ENHANCING IMAGE-CONDITIONAL COVERAGE IN SEGMENTATION: ADAPTIVE THRESHOLDING VIA DIFFERENTIABLE MISCOVERAGE LOSS

Anonymous authors
Paper under double-blind review

ABSTRACT

Current deep learning models for image segmentation often lack reliable uncertainty quantification, particularly at the image-specific level. While Conformal Risk Control (CRC) offers marginal statistical guarantees, achieving image-conditional coverage, which ensures prediction sets reliably capture ground truth for individual images, remains a significant challenge. This paper introduces a novel approach to address this gap by learning image-adaptive thresholds for conformal image segmentation. We first propose **AT** (Adaptive Thresholding), which frames threshold prediction as a supervised regression task. Building upon the insights from AT, we then introduce **COAT** (Conditional Optimization for Adaptive Thresholding), an innovative end-to-end differentiable framework. COAT directly optimizes image-conditional coverage by using a soft approximation of the True Positive Rate (TPR) as its loss function, enabling direct gradient-based learning of optimal image-specific thresholds. This novel differentiable miscoverage loss is key to enhancing conditional coverage. Our methods provide a robust pathway towards more trustworthy and interpretable uncertainty estimates in image segmentation, offering improved conditional guarantees crucial for safety-critical applications.

1 INTRODUCTION

Image segmentation is a fundamental computer vision task with critical applications in medical diagnostics, autonomous driving, and remote sensing. While deep learning has significantly advanced segmentation performance, reliable uncertainty quantification remains challenging but essential for safety-critical applications. Traditional evaluation metrics like Dice or IoU provide overall performance measures but fail to offer instance-wise reliability guarantees.

Conformal prediction (CP) has emerged as a powerful framework for providing distribution-free uncertainty quantification with finite-sample guarantees. It constructs prediction regions that contain the true label with a user-specified probability, regardless of the underlying data distribution. Recent work on Conformal Risk Control (CRC) (Angelopoulos et al., 2024) has extended this framework to handle more complex performance metrics beyond simple miscoverage, such as controlling the false negative rate in segmentation tasks.

However, a key limitation of standard CRC, particularly in image-level tasks like segmentation, is its focus on *marginal* guarantees. While CRC ensures that the average risk across a dataset is controlled, the risk for individual images (i.e., the *conditional* risk) can vary substantially. In safety-critical domains, ensuring that each image's prediction is reliable, rather than just the average over many images, is paramount. This image-specific variability in risk is a significant challenge that current approaches struggle to address effectively.

This paper proposes a novel approach to achieve image-conditional coverage in conformal image segmentation by learning image-adaptive thresholds. Our core idea is to train a model that predicts a unique threshold for each input image, aiming to satisfy the desired coverage level for that specific image. We introduce two distinct methods to realize this:

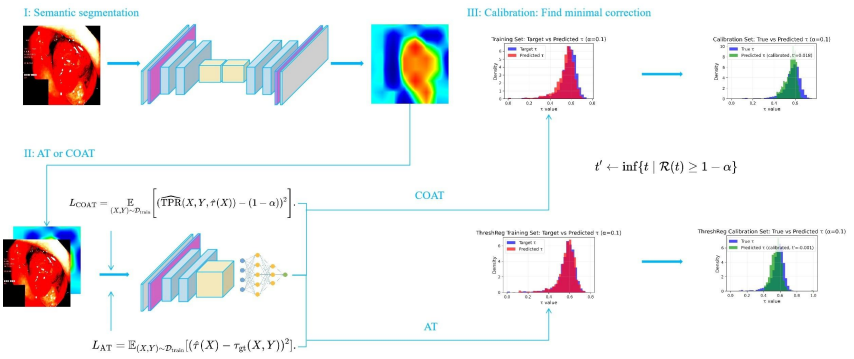

Figure 1: Schematic Overview of the COAT Framework Pipeline.

1. **AT** (Adaptive Thresholding): As an initial step, this method treats the problem of threshold prediction as a supervised regression task. We pre-compute optimal hard thresholds for training images that achieve the target coverage, and then train a neural network to predict these thresholds given the image and its base segmentation model outputs.
2. **COAT** (Conditional Optimization for Adaptive Thresholding): Building upon AT's concept, we propose an innovative end-to-end differentiable framework, which we name COAT (as illustrated in Figure 1). Instead of relying on pre-computed hard thresholds, COAT directly optimizes for image-conditional coverage. It achieves this by utilizing a soft, differentiable approximation of the True Positive Rate (TPR) to define its loss function, enabling direct gradient-based learning of optimal image-specific thresholds. This novel differentiable miscoverage loss is a key contribution for enhancing conditional coverage.

By learning image-adaptive thresholds and, particularly through the end-to-end differentiable optimization of COAT, our methods provide a robust pathway towards more trustworthy and interpretable uncertainty estimates in image segmentation, offering significantly improved conditional guarantees, which are crucial for the deployment of AI systems in high-stakes applications.

2 Preliminaries and Problem Setup

2.1 Problem Setup

For image segmentation, we consider an input image X_i with its ground truth binary mask $Y_i \subset \{1, 2, ..., N\}$, which delineates a region of interest. Our primary objective is to construct a prediction set $\widehat{C}(X_i) \subset \{1, 2, ..., N\}$ that controls the false negative rate (FNR) in expectation. The FNR quantifies the proportion of true positive pixels that are incorrectly excluded from the prediction set, a critical metric in applications such as medical imaging where missing regions of interest can have severe consequences. Specifically, we aim to ensure:

$$\mathbb{E}\left[1 - \frac{\left|\widehat{C}(X_i) \cap Y_i\right|}{|Y_i|}\right] \leq \alpha, \tag{1}$$

where $\alpha \in (0, 1)$ is a user-specified risk level. Here, $|Y_i|$ denotes the cardinality of the set Y_i (i.e., the number of positive pixels in the ground truth mask), and this metric is typically considered for cases where $|Y_i| > 0$. We also define ϵ as a small positive constant (e.g., 10^{-6}) used to prevent division by zero in certain calculations. The expectation in equation 1 is taken over random draws of the test data, reflecting the average performance of the prediction set $\widehat{C}(X_i)$.

