0.084 (0.002) (–)	3.492 (0.071)	0.370 (0.005)
			$\mathrm{CABE}_{\mathrm{add}}$	0.947 (0.003)	0.949 (0.003)	**0.074** (0.002) (↓ 11.9%)	**1.673** (0.016)	**0.585** (0.007)
		$\alpha = 0.15$	$\mathrm{CRC}_{\mathrm{add}}$	0.848 (0.007)	0.849 (0.007)	0.181 (0.004) (–)	1.717 (0.019)	0.608 (0.014)
			$\mathrm{CABE}_{\mathrm{add}}$	0.847 (0.005)	0.849 (0.006)	**0.154** (0.006) (↓ 14.9%)	**1.389** (0.008)	**0.712** (0.003)
		$\alpha = 0.20$	$\mathrm{CRC}_{\mathrm{add}}$	0.798 (0.007)	0.800 (0.007)	0.213 (0.003) (–)	1.520 (0.010)	0.667 (0.003)
			$\mathrm{CABE}_{\mathrm{add}}$	0.797 (0.005)	0.797 (0.005)	**0.181** (0.005) (↓ 15.0%)	**1.322** (0.010)	**0.740** (0.005)

Table 5: Quantitative performance comparison between CRC and CABE across different datasets, base models, and relaxed IoG thresholds (0.90 and 0.95) at a target risk level of $\alpha = 0.1$. The percentages with downward arrows indicate the relative reduction in the coverage gap achieved by CABE compared to the CRC baseline.

Dataset	Model	IOG	Method	Joint Coverage	Coverage	Coverage Gap	Size	Mean IoU
AFO	YOLOv8	0.9	CRC_{add}	0.900 (0.011)	0.907 (0.012)	0.119 (0.008)	**1.425 (0.008)**	0.696 (0.003)
			CABE_{add}	0.895 (0.011)	0.902 (0.012)	**0.113 (0.008) (↓ 5.0%)**	1.450 (0.026)	**0.732 (0.008)**
		0.95	CRC_{add}	0.900 (0.012)	0.906 (0.012)	0.119 (0.008)	1.652 (0.016)	0.618 (0.005)
			CABE_{add}	0.897 (0.012)	0.904 (0.013)	**0.111 (0.008) (↓ 6.9%)**	**1.649 (0.028)**	**0.667 (0.011)**
	YOLOv10	0.9	CRC_{add}	0.896 (0.011)	0.903 (0.011)	0.119 (0.008)	**1.318 (0.010)**	0.723 (0.004)
			CABE_{add}	0.897 (0.010)	0.904 (0.010)	**0.112 (0.006) (↓ 6.4%)**	1.345 (0.022)	**0.739 (0.008)**
		0.95	CRC_{add}	0.896 (0.011)	0.902 (0.010)	0.120 (0.007)	1.535 (0.010)	0.642 (0.004)
			CABE_{add}	0.896 (0.011)	0.903 (0.011)	**0.111 (0.006) (↓ 7.5%)**	**1.533 (0.027)**	**0.672 (0.012)**
Car	YOLOv8	0.9	CRC_{add}	0.896 (0.004)	0.901 (0.004)	0.114 (0.002)	**1.195 (0.002)**	**0.804 (0.001)**
			CABE_{add}	0.894 (0.005)	0.900 (0.005)	**0.105 (0.003) (↓ 8.6%)**	1.231 (0.012)	0.801 (0.005)
		0.95	CRC_{add}	0.896 (0.004)	0.902 (0.004)	0.113 (0.002)	**1.368 (0.003)**	0.727 (0.001)
			CABE_{add}	0.893 (0.004)	0.899 (0.004)	**0.100 (0.002) (↓ 11.0%)**	1.371 (0.010)	**0.750 (0.004)**
	YOLOv10	0.9	CRC_{add}	0.895 (0.003)	0.900 (0.004)	0.115 (0.002)	**1.192 (0.002)**	**0.807 (0.001)**
			CABE_{add}	0.894 (0.005)	0.900 (0.005)	**0.105 (0.003) (↓ 8.8%)**	1.225 (0.011)	0.802 (0.005)
		0.95	CRC_{add}	0.895 (0.003)	0.900 (0.004)	0.113 (0.002)	1.365 (0.002)	0.730 (0.001)
			CABE_{add}	0.894 (0.004)	0.900 (0.004)	**0.099 (0.003) (↓ 12.3%)**	**1.365 (0.010)**	**0.751 (0.004)**
Fire	YOLOv8	0.9	CRC_{add}	0.899 (0.004)	0.901 (0.004)	0.129 (0.003)	**1.176 (0.002)**	**0.819 (0.001)**
			CABE_{add}	0.899 (0.004)	0.901 (0.004)	**0.123 (0.003) (↓ 5.1%)**	1.196 (0.016)	0.797 (0.007)
		0.95	CRC_{add}	0.899 (0.004)	0.900 (0.004)	0.128 (0.002)	1.336 (0.003)	**0.753 (0.001)**
			CABE_{add}	0.899 (0.004)	0.900 (0.004)	**0.120 (0.002) (↓ 6.9%)**	**1.327 (0.012)**	0.746 (0.005)
	YOLOv10	0.9	CRC_{add}	0.898 (0.004)	0.899 (0.004)	0.127 (0.002)	**1.113 (0.002)**	**0.835 (0.001)**
			CABE_{add}	0.897 (0.003)	0.899 (0.003)	**0.124 (0.002) (↓ 2.9%)**	1.148 (0.011)	0.811 (0.004)
		0.95	CRC_{add}	0.897 (0.003)	0.899 (0.003)	0.127 (0.002)	**1.263 (0.003)**	**0.768 (0.001)**
			CABE_{add}	0.897 (0.003)	0.899 (0.003)	**0.120 (0.002) (↓ 5.2%)**	1.269 (0.010)	0.764 (0.003)

A.11 Performance under Varying IoG Targets

In previous sections, a strictly rigorous spatial coverage condition was enforced, requiring the ground truth object to be fully contained within the predicted box (i.e., an IoG requirement of 1.0). However, in many practical applications, a slightly relaxed coverage threshold (e.g., IoG ≥ 0.90 or 0.95) is sufficient and may provide a more favorable trade-off between safety guarantees and bounding box efficiency. In this section, we evaluate the performance of the CABE model against the CRC baseline when the target IoG is adjusted to 0.90 and 0.95.

As presented in Table 5, relaxing the IoG requirement allows both methods to generate more compact prediction sets (reduced Size) and achieve higher localization quality (Mean IoU) while strictly maintaining the target joint coverage across varying risk levels ($\alpha \in \{0.05, 0.1\}$). Despite this relaxed constraint, CABE consistently outperforms the CRC baseline. Most notably, CABE maintains a significantly lower Coverage Gap—with relative reductions ranging from 2.1% to 12.3%—demonstrating its superior ability to adaptively control instance-level risk. Furthermore, CABE generally yields tighter bounding boxes and better Mean IoU compared to CRC.

To rigorously validate these empirical observations, we conducted paired t-tests to evaluate the statistical significance of the performance differences between CABE and CRC, as detailed in Table 6and7. The statistical analysis confirms that the reductions in the Coverage Gap achieved by CABE are statistically significant ($p < 0.05$) across almost all evaluated datasets and IoG thresholds. Additionally, CABE's improvements in Mean IoU and bounding box Size are predominantly significant, underscoring the robustness of the adaptive boundary extension framework in optimizing localization quality while rigorously fulfilling varied spatial coverage guarantees.