However, this marginal guarantee, while ensuring that the average FNR across the entire dataset is controlled, does not guarantee consistent performance for individual images. Due to the inherent variability in image "difficulty" or characteristics, applying a single threshold to all images can lead to over-coverage for "easy" images and severe under-coverage for "difficult" ones. This implies that, while the FNR might be met on average, the conditional coverage (i.e., $1 -$ FNR for a single

image) for specific images can deviate significantly from the target level $1-\alpha$. For safety-critical applications, such image-to-image variability is unacceptable, necessitating a stronger guarantee: not only must the marginal FNR be controlled, but the prediction reliability for each image should also be as close as possible to the target level.

That is, for an input image $X_i \in \mathcal{D}_{\text{test}}$ and its corresponding ground-truth label $Y_i \in \mathcal{D}_{\text{test}}$, we should ensure the fulfillment of the following conditions:

$$\text{Coverage} = \frac{1}{|\mathcal{D}_{test}|} \sum_{X_i \in \mathcal{D}_{test}} \frac{\left|\widehat{C}(X_i) \cap Y_i\right|}{|Y_i|}. \tag{2}$$

Subsequently, we should strive to narrow the coverage gap (Kaur et al., 2025):

$$\text{Coverage Gap} = \frac{1}{|\mathcal{D}_{test}|} \sum_{X_i \in \mathcal{D}_{test}} \left(\left| \frac{\left|\widehat{C}(X_i) \cap Y_i\right|}{|Y_i|} - (1-\alpha) \right| \right). \tag{3}$$

2.2 Conformal Risk Control (CRC)

Conformal Risk Control (CRC) (Angelopoulos et al., 2024) extends the principles of conformal prediction to offer distribution-free guarantees on the expected value of any monotone loss function. For image segmentation, CRC achieves control over the false negative rate by applying a specific threshold to the pixel-wise probabilities generated by a base segmentation model. This optimal threshold is determined through a calibration procedure performed on a dedicated held-out dataset.

Given a base segmentation model that outputs a probability map $\widehat{p}(X_i) = (\widehat{p}_1(X_i), \ldots, \widehat{p}_N(X_i))$ for an input image X_i, where $\widehat{p}_j(X_i)$ estimates $\mathbb{P}(j \in Y_i | X_i)$ for pixel j, the CRC approach proceeds as follows:

1. **Prediction Set Definition**: For a given threshold τ, the prediction set $\widehat{C}(X_i, \tau)$ is defined by including all pixels j whose predicted probability $\widehat{p}_j(X_i)$ is greater than or equal to τ:
$$\widehat{C}(X_i, \tau) = \{j : \widehat{p}_j(X_i) \geq \tau\}. \tag{4}$$
2. **Calibrated Threshold Computation**: The calibrated threshold τ' is determined using a calibration dataset I_{cal}. It is the largest threshold that satisfies the empirical risk constraint on the calibration set:
$$\tau' = \sup\left\{ \tau : \frac{1}{n+1} \sum_{i \in \mathcal{D}_{\text{cal}}} \left(1 - \frac{|\widehat{C}(X_i, \tau) \cap Y_i|}{|Y_i|} \right) + \frac{B}{n+1} \leq \alpha \right\}, \tag{5}$$
where n is the size of the calibration set $\mathcal{D}_{\text{cal}}$, B is an upper bound of the loss function (typically $B = 1$ for the FNR).
3. **Final Prediction Set**: The final prediction set for a new test image X_i is then constructed using the calibrated threshold τ':
$$\widehat{C}(X_i) = \widehat{C}(X_i, \tau'). \tag{6}$$

This methodology guarantees that $\mathbb{E}\left[1 - \frac{|\widehat{C}(X_i) \cap Y_i|}{|Y_i|}\right] \leq \alpha$ over the data distribution, providing a distribution-free control of the false negative rate, as established by Theorem 1 in Angelopoulos *et al.* (Angelopoulos et al., 2024).

3 Methodology

We introduce two novel methods for learning an image-adaptive threshold $\widehat{\tau}(X)$ for conformal risk control in image segmentation. The first, **AT** (Adaptive Thresholding), serves as a supervised baseline, while the second, **COAT** (Conditional Optimization for Adaptive Thresholding), is our more advanced, end-to-end differentiable approach.

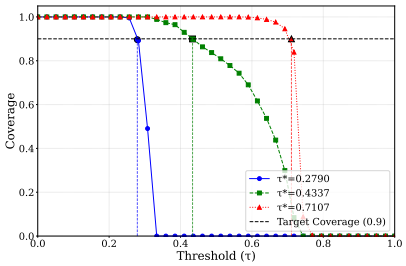

Figure 2: A figure presenting the relationship between the coverage rates of various images and the variable τ.

3.1 AT: Supervised Threshold Prediction

The AT approach frames the problem as learning a direct mapping from an image to its optimal segmentation threshold.

3.1.1 Prediction

The threshold predictor f_D takes the input image X and the corresponding probability map $\widehat{p}(X)$ from the base model to predict a single scalar threshold $\widehat{\tau}(X)$.

$$\widehat{\tau}(X) = f_D(X, \widehat{p}(X)). \tag{7}$$

The notation $(X, \widehat{p}(X))$ implies a combination of these inputs, typically through channel-wise concatenation and any necessary spatial alignment, to form the input tensor for f_D.

3.1.2 Training and Loss Function

This method requires a pre-computation step to generate a "ground-truth" threshold $\tau^*(X, Y)$ for each image in the training set. This τ^* is determined via numerical search (e.g., binary search) as the value that makes the TPR of the resulting hard segmentation mask equal to the target coverage level $1 - \alpha$.

The model f_D is then trained using a standard Mean Squared Error (MSE) loss between the predicted threshold $\widehat{\tau}(X)$ and the ground-truth threshold τ^*. The loss function over the training distribution is:

$$L_{\text{AT}} = \mathbb{E}_{(X,Y)\sim\mathcal{D}_{\text{train}}}[(\widehat{\tau}(X) - \tau^*(X, Y))^2]. \tag{8}$$

3.2 COAT: End-to-End Differentiable Miscoverage Loss

As shown in Figure 2, due to the non-continuous and non-increasing relationship between τ and the target coverage across different images, directly training τ may not necessarily yield satisfactory coverage performance and requires pre-calculating the relationship between τ and coverage. Moreover, the non-continuous and non-increasing nature also renders direct training of coverage infeasible. To circumvent the need for pre-calculating ground-truth thresholds, COAT enables end-to-end training by defining a fully differentiable loss function that directly optimizes for the target coverage.

3.2.1 Prediction

The prediction model f_D has the same architecture as in AT, and it also takes the probability map from the base model as input.

$$\widehat{\tau}(X) = f_D(X, \widehat{p}(X)). \tag{9}$$

3.2.2 Training and Loss Function

The core of this method is a differentiable approximation of the TPR. Instead of applying a hard threshold, we use the predicted threshold $\widehat{\tau}(X)$ to generate a soft, probabilistic segmentation mask

Algorithm 1 COAT: Conditional Optimization for Adaptive Thresholding

Input: labeled data $\mathcal{D}_{\text{train}}$, unlabel test data $\mathcal{D}_{\text{test}}$, target coverage $1-\alpha$, temperature T, small constant ϵ.
Randomly split $\mathcal{D}_{\text{train}}$ into $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_{\text{cal}}$.
Train a semantic segmentation model using $\mathcal{D}_1$.
Initialize parameters of threshold predictor f_D.
for epoch in training epochs **do**
  for each $(X_i, Y_i) \in \mathcal{D}_2$ **do**
    Obtain probability map $\widehat{p}_i \leftarrow \widehat{p}(X_i)$.
    Predict threshold: $\widehat{\tau}_i \leftarrow f_D(X_i, \widehat{p}(X_i))$.
    Compute soft mask: $M_{\text{soft}}(X_i) = \sigma\left(\frac{\widehat{p}(X_i) - \widehat{\tau}(X_i)}{T}\right)$.
    Compute differentiable TPR: $\widehat{\text{TPR}}(X_i, Y_i, \widehat{\tau}(X_i)) = \frac{|M_{\text{soft}}(X_i) \cdot Y_i|}{|Y_i| + \epsilon}$.
    Compute loss: $L_{\text{COAT}} \leftarrow (\widehat{\text{TPR}}(X_i, Y_i, \widehat{\tau}(X_i)) - (1-\alpha))^2$.
    Update parameters of f_D by descending the gradient ∇L_{COAT}.
  end for
end for
Compute base thresholds: $\widehat{\tau}_i \leftarrow f_D(X_i, \widehat{p}(X_i))$ for $(X_i, Y_i, \widehat{p}(X_i)) \in \mathcal{D}_{\text{cal}}$.
Define coverage function: $\mathcal{R}(t) \leftarrow \frac{1}{|\mathcal{D}_{\text{cal}}|} \sum \frac{|\{\widehat{p}(X_i) \geq \widehat{\tau}_i - t\} \cap Y_i|}{|Y_i|}$.
Find minimal correction: $t' \leftarrow \inf\{t \mid \mathcal{R}(t) \geq (|\mathcal{D}_{\text{cal}}| + 1)(1-\alpha)/|\mathcal{D}_{\text{cal}}|\}$.
for each $(X_i, \widehat{p}(X_i)) \in \mathcal{D}_{\text{test}}$ **do**
  Compute the base threshold: $\widehat{\tau}_i \leftarrow f_D(X_i, \widehat{p}(X_i))$.
  Calculate the adjusted threshold after calibration: $\tau_i' \leftarrow \text{clip}(\widehat{\tau}_i - t', 0, 1)$.
  Generate the prediction set: $\widehat{C}(X_i) \leftarrow \{\widehat{p}(X_i) \geq \tau_i'\}$.
end for
Output: $\widehat{C}(X_i)$ for $i \in \mathcal{D}_{\text{test}}$.

M_{soft}. The loss is then the MSE between the TPR calculated from this soft mask and the target coverage level $1-\alpha$.

The full loss function is defined as follows:

$$L_{\text{COAT}} = \mathop{\mathbb{E}}_{(X,Y)\sim\mathcal{D}_{\text{train}}}\left[(\widehat{\text{TPR}}(X, Y, \widehat{\tau}(X)) - (1-\alpha))^2\right],$$

where the predicted TPR, $\widehat{\text{TPR}}$, is computed via:

$$\widehat{\text{TPR}}(X, Y, \widehat{\tau}(X)) = \frac{|M_{\text{soft}}(X) \cdot Y|}{|Y| + \epsilon}.$$

Here, $Y[h, w]$ denotes the pixel value at (h, w) for the ground truth mask Y. The soft mask M_{soft} is defined using the sigmoid function $\sigma(\cdot)$ and a temperature parameter $T > 0$:

$$M_{\text{soft}}(X) = \sigma\left(\frac{\widehat{p}(X) - \widehat{\tau}(X)}{T}\right).$$

This formulation allows gradients to flow from the final loss back to the parameters of the threshold predictor f_D, enabling direct optimization towards the desired conditional coverage without intermediate supervision. As detailed in Algorithm 1, we present a comprehensive description of the COAT framework for image segmentation, covering all critical implementation components.

Remark: The primary objective of COAT is to learn the intricate relationship between an image's characteristics and its target conditional coverage. COAT achieves this through an innovative end-to-end differentiable miscoverage loss, which directly optimizes for the desired conditional coverage. This direct optimization circumvents the need to explicitly pre-calculate the complex, non-linear relationship between individual τ and coverage for each image. Following this learning phase, a calibration set is used to apply a global adjustment, t', to the predicted image-specific thresholds. This final calibration step, which can involve either a positive or negative t', statistically ensures the marginal coverage rate, as defined by Equation 2, across the entire dataset. This two-stage process—directly optimizing for image-conditional reliability and then performing a marginal calibration—provides a robust pathway towards more trustworthy and interpretable uncertainty estimates.