A.12 Image-level Coverage Density Distributions

In this appendix, we provide a detailed visualization of the image-level coverage distributions across different datasets to further highlight the advantages of the CABE method. Figure 10 illustrates the density distribution of image-level coverage for the $CRC_{add/mul}$ and $CABE_{add/mul}$ methods using the YOLOv8 model on the AFO, DroneVehicle, and Rescue datasets. The density plots clearly

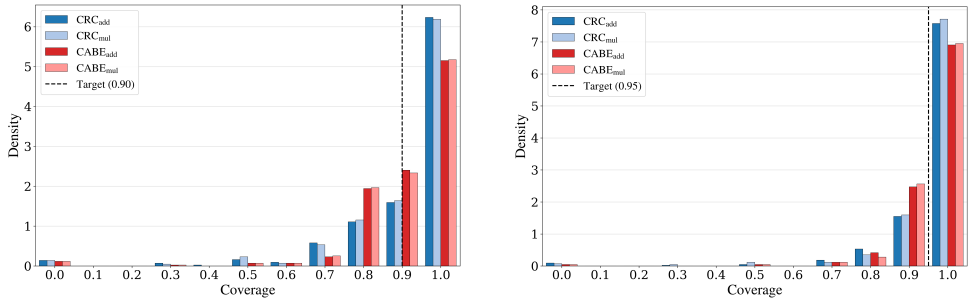
Table 6: Statistical significance testing (Paired t-test) comparing the performance of CABE against CRC across varying datasets and relaxed IoG targets for the YOLOv8 model at target risk level $\alpha = 0.1$.

Dataset	Model	IoG	Metric	t-statistic	p-value	Significant ($p < 0.05$)
AFO	YOLOv8	0.90	Coverage Gap	-3.044	6.68×10^{-3}	✓
			Size	4.231	4.52×10^{-4}	✓
			IoU	16.716	8.07×10^{-13}	✓
		0.95	Coverage Gap	-4.871	1.06×10^{-4}	✓
			Size	-0.379	7.09×10^{-1}	×
			IoU	17.165	5.02×10^{-13}	✓
DroneVehicle	YOLOv8	0.90	Coverage Gap	-13.125	5.62×10^{-11}	✓
			Size	12.752	9.21×10^{-11}	✓
			IoU	-2.533	2.03×10^{-2}	✓
		0.95	Coverage Gap	-25.647	3.32×10^{-16}	✓
			Size	1.412	1.74×10^{-1}	×
			IoU	25.076	5.04×10^{-16}	✓
Rescue	YOLOv8	0.90	Coverage Gap	-10.359	2.97×10^{-9}	✓
			Size	5.171	5.43×10^{-5}	✓
			IoU	-13.178	5.25×10^{-11}	✓
		0.95	Coverage Gap	-17.699	2.90×10^{-13}	✓
			Size	-3.133	5.47×10^{-3}	✓
			IoU	-5.904	1.10×10^{-5}	✓

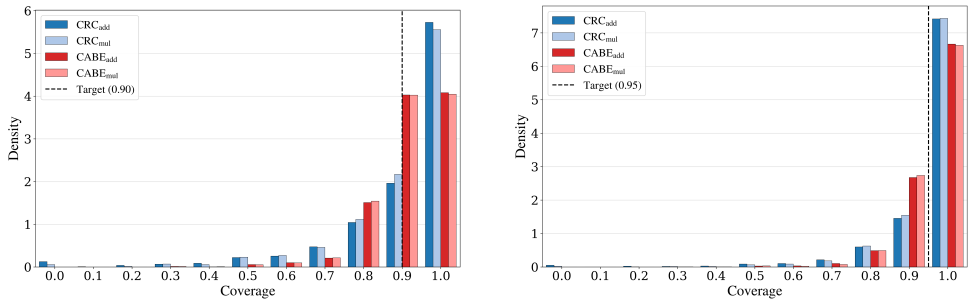
Table 7: Statistical significance testing (Paired t-test) comparing the performance of CABE against CRC across varying datasets and relaxed IoG targets for the YOLOv10 model at target risk level $\alpha = 0.1$.

Dataset	Model	IoG	Metric	t-statistic	p-value	Significant ($p < 0.05$)
AFO	YOLOv10	0.90	Coverage Gap	-3.761	0.001	✓
			Size	4.699	1.56×10^{-4}	✓
			IoU	7.698	2.95×10^{-7}	✓
		0.95	Coverage Gap	-4.504	2.43×10^{-4}	✓
			Size	-0.300	0.768	×
			IoU	10.272	3.41×10^{-9}	✓
DroneVehicle	YOLOv10	0.90	Coverage Gap	-15.769	2.28×10^{-12}	✓
			Size	12.675	1.02×10^{-10}	✓
			IoU	-4.923	9.45×10^{-5}	✓
		0.95	Coverage Gap	-27.374	9.94×10^{-17}	✓
			Size	-0.043	0.966	×
			IoU	20.462	2.10×10^{-14}	✓
Rescue	YOLOv10	0.90	Coverage Gap	-6.167	6.31×10^{-6}	✓
			Size	14.937	5.93×10^{-12}	✓
			IoU	-29.440	2.57×10^{-17}	✓
		0.95	Coverage Gap	-14.895	6.23×10^{-12}	✓
			Size	2.744	0.013	✓
			IoU	-6.807	1.69×10^{-6}	✓

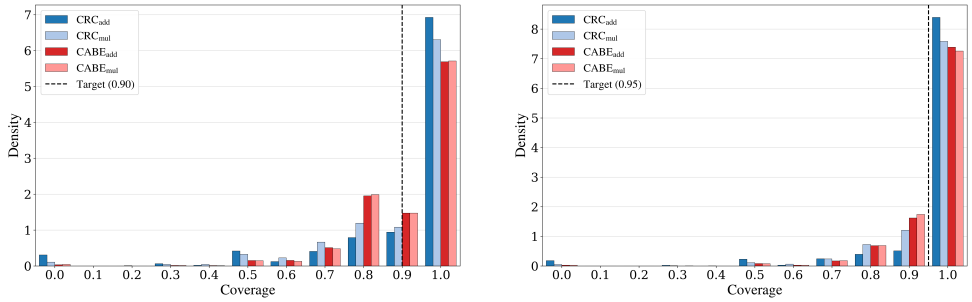
demonstrate that both $CABE_{add}$ and $CABE_{mul}$ achieve a substantially higher sample density directly at the target coverage level $(1-\alpha)$. Furthermore, under the proposed CABE framework, the number of samples falling into the low-coverage region is almost zero. This indicates that our method reliably prevents severe under-coverage failures on individual images. In contrast, the distribution of the baseline methods significantly underperforms, as both CRC_{add} and CRC_{mul} exhibit a wider spread with more samples failing to meet the safety requirements. These visual results firmly corroborate that CABE successfully overcomes the limitations of marginal global thresholds by delivering strict and consistent instance-level risk control across diverse scenarios.



(a) AFO Dataset



(b) DroneVehicle Dataset



(c) Rescue Dataset

Figure 10: Density distribution of image-level coverage for $\mathrm{CRC}_{\mathrm{add/mul}}$ additive and multiplicative methods and $\mathrm{CABE}_{\mathrm{add/mul}}$ additive and multiplicative methods using the YOLOv8 model on the AFO, DroneVehicle, and Rescue datasets.