3.3 Theoretical Guarantees

Theorem 1 (Coverage Guarantees). *Let $\mathcal{D}_{cal} = \{(X_i, Y_i)\}_{i=1}^{n}$ be the calibration set and (X_{n+1}, Y_{n+1}) be a test sample. Suppose $\{(X_i, Y_i)\}_{i=1}^{n+1}$ are exchangeable. Then the final prediction set $\widehat{C}(X_{n+1})$ given by AT or COAT satisfies:*

$$\mathbb{E}\left[\frac{|\widehat{C}(X_{n+1}) \cap Y_{n+1}|}{|Y_{n+1}|}\right] \geq 1 - \alpha. \tag{10}$$

This theorem shows that both AT and COAT provide a powerful finite-sample guarantee for marginal coverage, i.e., the mean TPR. The proof is provided in Appendix A.1. We also show in Appendix A.2 that they can achieve asymptotic conditional validity under appropriate assumptions.

3.4 Related Work

3.4.1 Conformal Risk Control

Conformal prediction (Vovk et al., 2005) provides distribution-free uncertainty quantification with finite-sample guarantees. Its split conformal variant (Lei et al., 2018) is widely used for its computational efficiency. Recent extensions, notably Conformal Risk Control (CRC) (Angelopoulos et al., 2024; Bates et al., 2021), allow for controlling the expected value of various monotone loss functions, such as the false negative rate in segmentation.

While initial CRC applications often used a single global threshold, recent works have explored more nuanced control. Teneggi *et al.* (Teneggi et al., 2023) proposed grouping pixels to control risk through a convex surrogate loss, and further extended this to semantic-specific control for medical imaging (Teneggi et al., 2025). Bereska *et al.* (Bereska et al., 2025) introduced Spatially-Aware Conformal Prediction (SACP) to adapt uncertainty estimates based on proximity to critical structures. He *et al.* (He et al., 2025) integrated CRC into model training for quality assurance in radiation therapy. These advancements highlight a growing need for more adaptive and context-aware risk control beyond global guarantees, which our image-conditional approach directly addresses.

3.4.2 Conditional Conformal Prediction

Achieving conditional coverage – ensuring prediction sets attain the desired coverage for every possible covariate value – is generally impossible without strong distributional assumptions (Vovk, 2012; Foygel Barber et al., 2021). However, for high-stakes applications, marginal guarantees are often insufficient due to potential disparities in coverage across subpopulations.

Many works aim to improve conditional validity by modifying the calibration step (Lei & Wasserman, 2014; Guan, 2023; Barber et al., 2023) or altering the initial prediction rule (Romano et al., 2019; Sesia & Romano, 2021; Chernozhukov et al., 2021). Some research focuses on coverage under covariate shift (Lei & Wasserman, 2014; Tibshirani et al., 2019; Izbicki et al., 2022), with frameworks like that by Gibbs *et al.* (Gibbs et al., 2025) aiming for exact finite-sample coverage across shifts.

Group conditional guarantees have also been explored (Toccaceli & Gammerman, 2019; Gupta et al., 2020; Ding et al., 2023). Mondrian conformal prediction (Vovk et al., 2003) provides exact coverage for disjoint groups. More flexible methods for overlapping groups exist (Foygel Barber et al., 2021; Jung et al., 2023), though they can be computationally intensive or rely on distributional assumptions. Other approaches learn features to improve conditional coverage, such as density-based atypicality (Yuksekgonul et al., 2023) or learning covariate space partitions (Kiyani et al., 2024).

Our method contributes to this line by providing a novel way to achieve image-conditional coverage in segmentation, moving beyond group-based approaches to an instance-specific adaptive thresholding mechanism, particularly through the end-to-end differentiable optimization.

4 Experiments

This section presents the empirical evaluation of our proposed AT and COAT methods. We assess their performance in controlling the False Negative Rate (FNR) on diverse image segmentation tasks.

Our experiments aim to demonstrate the effectiveness of image-adaptive thresholding compared to approaches like CRC Angelopoulos et al. (2024) and AA-CRC (Blot et al., 2025), and to validate the robustness of our methods across different base segmentation models and datasets. Detailed experimental settings can be found in Appendixs A.3 and A.4.

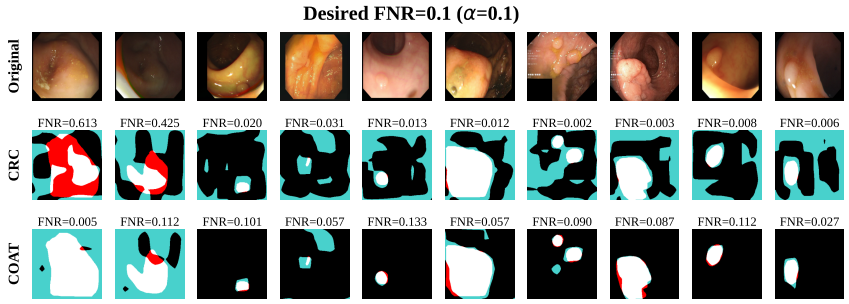

Figure 3: Qualitative comparison of CRC and COAT prediction sets at a significance level of $\alpha = 0.1$. The top row shows original polyp images, the middle row displays CRC prediction sets, and the bottom row presents COAT prediction sets. White pixels represent true positives, red false negatives, and cyan false positives. FNR values are given for each prediction. COAT demonstrates more consistent coverage and false negatives across images compared to CRC, highlighting the merit of our conditional risk control approach.

As shown in Figure 3 and Figure 4, we conducted a qualitative analysis on the polyp dataset (with alpha = 0.1) and the skin dataset (with alpha = 0.2) respectively. The base segmentation models presented in both cases are PSPNet. It can be observed that COAT is capable of better maintaining the given target coverage rate for each image. On the polyp dataset, COAT maintains a more stable FNR while also achieving a lower false positive rate. On the skin dataset, COAT consistently keeps the images at the given coverage rate. COAT adaptively adjusts thresholds to achieve the target False Negative Rate (FNR); while this may increase false positives, such instances often stem from the base model's inherent uncertainties. In safety-critical scenarios where false negatives are more detrimental than false positives, COAT's precise FNR control proves a significant advantage for robust risk management.

As shown in Table 1 and Figure 6, we randomly partitioned the dataset 20 times and then tested the mean and standard deviation of Marginal Coverage and Coverage Gap. Across all base models and datasets, COAT consistently outperformed CRC and AA-CRC in terms of Coverage Gap, with COAT consistently achieving the best Coverage Gap.

In addition, we have plotted the training progress of COAT. As can be seen from the loss function in Figure 5, regardless of the segmentation model employed and the corresponding dataset used, the training of COAT's loss function is highly stable, with a rapid decline that eventually approaches zero. The qualitative results for the Fire dataset are in Appendix A.5. We also analyzed the reference

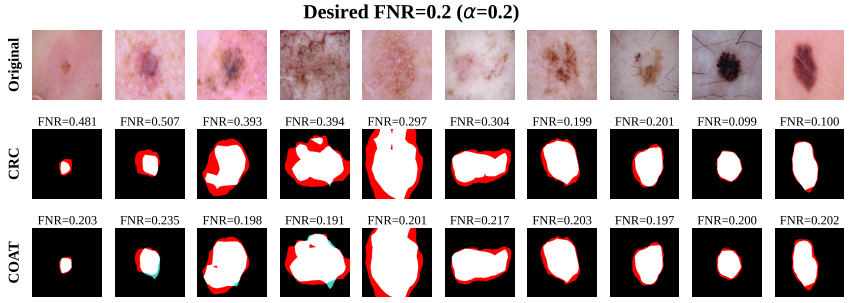

Figure 4: Qualitative comparison of CRC and COAT prediction sets at a significance level of $\alpha = 0.2$.

Dataset	Model	Method	$\alpha = 0.1$		$\alpha = 0.2$	
			Marginal Coverage	Coverage Gap	Marginal Coverage	Coverage Gap
Polyp	Deeplab v3+	CRC	0.907 (0.015)	0.145 (0.010)	0.796 (0.028)	0.232 (0.007)
		AA-CRC	0.900 (0.017)	0.122 (0.015)	0.797 (0.025)	0.167 (0.020)
		AT	0.899 (0.024)	0.115 (0.014)	0.802 (0.047)	0.174 (0.019)
		COAT	0.899 (0.016)	**0.106 (0.010)**	0.802 (0.020)	**0.162 (0.011)**
	UNet	CRC	0.901 (0.018)	0.135 (0.013)	0.797 (0.025)	0.256 (0.016)
		AA-CRC	0.908 (0.017)	0.131 (0.012)	0.793 (0.024)	0.217 (0.017)
		AT	0.903 (0.013)	0.127 (0.009)	0.805 (0.022)	0.209 (0.013)
		COAT	0.904 (0.022)	**0.122 (0.012)**	0.803 (0.018)	**0.199 (0.014)**
	PSPNet	CRC	0.906 (0.019)	0.150 (0.015)	0.804 (0.026)	0.249 (0.008)
		AA-CRC	0.908 (0.018)	0.119 (0.016)	0.796 (0.022)	0.162 (0.025)
		AT	0.899 (0.018)	0.119 (0.014)	0.796 (0.020)	0.166 (0.014)
		COAT	0.894 (0.016)	**0.110 (0.015)**	0.796 (0.021)	**0.144 (0.013)**
	SINet	CRC	0.904 (0.018)	0.149 (0.014)	0.803 (0.026)	0.255 (0.009)
		AA-CRC	0.906 (0.026)	0.126 (0.014)	0.799 (0.022)	0.182 (0.014)
		AT	0.899 (0.024)	0.119 (0.013)	0.809 (0.051)	0.170 (0.014)
		COAT	0.896 (0.016)	**0.102 (0.010)**	0.800 (0.021)	**0.148 (0.014)**
Fire	Deeplab v3+	CRC	0.901 (0.003)	0.067 (0.001)	0.803 (0.005)	0.092 (0.001)
		AA-CRC	0.903 (0.004)	0.062 (0.002)	0.804 (0.005)	0.083 (0.006)
		AT	0.901 (0.003)	0.061 (0.003)	0.806 (0.031)	0.086 (0.015)
		COAT	0.900 (0.002)	**0.058 (0.001)**	0.799 (0.003)	**0.076 (0.002)**
	UNet	CRC	0.899 (0.005)	0.077 (0.001)	0.802 (0.006)	0.103 (0.001)
		AA-CRC	0.902 (0.005)	0.068 (0.004)	0.803 (0.005)	0.088 (0.006)
		AT	0.900 (0.003)	0.063 (0.003)	0.800 (0.004)	0.085 (0.006)
		COAT	0.900 (0.003)	**0.061 (0.001)**	0.800 (0.003)	**0.079 (0.002)**
	PSPNet	CRC	0.901 (0.003)	0.065 (0.001)	0.801 (0.005)	0.091 (0.001)
		AA-CRC	0.904 (0.005)	0.063 (0.003)	0.808 (0.008)	0.079 (0.010)
		AT	0.904 (0.019)	0.065 (0.003)	0.799 (0.004)	0.089 (0.005)
		COAT	0.900 (0.003)	**0.060 (0.001)**	0.800 (0.004)	**0.077 (0.002)**
	SINet	CRC	0.901 (0.004)	0.071 (0.001)	0.800 (0.006)	0.101 (0.001)
		AA-CRC	0.903 (0.008)	0.063 (0.002)	0.803 (0.009)	0.082 (0.006)
		AT	0.900 (0.003)	0.065 (0.004)	0.799 (0.005)	0.090 (0.007)
		COAT	0.900 (0.002)	**0.059 (0.002)**	0.799 (0.004)	**0.080 (0.003)**
Skin	Deeplab v3+	CRC	0.900 (0.003)	0.072 (0.001)	0.802 (0.006)	0.107 (0.002)
		AA-CRC	0.905 (0.004)	0.057 (0.003)	0.806 (0.005)	0.079 (0.010)
		AT	0.904 (0.016)	0.061 (0.009)	0.809 (0.039)	0.090 (0.023)
		COAT	0.899 (0.003)	**0.054 (0.001)**	0.800 (0.005)	**0.073 (0.002)**
	UNet	CRC	0.900 (0.003)	0.062 (0.001)	0.800 (0.006)	0.097 (0.002)
		AA-CRC	0.908 (0.003)	0.056 (0.003)	0.807 (0.005)	0.081 (0.004)
		AT	0.899 (0.003)	0.059 (0.002)	0.800 (0.004)	0.090 (0.006)
		COAT	0.899 (0.003)	**0.054 (0.001)**	0.800 (0.004)	**0.079 (0.002)**
	PSPNet	CRC	0.902 (0.003)	0.069 (0.001)	0.804 (0.006)	0.103 (0.001)
		AA-CRC	0.906 (0.005)	0.057 (0.005)	0.806 (0.005)	0.071 (0.011)
		AT	0.903 (0.015)	0.061 (0.008)	0.809 (0.039)	0.076 (0.025)
		COAT	0.899 (0.003)	**0.050 (0.002)**	0.799 (0.004)	**0.064 (0.002)**
	SINet	CRC	0.905 (0.004)	0.078 (0.001)	0.806 (0.004)	0.113 (0.001)
		AA-CRC	0.905 (0.005)	0.063 (0.006)	0.805 (0.005)	0.075 (0.010)
		AT	0.906 (0.022)	0.065 (0.010)	0.800 (0.003)	0.074 (0.005)
		COAT	0.899 (0.003)	**0.055 (0.001)**	0.800 (0.004)	**0.071 (0.002)**