A.13 Training Loss Curves

In this section, we provide the training loss curves for the boundary expansion models optimized via the CABE framework. As illustrated in Figure 11, the models demonstrate stable and rapid convergence across all three datasets (AFO, DroneVehicle, and Rescue) for both YOLOv8 and

YOLOv10 base detectors. This rapid convergence confirms the effectiveness and stability of the proposed differentiable soft-IoG training objective.

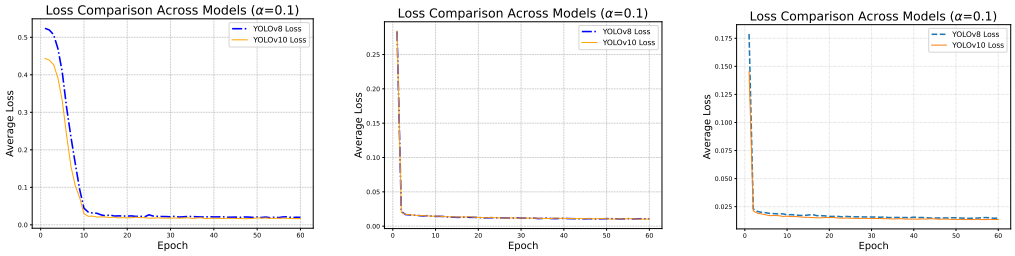


Figure 11: Training loss curves for different boundary expansion models trained using the CABE method; datasets shown from left to right are AFO, DroneVehicle, and Rescue, respectively.

A.14 Efficacy of Uncalibrated CABE

In this appendix, to verify the efficiency of the CABE model in an uncalibrated state, we conduct a comprehensive comparison among the baseline Conformal Risk Control ($\mathrm{CRC}_{\mathrm{mul}}$), the adaptive model without global calibration ($\mathrm{CABE}_{\mathrm{nocel}}$), and the complete model with final calibration ($\mathrm{CABE}_{\mathrm{mul}}$). Table 8 presents the quantitative results across three datasets evaluated at a target risk level of $\alpha = 0.1$ (i.e., a desired coverage of 90%).

The analysis of the experimental results demonstrates that relying solely on the adaptive boundaries output by the neural network ($\mathrm{CABE}_{\mathrm{nocel}}$) can already significantly reduce the Coverage Gap. This indicates that during the training phase, the model successfully learns effective image-adaptive feature patterns to represent uncertainty across instances of varying difficulty. Furthermore, in specific datasets such as DroneVehicle, $\mathrm{CABE}_{\mathrm{nocel}}$ generates more compact bounding boxes (smaller Size) and achieves higher localization precision (higher Mean IoU). However, due to the lack of post-hoc conformal calibration to provide rigorous statistical guarantees, the direct output of the neural network is prone to marginal coverage fluctuations. For example, in the DroneVehicle dataset, the Joint Coverage of $\mathrm{CABE}_{\mathrm{nocel}}$ ranges only from 0.873 to 0.878, failing to meet the predefined 0.90 target safety threshold.

After introducing the final global calibration step, the complete $\mathrm{CABE}_{\mathrm{mul}}$ method successfully restores and strictly maintains both the joint coverage and box coverage at the target safety level. More importantly, the calibrated $\mathrm{CABE}_{\mathrm{mul}}$ perfectly preserves the advantage of a significantly reduced coverage gap achieved by the neural network (yielding a relative reduction of 7.9% to 12.5% compared to the baseline $\mathrm{CRC}_{\mathrm{mul}}$). This comparison highlights that the uncalibrated CABE model is highly efficient in optimizing the uniformity and compactness of conditional coverage, while the subsequent calibration step acts as an indispensable baseline guarantee for marginal coverage safety.

A.15 Time Metrics and Ablation Study

In this section, we comprehensively evaluate the computational efficiency of the proposed method and present the quantitative results of the ablation study regarding the output dimensionality and expansion schemes of the CABE predictor models. At a significance level of $\alpha = 0.1$, we performed 20 independent runs for each method based on the YOLOv8 object detection model and calculated the mean and standard deviation of the results.

Specifically, we investigate three dimensionality settings: CABE^1 (1D scalar), CABE^2 (2D anisotropic), and CABE^4 (4D expansion). Furthermore, for each dimensionality, we evaluate two distinct expansion schemes: Additive, which applies pixel-level offsets, and Multiplicative , which applies ratio-based scaling.

First, Table 9 compares the time overhead of CRC and CABE with different output dimensions and schemes during the training, calibration, and testing phases. The experimental data indicate that across the training and calibration phases of all datasets, CABE^2 incurs only a minimal time increment compared to CABE^1; however, as the dimensionality further extends to CABE^4, the

Table 8: Comparison of Joint Coverage, Box Coverage, Coverage Gap, Size, and Mean IoU across different models and methods (baseline $\mathrm{CRC}_{\mathrm{mul}}$, uncalibrated $\mathrm{CABE}_{\mathrm{nocel}}$, and fully calibrated $\mathrm{CABE}_{\mathrm{mul}}$) at the target risk level $\alpha = 0.1$. The results for each dataset represent the mean and standard deviation over 20 random splits. The percentages with downward arrows in the Coverage Gap column indicate the relative reduction compared to the baseline $\mathrm{CRC}_{\mathrm{mul}}$.

Dataset	Model	Method	Joint Coverage	Box Coverage	Coverage Gap	Size	IoU
AFO	YOLOv8	$\mathrm{CRC}_{\mathrm{mul}}$	0.900 (0.012)	0.906 (0.012)	0.119 (0.008)	**1.942 (0.017)**	0.535 (0.005)
		$\mathrm{CABE}_{\mathrm{nocel}}$	0.901 (0.011)	0.907 (0.011)	0.108 (0.008) (↓ 9.3%)	1.950 (0.046)	**0.573 (0.008)**
		$\mathrm{CABE}_{\mathrm{mul}}$	0.903 (0.009)	0.909 (0.010)	**0.107 (0.007) (↓ 10.1%)**	1.959 (0.040)	0.571 (0.007)
	YOLOv10	$\mathrm{CRC}_{\mathrm{mul}}$	0.896 (0.011)	0.902 (0.010)	0.120 (0.008)	1.805 (0.012)	0.556 (0.004)
		$\mathrm{CABE}_{\mathrm{nocel}}$	0.892 (0.011)	0.899 (0.011)	0.113 (0.007) (↓ 6.1%)	**1.788 (0.038)**	**0.586 (0.007)**
		$\mathrm{CABE}_{\mathrm{mul}}$	0.899 (0.010)	0.906 (0.010)	**0.109 (0.005) (↓ 9.2%)**	1.811 (0.030)	0.579 (0.010)
DroneVehicle	YOLOv8	$\mathrm{CRC}_{\mathrm{mul}}$	0.896 (0.004)	0.902 (0.004)	0.112 (0.002)	1.621 (0.003)	0.625 (0.001)
		$\mathrm{CABE}_{\mathrm{nocel}}$	0.878 (0.009)	0.884 (0.009)	0.106 (0.005) (↓ 5.4%)	**1.576 (0.021)**	**0.664 (0.008)**
		$\mathrm{CABE}_{\mathrm{mul}}$	0.894 (0.004)	0.900 (0.004)	**0.099 (0.002) (↓ 11.6%)**	1.612 (0.011)	0.650 (0.004)
	YOLOv10	$\mathrm{CRC}_{\mathrm{mul}}$	0.895 (0.003)	0.900 (0.004)	0.112 (0.002)	1.615 (0.003)	0.629 (0.002)
		$\mathrm{CABE}_{\mathrm{nocel}}$	0.873 (0.009)	0.879 (0.009)	0.108 (0.005) (↓ 3.9%)	**1.559 (0.021)**	**0.669 (0.007)**
		$\mathrm{CABE}_{\mathrm{mul}}$	0.894 (0.004)	0.899 (0.004)	**0.098 (0.003) (↓ 12.5%)**	1.603 (0.010)	0.652 (0.004)
Rescue	YOLOv8	$\mathrm{CRC}_{\mathrm{mul}}$	0.898 (0.004)	0.900 (0.004)	0.128 (0.003)	**1.567 (0.004)**	**0.657 (0.002)**
		$\mathrm{CABE}_{\mathrm{nocel}}$	0.907 (0.006)	0.909 (0.006)	**0.115 (0.003) (↓ 10.2%)**	1.574 (0.029)	0.642 (0.010)
		$\mathrm{CABE}_{\mathrm{mul}}$	0.907 (0.006)	0.910 (0.006)	**0.115 (0.003) (↓ 10.2%)**	1.575 (0.029)	0.642 (0.010)
	YOLOv10	$\mathrm{CRC}_{\mathrm{mul}}$	0.898 (0.002)	0.899 (0.003)	0.126 (0.002)	**1.484 (0.003)**	**0.668 (0.001)**
		$\mathrm{CABE}_{\mathrm{nocel}}$	0.906 (0.007)	0.907 (0.007)	**0.116 (0.004) (↓ 7.9%)**	1.503 (0.023)	0.658 (0.008)
		$\mathrm{CABE}_{\mathrm{mul}}$	0.906 (0.007)	0.907 (0.007)	**0.116 (0.004) (↓ 7.9%)**	1.503 (0.023)	0.658 (0.008)

Table 9: Comparison of time overhead across training, calibration, and testing phases for different datasets. The evaluation utilizes the YOLOv8 base model with $\alpha = 0.1$ over 20 random runs, contrasting both Additive (add) and Multiplicative (mul) expansion schemes.

Data	Method	Number of Set Samples			Training Time (s)	Calibration Time (s)	Testing Time (s)
		Train	Calibration	Testing			
AFO	$\mathrm{CRC}_{\mathrm{add}}$	0	1275	546	0.000 (0.000)	0.411 (0.091)	0.015 (0.002)
	$\mathrm{CRC}_{\mathrm{mul}}$	0	1275	546	0.000 (0.000)	0.524 (0.102)	0.014 (0.002)
	$\mathrm{CABE}^{1}_{\mathrm{add}}$	893	382	546	441.756 (7.865)	5.317 (0.286)	6.117 (0.297)
	$\mathrm{CABE}^{1}_{\mathrm{mul}}$	893	382	546	441.756 (7.865)	5.228 (0.441)	6.374 (0.507)
	$\mathrm{CABE}^{2}_{\mathrm{add}}$	893	382	546	443.075 (17.568)	5.386 (0.282)	6.151 (0.270)
	$\mathrm{CABE}^{2}_{\mathrm{mul}}$	893	382	546	443.075 (17.568)	5.321 (0.459)	6.407 (0.450)
	$\mathrm{CABE}^{4}_{\mathrm{add}}$	893	382	546	445.376 (24.795)	5.403 (0.169)	6.179 (0.362)
	$\mathrm{CABE}^{4}_{\mathrm{mul}}$	893	382	546	445.376 (24.795)	5.415 (0.388)	6.442 (0.387)
DroneVehicle	$\mathrm{CRC}_{\mathrm{add}}$	0	6064	2599	0.000 (0.000)	2.850 (0.237)	0.140 (0.033)
	$\mathrm{CRC}_{\mathrm{mul}}$	0	6064	2599	0.000 (0.000)	3.662 (0.478)	0.081 (0.011)
	$\mathrm{CABE}^{1}_{\mathrm{add}}$	4245	1819	2599	855.364 (7.076)	5.803 (0.313)	5.771 (0.316)
	$\mathrm{CABE}^{1}_{\mathrm{mul}}$	4245	1819	2599	855.364 (7.076)	5.672 (0.514)	5.518 (0.215)
	$\mathrm{CABE}^{2}_{\mathrm{add}}$	4245	1819	2599	871.768 (5.916)	5.852 (0.375)	5.775 (0.441)
	$\mathrm{CABE}^{2}_{\mathrm{mul}}$	4245	1819	2599	871.768 (5.916)	6.540 (0.300)	5.820 (0.523)
	$\mathrm{CABE}^{4}_{\mathrm{add}}$	4245	1819	2599	947.905 (32.253)	6.250 (0.697)	5.778 (0.395)
	$\mathrm{CABE}^{4}_{\mathrm{mul}}$	4245	1819	2599	947.905 (32.253)	6.765 (0.505)	5.867 (0.293)
Rescue	$\mathrm{CRC}_{\mathrm{add}}$	0	10415	4464	0.000 (0.000)	3.073 (0.211)	0.065 (0.013)
	$\mathrm{CRC}_{\mathrm{mul}}$	0	10415	4464	0.000 (0.000)	4.167 (0.351)	0.084 (0.012)
	$\mathrm{CABE}^{1}_{\mathrm{add}}$	7291	3124	4464	2209.410 (43.090)	18.809 (0.777)	24.467 (0.953)
	$\mathrm{CABE}^{1}_{\mathrm{mul}}$	7291	3124	4464	2209.410 (43.090)	21.887 (0.737)	27.121 (0.616)
	$\mathrm{CABE}^{2}_{\mathrm{add}}$	7291	3124	4464	2211.890 (63.805)	18.897 (0.759)	24.764 (1.733)
	$\mathrm{CABE}^{2}_{\mathrm{mul}}$	7291	3124	4464	2211.890 (63.805)	21.922 (0.424)	27.246 (0.907)
	$\mathrm{CABE}^{4}_{\mathrm{add}}$	7291	3124	4464	2258.603 (55.454)	19.251 (1.218)	24.896 (1.002)
	$\mathrm{CABE}^{4}_{\mathrm{mul}}$	7291	3124	4464	2258.603 (55.454)	22.261 (1.272)	27.702 (1.413)

computational latency begins to rise significantly. It is worth noting that the choice between Additive and Multiplicative schemes has a negligible impact on computational latency across all phases.

Second, Table 10 provides the quantitative metrics of the ablation study, aiming to evaluate the impact of expansion dimensionality on risk control performance. The results show that, compared to CABE^{1} and CABE^{4}, CABE^{2} optimizes the average size of prediction boxes while maintaining target coverage, achieving the lowest or tied-lowest Coverage Gap. Regarding the expansion schemes,

both Additive and Multiplicative variants exhibit similar trends across dimensions, demonstrating that the method's effectiveness is robust to the specific arithmetic implementation of the expansion.