Table 1: Marginal Coverage and Coverage Gap Results at $\alpha = 0.1$ and $\alpha = 0.2$ Across Different Models and Conformal Methods. Each dataset result is the mean and standard deviation of 20 random splits.

temperature T in the COAT method, examining how different temperatures affect the coverage gap (see Appendix A.6).

Remark on Coverage Gap Limitations: It is important to note that the coverage gap cannot be reduced to zero in practice due to several fundamental limitations. First, finite sample effects mean that with limited calibration data, perfect estimation of image-specific coverage is statistically impossible. Second, there exists an inherent bias-variance trade-off in threshold prediction - while our adaptive methods reduce bias by learning image-specific patterns, they introduce variance through the learned predictor. Third, model capacity constraints limit how well our threshold predictor can capture the complex relationship between image characteristics and optimal thresholds.

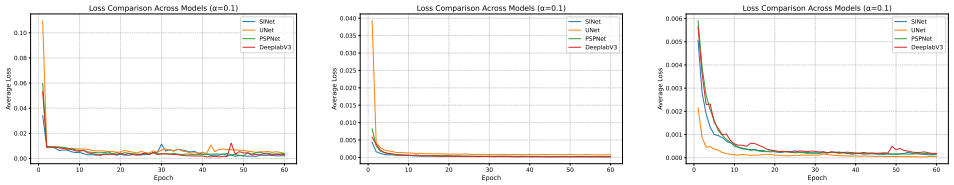

Figure 5: Loss function graphs for different segmentation models trained using the COAT method, with datasets shown from left to right being polyp, fire, and skin respectively.

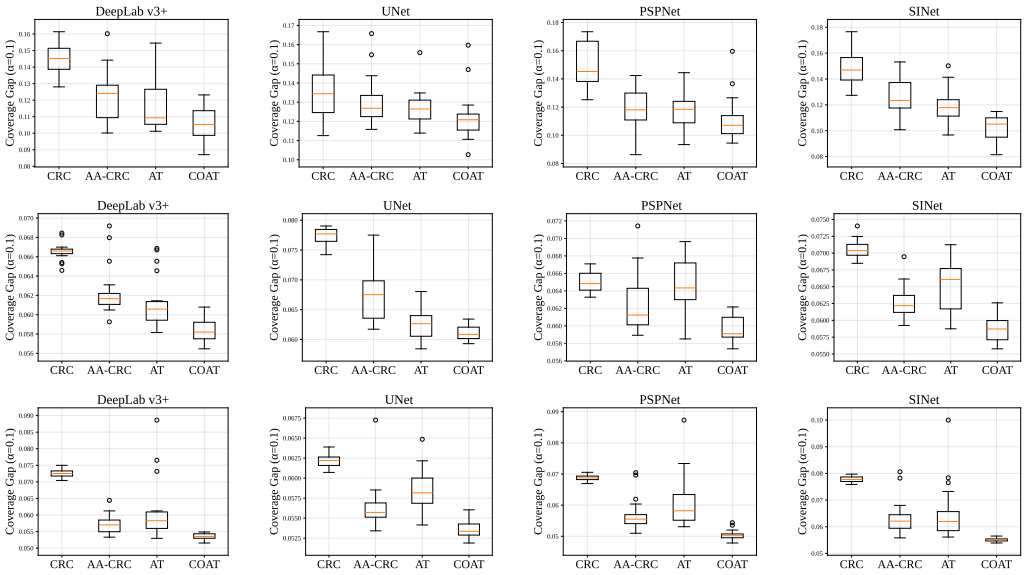

Figure 6: The box plot results for the Coverage Gap obtained from different datasets and various base segmentation models are presented. For each experimental setup, 20 random splits were conducted. The three rows of plots, from top to bottom, display the results for the polyp, fire, and skin datasets, respectively.

Despite these theoretical limitations, our COAT method consistently achieves the smallest coverage gap across all experimental settings, demonstrating its effectiveness in learning meaningful image-adaptive patterns. The end-to-end differentiable optimization in COAT provides a principled approach that directly optimizes for the target coverage, leading to more reliable conditional guarantees compared to both global thresholding (CRC) and supervised approaches (AA-CRC and AT). This improvement is particularly valuable for safety-critical applications where consistent per-image reliability is paramount.

5 Conclusion

In this paper, we address the limitation of existing segmentation methods that provide only marginal guarantees and lack image-level conditional coverage in safety-critical applications by proposing two novel methods for learning adaptive thresholds: AT and COAT. As our core contribution, COAT introduces an innovative end-to-end differentiable miscoverage loss, enabling the precise learning of an optimal threshold for each image by directly optimizing the conditional coverage target. Extensive experiments across multiple datasets demonstrate that our methods, particularly COAT, significantly reduce the Coverage Gap while maintaining the target marginal coverage rate, thereby exhibiting stronger consistency and reliability across different images. This work provides a robust pathway toward building more trustworthy and interpretable AI systems for critical applications.