Remark 4 *Synthesizing the above experimental results, we can draw the following conclusions: the time overhead of all methods during the testing phase remains nearly identical regardless of changes in output dimensionality or expansion schemes, which indicates that expanding the parameter space of the predictor does not sacrifice the real-time performance of the model during inference; the leap in training cost from* CABE^2 *to* CABE^4 *is significantly higher than the increment from* CABE^1 *to* CABE^2*, which confirms that the computational burden increases non-linearly with the increase of prediction dimensionality; the higher-dimensional parameterization scheme (*CABE^4*) has entered a stage of diminishing returns, failing to yield substantial improvements in accuracy while increasing the training burden. In summary, we select* CABE^2 *as the final configuration, as it achieves optimal prediction quality optimization with a minimal computational increment.*

Table 10: Quantitative ablation study comparing Additive (add) and Multiplicative (mul) schemes across different CABE dimensionality settings. The results are reported as mean (standard deviation) based on the YOLOv8 model with $\alpha = 0.1$ over 20 random runs.

Dataset	Method	Joint Coverage	Box Coverage	Coverage Gap	Size	IoU
AFO	CABE^1_{add}	0.895 (0.010)	0.902 (0.012)	0.111 (0.008)	1.990 (0.044)	0.584 (0.009)
	CABE^1_{mul}	0.896 (0.010)	0.903 (0.011)	**0.110 (0.007)**	1.962 (0.041)	**0.587 (0.008)**
	CABE^2_{add}	0.895 (0.011)	0.902 (0.012)	0.111 (0.008)	1.930 (0.029)	0.578 (0.010)
	CABE^2_{mul}	0.895 (0.012)	0.902 (0.013)	0.111 (0.009)	1.924 (0.033)	0.579 (0.011)
	CABE^4_{add}	0.895 (0.013)	0.902 (0.014)	0.111 (0.008)	**1.896 (0.039)**	0.573 (0.009)
	CABE^4_{mul}	0.896 (0.012)	0.902 (0.013)	0.111 (0.008)	1.913 (0.045)	0.570 (0.010)
DroneVehicle	CABE^1_{add}	0.894 (0.004)	0.900 (0.004)	0.099 (0.002)	1.625 (0.011)	0.652 (0.004)
	CABE^1_{mul}	0.894 (0.004)	0.900 (0.004)	0.099 (0.002)	1.626 (0.010)	0.652 (0.003)
	CABE^2_{add}	0.894 (0.004)	0.900 (0.004)	**0.099 (0.002)**	1.605 (0.010)	0.653 (0.004)
	CABE^2_{mul}	0.894 (0.004)	0.900 (0.004)	0.099 (0.002)	1.617 (0.010)	0.651 (0.004)
	CABE^4_{add}	0.895 (0.005)	0.901 (0.005)	0.100 (0.002)	**1.581 (0.010)**	**0.654 (0.003)**
	CABE^4_{mul}	0.895 (0.004)	0.901 (0.004)	0.100 (0.002)	1.603 (0.012)	0.649 (0.003)
Rescue	CABE^1_{add}	0.898 (0.003)	0.900 (0.003)	0.120 (0.002)	1.587 (0.018)	0.649 (0.005)
	CABE^1_{mul}	0.899 (0.003)	0.901 (0.003)	0.119 (0.002)	1.557 (0.011)	**0.658 (0.003)**
	CABE^2_{add}	0.899 (0.004)	0.901 (0.004)	**0.119 (0.002)**	1.555 (0.014)	0.650 (0.004)
	CABE^2_{mul}	0.899 (0.004)	0.901 (0.004)	0.119 (0.002)	1.546 (0.010)	0.653 (0.004)
	CABE^4_{add}	0.898 (0.004)	0.900 (0.004)	0.120 (0.002)	**1.527 (0.009)**	0.653 (0.004)
	CABE^4_{mul}	0.898 (0.004)	0.900 (0.004)	0.120 (0.002)	1.536 (0.010)	0.650 (0.004)

A.16 Hyperparameter Sensitivity Analysis

In this section, we conduct a grid search sensitivity analysis for the hyperparameters β and λ_{reg} across various datasets using the YOLOv8 model. All experiments are performed with 10 random splits, and the results are reported as mean $\pm$ standard deviation.

For the β analysis, we set $\alpha = 0.1$ and evaluate $\beta \in \{1, 10, 100, 500, 1000, 5000\}$ (Table 11). The grid search results indicate that the coverage gap is minimized across all three datasets when $\beta = 100$. Regarding the regularization parameter λ_{reg}, with $\alpha = 0.1$, we test $\lambda_{\text{reg}} \in \{0.001, 0.01, 0.1, 0.5, 1, 10, 100\}$ (Table 12). The findings demonstrate that $\lambda_{\text{reg}} = 0.1$ consistently yields the optimal coverage gap among all tested values. Based on these results, we set $\beta = 100$ and $\lambda_{\text{reg}} = 0.1$ for all subsequent experiments to ensure optimal performance.

A.17 Sensitivity Analysis of Training Set Ratios

To evaluate the robustness of the CABE model and the effect of the training dataset size on its performance, we conducted a sensitivity analysis varying the CABE training ratio from 0.2 to 0.8 across the AFO, DroneVehicle, and Rescue datasets. The results, summarized in Table 13, demonstrate that CABE consistently reduces the coverage gap compared to the CRC baseline, regardless of the training-calibration split. Notably, as the training ratio increases, the coverage gap generally exhibits a downward trend, while maintaining stable joint and box coverages around the target nominal level.

Table 11: Joint Coverage, Box Coverage, Coverage Gap, Size, and Mean IoU results at $\alpha = 0.1$ across different β values. Each dataset result represents the mean and standard deviation over 10 random splits.

Dataset	β	Joint Coverage	Box Coverage	Coverage Gap	Size	IoU
AFO	1	0.895 (0.011)	0.902 (0.012)	0.117 (0.007)	1.895 (0.054)	**0.571 (0.014)**
	10	0.897 (0.014)	0.903 (0.016)	0.113 (0.009)	**1.859 (0.028)**	0.567 (0.010)
	100	0.901 (0.009)	0.908 (0.011)	**0.110 (0.006)**	1.958 (0.032)	0.560 (0.014)
	500	0.893 (0.009)	0.900 (0.010)	0.115 (0.007)	1.998 (0.042)	0.566 (0.013)
	1000	0.892 (0.015)	0.899 (0.018)	0.118 (0.013)	2.040 (0.063)	0.557 (0.018)
	5000	0.891 (0.014)	0.897 (0.015)	0.119 (0.007)	2.148 (0.117)	0.558 (0.012)
DroneVehicle	1	0.894 (0.005)	0.900 (0.005)	0.110 (0.003)	1.607 (0.010)	0.631 (0.004)
	10	0.893 (0.003)	0.899 (0.003)	0.102 (0.002)	**1.571 (0.006)**	**0.651 (0.003)**
	100	0.891 (0.004)	0.897 (0.004)	**0.102 (0.003)**	1.624 (0.010)	0.643 (0.004)
	500	0.892 (0.005)	0.899 (0.005)	0.102 (0.004)	1.667 (0.021)	0.633 (0.004)
	1000	0.894 (0.004)	0.900 (0.003)	0.103 (0.002)	1.674 (0.013)	0.631 (0.002)
	5000	0.892 (0.004)	0.898 (0.004)	0.106 (0.004)	1.683 (0.031)	0.631 (0.007)
Rescue	1	0.899 (0.005)	0.901 (0.005)	0.139 (0.003)	2.117 (0.051)	0.520 (0.009)
	10	0.899 (0.004)	0.901 (0.004)	0.124 (0.003)	**1.539 (0.009)**	0.638 (0.003)
	100	0.900 (0.005)	0.901 (0.005)	**0.120 (0.003)**	1.579 (0.009)	0.637 (0.004)
	500	0.899 (0.005)	0.901 (0.005)	0.121 (0.003)	1.619 (0.026)	0.643 (0.005)
	1000	0.899 (0.005)	0.901 (0.005)	0.123 (0.003)	1.643 (0.045)	**0.634 (0.011)**
	5000	0.898 (0.005)	0.900 (0.005)	0.126 (0.003)	1.682 (0.038)	0.626 (0.011)

Table 12: Joint Coverage, Box Coverage, Coverage Gap, Size, and Mean IoU results at $\alpha = 0.1$ across different λ_{reg} values. Each dataset result represents the mean and standard deviation over 10 random splits.

Dataset	λ_{reg}	Joint Coverage	Box Coverage	Coverage Gap	Size	IoU
AFO	0.001	0.893 (0.006)	0.900 (0.008)	0.120 (0.006)	2.727 (0.236)	**0.593 (0.009)**
	0.01	0.894 (0.010)	0.901 (0.010)	0.116 (0.007)	2.254 (0.147)	0.592 (0.007)
	0.1	0.895 (0.015)	0.902 (0.016)	**0.111 (0.010)**	1.920 (0.037)	0.580 (0.010)
	0.5	0.895 (0.014)	0.902 (0.016)	0.112 (0.009)	1.866 (0.037)	0.575 (0.010)
	1	0.893 (0.015)	0.899 (0.016)	0.115 (0.010)	**1.844 (0.045)**	0.574 (0.013)
	10	0.901 (0.016)	0.908 (0.017)	0.118 (0.010)	1.916 (0.052)	0.547 (0.014)
	100	0.897 (0.013)	0.904 (0.012)	0.121 (0.008)	1.987 (0.048)	0.547 (0.013)
DroneVehicle	0.001	0.893 (0.004)	0.899 (0.004)	0.104 (0.006)	1.780 (0.071)	0.635 (0.013)
	0.01	0.893 (0.004)	0.899 (0.004)	0.100 (0.002)	1.670 (0.019)	0.650 (0.005)
	0.1	0.892 (0.003)	0.898 (0.003)	**0.100 (0.002)**	1.603 (0.010)	**0.654 (0.003)**
	0.5	0.894 (0.003)	0.900 (0.004)	0.101 (0.002)	**1.573 (0.005)**	0.650 (0.004)
	1	0.894 (0.004)	0.900 (0.005)	0.103 (0.002)	1.569 (0.007)	0.649 (0.005)
	10	0.895 (0.005)	0.900 (0.005)	0.110 (0.003)	1.575 (0.012)	0.638 (0.006)
	100	0.895 (0.005)	0.900 (0.006)	0.118 (0.003)	1.643 (0.014)	0.614 (0.005)
Rescue	0.001	0.897 (0.003)	0.899 (0.003)	0.132 (0.002)	2.020 (0.060)	0.618 (0.008)
	0.01	0.899 (0.002)	0.901 (0.002)	0.124 (0.001)	1.729 (0.037)	0.637 (0.007)
	0.1	0.899 (0.004)	0.901 (0.004)	**0.119 (0.003)**	1.555 (0.014)	**0.651 (0.003)**
	0.5	0.899 (0.005)	0.901 (0.005)	0.122 (0.003)	**1.524 (0.010)**	0.648 (0.005)
	1	0.899 (0.004)	0.901 (0.004)	0.124 (0.003)	1.538 (0.012)	0.642 (0.005)
	10	0.900 (0.005)	0.902 (0.005)	0.132 (0.003)	1.713 (0.026)	0.594 (0.007)
	100	0.899 (0.004)	0.901 (0.004)	0.137 (0.003)	2.016 (0.037)	0.549 (0.007)

Additionally, the mean prediction interval width (Size) remains competitive, and the prediction quality (Mean IoU) shows slight improvements with more training data. This indicates that CABE is robust to smaller training sets but can incrementally benefit from a larger training split to achieve tighter coverage gaps and higher bounding box quality.

Table 13: Performance comparison of CABE and CRC baselines using YOLOv8 and YOLOv10 across AFO, DroneVehicle, and Rescue datasets with varying training set ratios. The reported values represent the mean (standard deviation) evaluated over 10 independent runs. The percentages in parentheses with downward arrows indicate the relative reduction in the coverage gap compared to the CRC baseline.