REFERENCES

M. Aktaş. Fire segmentation dataset - kaggle, 2023.

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.

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.

Jacqueline Isabel Bereska, Hamed Karimi, and Reza Samavi. Sacp: Spatially-aware conformal prediction in uncertainty quantification of medical image segmentation. *Machine Learning Research*, 2025.

Jorge Bernal, Nima Tajkbaksh, Francisco Javier Sanchez, Bogdan J Matuszewski, Hao Chen, Lequan Yu, Quentin Angermann, Olivier Romain, Bjørn Rustad, Ilangko Balasingham, et al. Comparative validation of polyp detection methods in video colonoscopy: results from the miccai 2015 endoscopic vision challenge. *IEEE transactions on medical imaging*, 36(6):1231–1249, 2017.

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.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

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

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.

Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020a.

Deng-Ping Fan, Ge-Peng Ji, Tao Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. Pranet: Parallel reverse attention network for polyp segmentation. In *International conference on medical image computing and computer-assisted intervention*, pp. 263–273. Springer, 2020b.

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.

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.

Kevin He, David Adam, Sarah Han-Oh, and Anqi Liu. Training-aware risk control for intensity modulated radiation therapies quality assurance with conformal prediction. In Stefan Hegselmann, Helen Zhou, Elizabeth Healey, Trenton Chang, Caleb Ellington, Vishwali Mhasawade, Sana Tonekaboni, Peniel Argaw, and Haoran Zhang (eds.), *Proceedings of the 4th Machine Learning for Health Symposium*, volume 259 of *Proceedings of Machine Learning Research*, pp. 456–470. PMLR, 2025.

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.

Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas De Lange, Dag Johansen, and Håvard D Johansen. Kvasir-seg: A segmented polyp dataset. In *International conference on multimedia modeling*, pp. 451–462. Springer, 2019.

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 *Forty-first International Conference on Machine Learning*, 2024.

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.

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

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

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

Juan Silva, Aymeric Histace, Olivier Romain, Xavier Dray, and Bertrand Granado. Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer. *International journal of computer assisted radiology and surgery*, 9(2):283–293, 2014.

Nima Tajbakhsh, Suryakanth R Gurudu, and Jianming Liang. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE transactions on medical imaging*, 35(2):630–644, 2015.

Jacopo Teneggi, Matthew Tivnan, Web Stayman, and Jeremias Sulam. How to trust your diffusion model: A convex optimization approach to conformal risk control. In *International Conference on Machine Learning*, pp. 33940–33960. PMLR, 2023.

Jacopo Teneggi, J Webster Stayman, and Jeremias Sulam. Conformal risk control for semantic uncertainty quantification in computed tomography. *arXiv preprint arXiv:2503.00136*, 2025.

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

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

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Sci. Data*, 5:180161, 2018. doi: 10.1038/sdata.2018.161.

David Vázquez, Jorge Bernal, F Javier Sánchez, Gloria Fernández-Esparrach, Antonio M López, Adriana Romero, Michal Drozdzal, and Aaron Courville. A benchmark for endoluminal scene segmentation of colonoscopy images. *Journal of healthcare engineering*, 2017(1):4037190, 2017.

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.

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.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

A Appendix

Large Language Model (LLM) Usage

During the preparation of this manuscript, a large language model (LLM), specifically [Gemini-2.5-Flash], was employed as a general-purpose assist tool. The LLM's contributions were primarily in the following areas:

- **Code Optimization for Experimental Visualization**: The LLM assisted in optimizing and refining Python code snippets used for experimental visualization and data processing routines. This collaboration led to more efficient and readable implementations, particularly for generating the figures (e.g., Figure 3, Figure 4, Figure 7) and tables (e.g., Table 1, Table 2) presented in the paper.
- **Writing Assistance and Refinement**: The LLM was utilized for drafting and refining certain sections of the paper, including improving clarity, grammar, and stylistic coherence. This involved generating initial textual descriptions and polishing existing content to enhance its overall quality and readability.

The authors maintained full responsibility for reviewing, editing, and validating all content generated or optimized with the assistance of the LLM, ensuring its accuracy, originality, and adherence to scientific standards. The LLM was not involved in the core research ideation, experimental design, data collection, or primary analysis leading to the scientific conclusions. The scientific content, conclusions, and any potential errors remain solely the responsibility of the authors.

A.1 Proof of Theorem 1

Proof. For each sample (X_i, Y_i), we define a parameterized loss function $L_i(t)$ for $t \in [-1, 1]$, where t is a global correction parameter. The loss is the false negative rate (FNR) for the prediction set formed by the adjusted threshold:

$$L_i(t) = 1 - \frac{|\{\widehat{p}(X_i) \geq \text{clip}(\widehat{\tau}_i - t, 0, 1)\} \cap Y_i|}{|Y_i|}. \tag{11}$$

Here, $\widehat{\tau}_i = f_D(X_i, \widehat{p}(X_i))$ is the base threshold predicted by AT or COAT. This loss function $L_i(t)$ is a non-increasing and right-continuous function of t. It satisfies

$$L_i(1) = 0 \leq \alpha \quad \text{and} \quad \sup_t L_i(t) \leq 1. \tag{12}$$

We compute the empirical risk on the calibration set:

$$\bar{L}_n(t) = \frac{1}{n}\sum_{i=1}^{n} L_i(t) = 1 - \mathcal{R}(t), \tag{13}$$

where $\mathcal{R}(t)$ is the empirical coverage defined in Algorithm 1. Define

$$t' = \inf\left\{t \,\middle|\, \frac{n}{n+1}\bar{L}_n(t) + \frac{1}{n+1} \le \alpha\right\} = \inf\left\{t \,\middle|\, \mathcal{R}(t) \ge \frac{n+1}{n}(1-\alpha)\right\}. \tag{14}$$

By Theorem 1 in Angelopoulos et al. (2024), we have $\mathbb{E}\left[L_{n+1}(t')\right] \le \alpha$, which implies

$$\mathbb{E}\left[\frac{|\widehat{C}(X_{n+1}) \cap Y_{n+1}|}{|Y_{n+1}|}\right] \ge 1 - \alpha \tag{15}$$

as $\widehat{C}(X_{n+1}) = \{\widehat{p}(X_{n+1}) \ge \text{clip}(\widehat{\tau}_{n+1} - t', 0, 1)\}$. □

A.2 Asymptotic Validity of Conditional Coverage

In this section, we formally establish the asymptotic validity of the conditional coverage provided by our proposed methods, AT and COAT. Specifically, we will demonstrate that as the sample size used to train the base threshold models tends to infinity, the conditional coverage rate (i.e., the TPR of prediction set for each test sample) is asymptotically guaranteed to be not less than the target level $1-\alpha$ under appropriate assumptions. To this end, we begin by introducing the necessary notations.