Dataset	Model	Method	Train Ratio	Joint Coverage	Box Coverage	Coverage Gap	Size	IoU
AFO	YOLOv8	CRC	-	0.901 (0.013)	0.908 (0.013)	0.118 (0.009)	1.943 (0.018)	0.535 (0.005)
		CABE	0.2	0.899 (0.007)	0.906 (0.008)	0.115 (0.007) (↓ 2.5%)	2.098 (0.067)	0.552 (0.013)
			0.3	0.896 (0.013)	0.903 (0.014)	0.113 (0.007) (↓ 4.2%)	1.961 (0.039)	0.571 (0.011)
			0.4	0.897 (0.016)	0.903 (0.017)	0.113 (0.009) (↓ 4.2%)	1.939 (0.039)	0.571 (0.012)
			0.6	0.894 (0.016)	0.900 (0.017)	0.113 (0.011) (↓ 4.2%)	1.920 (0.033)	0.578 (0.010)
			0.7	0.896 (0.015)	0.902 (0.016)	**0.111** (0.010) (↓ 5.9%)	1.919 (0.037)	**0.581** (0.010)
			0.8	0.894 (0.013)	0.900 (0.015)	0.112 (0.010) (↓ 5.1%)	**1.917** (0.055)	0.579 (0.016)
	YOLOv10	CRC	-	0.896 (0.011)	0.901 (0.011)	0.121 (0.009)	1.804 (0.012)	0.556 (0.005)
		CABE	0.2	0.896 (0.006)	0.902 (0.006)	0.117 (0.006) (↓ 3.3%)	1.912 (0.051)	0.567 (0.020)
			0.3	0.897 (0.008)	0.903 (0.007)	0.112 (0.005) (↓ 7.4%)	1.826 (0.024)	0.581 (0.008)
			0.4	0.898 (0.007)	0.904 (0.008)	0.111 (0.005) (↓ 8.3%)	1.821 (0.030)	0.579 (0.013)
			0.6	0.900 (0.010)	0.906 (0.011)	**0.109** (0.006) (↓ 9.9%)	1.801 (0.029)	0.580 (0.009)
			0.7	0.899 (0.013)	0.905 (0.013)	**0.109** (0.006) (↓ 9.9%)	1.800 (0.037)	0.580 (0.014)
			0.8	0.895 (0.012)	0.901 (0.012)	0.111 (0.007) (↓ 8.3%)	**1.798** (0.031)	**0.588** (0.014)
DroneVehicle	YOLOv8	CRC	-	0.896 (0.004)	0.902 (0.004)	0.112 (0.002)	1.621 (0.003)	0.625 (0.001)
		CABE	0.2	0.895 (0.004)	0.901 (0.004)	0.101 (0.004) (↓ 9.8%)	1.615 (0.009)	0.645 (0.006)
			0.3	0.894 (0.004)	0.900 (0.004)	0.100 (0.002) (↓ 10.7%)	1.608 (0.007)	0.648 (0.004)
			0.4	0.894 (0.004)	0.900 (0.004)	0.100 (0.002) (↓ 10.7%)	1.610 (0.010)	0.648 (0.003)
			0.6	0.893 (0.004)	0.899 (0.004)	0.100 (0.002) (↓ 10.7%)	**1.608** (0.008)	0.650 (0.004)
			0.7	0.894 (0.004)	0.900 (0.004)	**0.099** (0.002) (↓ 11.6%)	1.612 (0.009)	**0.651** (0.003)
			0.8	0.895 (0.006)	0.901 (0.006)	**0.099** (0.003) (↓ 11.6%)	1.610 (0.012)	0.650 (0.006)
	YOLOv10	CRC	-	0.895 (0.003)	0.900 (0.004)	0.112 (0.002)	1.615 (0.003)	0.629 (0.002)
		CABE	0.2	0.894 (0.003)	0.900 (0.003)	0.100 (0.002) (↓ 10.7%)	1.607 (0.011)	0.650 (0.004)
			0.3	0.893 (0.003)	0.899 (0.003)	0.100 (0.002) (↓ 10.7%)	1.604 (0.007)	0.651 (0.004)
			0.4	0.894 (0.003)	0.900 (0.003)	0.099 (0.002) (↓ 11.6%)	1.605 (0.010)	0.652 (0.003)
			0.6	0.894 (0.003)	0.899 (0.003)	**0.098** (0.002) (↓ 12.5%)	1.604 (0.009)	**0.652** (0.003)
			0.7	0.894 (0.004)	0.899 (0.004)	**0.098** (0.003) (↓ 12.5%)	**1.603** (0.010)	**0.652** (0.004)
			0.8	0.894 (0.004)	0.900 (0.004)	**0.098** (0.002) (↓ 12.5%)	1.605 (0.009)	**0.652** (0.004)
Rescue	YOLOv8	CRC	-	0.898 (0.004)	0.900 (0.004)	0.128 (0.003)	1.567 (0.004)	0.657 (0.002)
		CABE	0.2	0.899 (0.003)	0.901 (0.003)	0.121 (0.002) (↓ 5.5%)	1.554 (0.011)	0.648 (0.005)
			0.3	0.899 (0.003)	0.901 (0.003)	0.120 (0.002) (↓ 6.2%)	1.551 (0.008)	0.649 (0.003)
			0.4	0.899 (0.003)	0.901 (0.003)	0.120 (0.002) (↓ 6.2%)	1.552 (0.011)	0.650 (0.003)
			0.6	0.899 (0.004)	0.901 (0.004)	**0.119** (0.002) (↓ 7.0%)	**1.547** (0.009)	0.651 (0.004)
			0.7	0.899 (0.004)	0.901 (0.004)	**0.119** (0.002) (↓ 7.0%)	1.549 (0.012)	0.652 (0.004)
			0.8	0.899 (0.006)	0.900 (0.006)	**0.119** (0.003) (↓ 7.0%)	1.549 (0.015)	**0.653** (0.007)
	YOLOv10	CRC	-	0.898 (0.003)	0.899 (0.003)	0.126 (0.002)	1.484 (0.003)	0.668 (0.001)
		CABE	0.2	0.897 (0.003)	0.899 (0.003)	0.125 (0.002) (↓ 0.5%)	1.483 (0.007)	0.664 (0.004)
			0.3	0.898 (0.003)	0.900 (0.003)	0.124 (0.002) (↓ 1.3%)	1.481 (0.007)	0.665 (0.003)
			0.4	0.897 (0.003)	0.899 (0.003)	0.124 (0.002) (↓ 1.4%)	1.479 (0.007)	0.666 (0.003)
			0.6	0.897 (0.003)	0.899 (0.003)	**0.120** (0.002) (↓ 4.8%)	**1.479** (0.009)	0.667 (0.004)
			0.7	0.897 (0.003)	0.899 (0.003)	**0.120** (0.003) (↓ 4.8%)	**1.479** (0.011)	0.668 (0.003)
			0.8	0.896 (0.003)	0.898 (0.003)	**0.120** (0.002) (↓ 4.8%)	1.480 (0.008)	**0.669** (0.005)

A.18 CABE Core Code Implementation

To enhance research reproducibility, this appendix provides the core PyTorch implementation details for the **CABE**method. The implementation is divided into three primary stages: Multi-Class Threshold CRC Calibration ,the end-to-end training of the adaptive boundary predictor and the subsequent post-hoc calibration and inference process.

A.18.1 Algorithm for classification thresholds

This stage aims to control the classification recall rate by establishing a global confidence threshold. We employ a scalar CRC strategy to identify a single threshold λ for all categories. Unlike object-level pooling, we optimize for the **image-level** risk, ensuring that the expected False Negative Rate (FNR) per image is bounded by α_1, subject to the finite-sample correction required by the CRC framework.

Algorithm 2 Python Pseudo-Code for Classification CRC: Calibration Phase. This algorithm calculates the single global threshold λ. It computes the empirical risk as the average FNR across images and applies the finite-sample correction term ($\frac{1}{n+1}$) to satisfy the risk budget α_1.

```python
import numpy as np

def calibrate_classification_thresholds(calib_data, alpha_1):
    """
    Phase 1: Compute a single global conformal threshold lambda for all classes.
    Objective: Ensure the CRC-corrected Image-level FNR <= alpha_1.

    :param calib_data: List of arrays. Each element `calib_data[i]` contains the
                       prediction scores of the matched ground-truth objects
                       for the i-th calibration image.
                       Example: [[0.9, 0.8], [0.4], [0.99, 0.95, 0.88], ...]
    :param alpha_1: Target risk level (e.g., 0.01 implies target coverage 0.99).
    """
    n = len(calib_data) # Number of calibration images
    if n == 0: return 0.0

    def compute_crc_risk(lam):
        """
        Computes the CRC upper bound of the image-level risk for a candidate lambda.
        Risk definition: Image-level False Negative Rate (1 - Recall).
        """
        image_risks = []
        for scores in calib_data:
            scores = np.array(scores)
            num_gt = len(scores)

            if num_gt == 0:
                # If an image has no objects, risk is 0 (pass by default)
                # or excluded depending on definition. We assume risk=0 here.
                image_risks.append(0.0)
                continue

            # Calculate Recall for this specific image
            # Count how many matched objects have score >= lambda
            num_recalled = np.sum(scores >= lam)
            recall_i = num_recalled / num_gt

            # Risk_i = 1 - Coverage_i
            image_risks.append(1.0 - recall_i)

        # Empirical Risk = Average of Image Risks
        R_hat = np.mean(image_risks)

        # CRC Finite-Sample Correction (Equation 3 in paper)
        # Bound = (n / (n+1)) * R_hat + (B / (n+1)), where B=1 for FNR
        crc_bound = (n / (n + 1)) * R_hat + (1.0 / (n + 1))

        return crc_bound

    # Binary search for the maximal threshold lambda
    # We want the highest lambda such that the risk bound is still <= alpha_1
    low, high = 0.0, 1.0
    final_threshold = 0.0

    # 20 iterations give precision ~1e-6
    for _ in range(20):
        mid = (low + high) / 2.0

        # If risk is controlled, try to increase lambda (make it tighter/less conservative)
        if compute_crc_risk(mid) <= alpha_1:
            final_threshold = mid
            low = mid
        else:
            # Risk is too high, must lower lambda (accept more predictions)
            high = mid

    return final_threshold

def apply_classification_inference(test_predictions, global_lambda):
    """
    [Phase 4] Inference: Filter test predictions using the global calibrated threshold.
    """
    final_results = []
    for pred in test_predictions:
        # Remove predictions where score < global_lambda
        if pred['score'] >= global_lambda:
            final_results.append(pred)
    return final_results
```

A.18.2 ALGORITHM FOR CABE TRAINING PHASE

The first part details the training procedure for the adaptive boundary extension predictor f_θ. Its core components include a ResNet-50 backbone and the novel differentiable miscoverage loss L_{CABE}. This loss function enables a direct gradient-based optimization process to approximate image-conditional coverage targets.

Algorithm 3 Python Pseudo-Code for CABE: Training Phase. This code details the end-to-end training of the `BoundaryPredictor` network. It includes the network architecture, the differentiable surrogate loss $\mathcal{L}_{CABE}$ approximation using `compute_soft_iog`, and the main training execution within the `cabe_training_step` function.

```python
import torch
import torch.nn as nn
import torch.nn.functional as F
from torchvision.models import resnet50, ResNet50_Weights
class AdaptiveLocalizationNet(nn.Module):
    """Adaptive predictor network f_theta for image-specific adjustment."""
    def __init__(self):
        super().__init__()
        # Backbone: ResNet-50 pretrained on ImageNet
        self.backbone = resnet50(weights=ResNet50_Weights.IMAGENET1K_V1)
        nf = self.backbone.fc.in_features
        self.backbone.fc = nn.Identity()  # Remove default classifier
        # MLP Head for expansion ratio prediction (outputs dw, dh)
        self.head = nn.Sequential(
            nn.Linear(nf, 512), nn.ReLU(True), nn.Dropout(0.3),
            nn.Linear(512, 128), nn.ReLU(True), nn.Dropout(0.2),
            nn.Linear(128, 2)
        )
        # Conservative initialization: encourages small initial adjustments
        nn.init.constant_(self.head[-1].bias, -4.0)
        nn.init.constant_(self.head[-1].weight, 0.0)

    def forward(self, x):
        # Sigmoid restricts output ratios to the [0, 1] range
        return F.sigmoid(self.head(self.backbone(x)))
def cabe_training_step(model, optimizer, imgs, gt_batch, pred_batch, target_cov, beta=100):
    model.train()
    optimizer.zero_grad()
    # 1. Predict image-specific adaptive expansion ratios (deltas)
    deltas = model(imgs)
    loss_batch = 0.0
    valid_img_cnt = 0
    for i in range(len(imgs)):
        if len(gt_batch[i]) == 0:
            continue
        valid_img_cnt += 1
        # 2. Expand boxes differentiably using predicted ratios
        expanded = expand_box_tensor(pred_batch[i].T, deltas[i,0], deltas[i,1])
        # 3. Calculate soft Intersection-over-Ground-truth (IoG) proxy
        inter_area = calculate_intersection(expanded, gt_batch[i].T)
        gt_area = calculate_area(gt_batch[i].T)
        cov_ratios = inter_area / torch.clamp(gt_area, min=1e-6)
        # 4. Apply power-law penalty beta (LOSS_POWER_P) to force containment
        l_main = (torch.mean(torch.pow(cov_ratios, beta)) - target_cov)**2
        loss_batch += l_main + 0.1 * torch.norm(deltas[i], p=2)
    loss = loss_batch / valid_img_cnt
    loss.backward()
    optimizer.step()
    return loss.item()
```

A.18.3 ALGORITHM FOR CABE CALIBRATION PHASE

Upon completing the training of the boundary expansion predictor network f_θ, this section details the post-hoc calibration procedure. The objective is to determine the optimal global correction parameter t' utilizing a held-out calibration set and a binary search.

To accommodate different expansion strategies, the algorithm supports two modes: *Pixel-level* (additive offset based on image size) and *Ratio-level* (relative scaling based on box size). By calculating empirical risk scores, the algorithm identifies the minimal t' that satisfies the target spatial risk level α_2, providing statistical safety guarantees for the adaptive predictions.

Algorithm 4 Python Pseudo-Code for CABE: Calibration Phase. This code illustrates the unified procedure for finding the calibration value t'. The function `calibrate_cabe` leverages the trained predictor to iterate through the calibration set. It utilizes binary search to calculate the optimal global parameter `t_prime` (t'), supporting both `pixel` and `ratio` modes to ensure the model achieves the marginal coverage property under the risk constraint α_2.

```python
def calibrate_cabe(model, calib_loader, target_cov, mode='pixel'):
    """
    Finds the minimal global calibration parameter t_prime via binary search.
    Args:
        target_cov: Equivalent to (1 - alpha_2).
        mode: 'pixel' for additive offset (Scheme A),
              'ratio' for relative scaling (Scheme B).
    """
    model.eval()
    calib_data = []
    # Step 1: Collect adaptive base predictions from the trained model
    with torch.no_grad():
        for imgs, metas, _, _ in calib_loader:
            # Predict adaptive parameters (dw, dh) for each image
            deltas = model(imgs.to(DEVICE))
            for i in range(len(imgs)):
                calib_data.append({
                    'dw': deltas[i, 0].item(),
                    'dh': deltas[i, 1].item(),
                    'matches': metas[i]['raw_matches'],
                    'size': metas[i]['orig_size'] # (W_img, H_img)
                })
    def expand_box(bbox, dw, dh, t, w_img, h_img, mode):
        """Applies expansion logic based on the selected calibration mode."""
        w_box = bbox[2] - bbox[0]
        h_box = bbox[3] - bbox[1]
        if mode == 'pixel':
            # Scheme A: Pixel-level Additive
            # The network output is scaled by image size, then t is added.
            dx = (dw * w_img) + t
            dy = (dh * h_img) + t
        elif mode == 'ratio':
            # Scheme B: Ratio-level (Multiplicative-like)
            # The network output and t are treated as ratios of the box size.
            dx = (dw + t) * w_box
            dy = (dh + t) * h_box
        # Apply symmetric expansion to the bounding box
        return [bbox[0]-dx, bbox[1]-dy, bbox[2]+dx, bbox[3]+dy]
    def eval_coverage(val):
        """Macro-average (image-level) coverage evaluation logic."""
        tot_img_cov, cnt = 0.0, 0
        for item in calib_data:
            if not item['matches']: continue

            # Evaluate coverage for all objects within a single image
            match_covs = [
                calculate_iou_coverage(m['gt_bbox'],
                    expand_box(m['pred_bbox'], item['dw'], item['dh'],
                               val, item['size'][0], item['size'][1], mode))
                for m in item['matches']
            ]
            # Per-image average coverage
            tot_img_cov += np.mean(match_covs)
            cnt += 1
        return tot_img_cov / cnt if cnt > 0 else 0
    # Step 2: Configure Binary Search Range based on Mode
    if mode == 'pixel':
        low, high = -1000.0, 1000.0  # Range in pixels
    else:
        low, high = -1.0, 5.0        # Range in ratio units
    t_prime = high
    # Step 3: Execute Binary Search
    for _ in range(50): # 50 iterations for high precision
        mid = (low + high) / 2
        # Check if the current mid-point satisfies target coverage
        if eval_coverage(mid) >= target_cov:
            t_prime = mid
            high = mid # Seek a tighter (smaller) threshold
        else:
            low = mid  # Coverage insufficient, increase the offset
    return t_prime
```