For any (X, Y), let $\widehat{\tau}_m(X)$ be the base threshold estimated by AT or COAT, where m is the sample size of the dataset used to train $\widehat{\tau}_m$. Let $\mathcal{D}_{\text{cal}} = \{(X_i, Y_i)\}_{i=1}^n$ be the calibration set and (X_{n+1}, Y_{n+1}) be a test sample. Similar to A.1, we define a FNR function for (X_i, Y_i):

$$L_{i,m}(t) = 1 - \frac{|\{\widehat{p}(X_i) \ge \text{clip}(\widehat{\tau}_m(X_i) - t, 0, 1)\} \cap Y_i|}{|Y_i|}, \quad t \in [-1, 1]. \tag{16}$$

The empirical risk on the calibration set is:

$$\bar{L}_{n,m}(t) = \frac{1}{n}\sum_{i=1}^{n} L_{i,m}(t). \tag{17}$$

By Algorithm 1, the correction term for (X_{n+1}, Y_{n+1}) is computed as:

$$t'_m = \inf\left\{t \,\middle|\, \bar{L}_{n,m}(t) \le C_n(\alpha) := \frac{n+1}{n}\alpha - \frac{1}{n}\right\}. \tag{18}$$

Suppose the ground-truth threshold function is τ^*. Then for $t \in [-1, 1]$, we can define:

$$L_i^*(t) = 1 - \frac{|\{\widehat{p}(X_i) \ge \text{clip}(\tau^*(X_i) - t, 0, 1)\} \cap Y_i|}{|Y_i|}, \tag{19}$$

$$\bar{L}_n^*(t) = \frac{1}{n}\sum_{i=1}^{n} L_i^*(t), \tag{20}$$

$$t^* = \inf\left\{t \mid \bar{L}_n^*(t) \le C_n(\alpha)\right\}. \tag{21}$$

The prediction set for (X_{n+1}, Y_{n+1}) given by AT or COAT can be expressed as:

$$\widehat{C}(X_{n+1}) = \{\widehat{p}(X_{n+1}) \ge \text{clip}(\widehat{\tau}_m(X_{n+1}) - t'_m, 0, 1)\}. \tag{22}$$

The result is summarized by the following theorem.

Theorem 2 (Asymptotic Conditional Validity)**.** *Assume that for any (X, Y), as $m \to \infty$, $\widehat{\tau}_m(X) \xrightarrow{p} \tau^*(X, Y)$ and*

$$|\{\widehat{p}(X) < \tau^*(X, Y)\} \cap Y|/|Y| = \alpha \quad \textit{almost surely}. \tag{23}$$

If $\{(X_i, Y_i)\}_{i=1}^{n+1}$ are i.i.d. and $\widehat{C}(X_{n+1})$ is the prediction set given by AT or COAT, then for any $\varepsilon > 0$, as $m \to \infty$,

$$\mathbb{P}\left(\frac{|\widehat{C}(X_{n+1}) \cap Y_{n+1}|}{|Y_{n+1}|} \;\ge\; 1 - \alpha - \varepsilon\right) \;\to\; 1. \tag{24}$$

Proof. By assumption, for any fixed t, the consistency $\widehat{\tau}_m \xrightarrow{p} \tau^*$ and the Continuous Mapping Theorem imply $L_{i,m}(t) \xrightarrow{p} L_i^*(t)$ and thus $\bar{L}_{n,m}(t) \xrightarrow{p} \bar{L}_n^*(t)$ as $m \to \infty$. Since we assume that for any (X, Y),

$$|\{\widehat{p}(X) < \tau^*(X, Y)\} \cap Y|/|Y| = \alpha \quad a.s., \tag{25}$$

by definitions we have

$$L_i^*(0) = \bar{L}_n^*(0) = \alpha \quad a.s.. \tag{26}$$

As $\bar{L}_n^*(t)$ is non-increasing, $\bar{L}_n^*(1) = 0$, and $C_n(\alpha) = (n+1)\alpha/n - 1/n < \alpha$,

$$\left\{t \mid \bar{L}_n^*(t) \le C_n(\alpha)\right\} \neq \varnothing \quad \text{and} \quad t^* \ge 0 \quad a.s.. \tag{27}$$

$\bar{L}_n^*(t)$ is a non-increasing, right-continuous step function with finite steps determined by the size of calibration set $\mathcal{D}_{\text{cal}} = \{(X_i, Y_i)\}_{i=1}^n$, which implies $C_n(\alpha)$ is not one of the jump values of $\bar{L}_n^*(t)$ almost surely. Therefore, there exists an $\eta > 0$ such that for any small $\delta > 0$,

$$\bar{L}_n^*(t^* - \delta) \ge C_n(\alpha) + \eta \quad \text{and} \quad \bar{L}_n^*(t^* + \delta) \le C_n(\alpha) - \eta \quad a.s.. \tag{28}$$

From the pointwise convergence $\bar{L}_{n,m}(t) \xrightarrow{p} \bar{L}_n^*(t)$, we have:

$$\mathbb{P}\left(\bar{L}_{n,m}(t^* - \delta) > C_n(\alpha)\right) \to 1 \quad \text{and} \quad \mathbb{P}\left(\bar{L}_{n,m}(t^* + \delta) < C_n(\alpha)\right) \to 1. \tag{29}$$

Since $t'_m = \inf\left\{t \mid \bar{L}_{n,m}(t) \le C_n(\alpha)\right\}$ and $\bar{L}_{n,m}(t)$ is non-increasing, we have:

$$\{\bar{L}_{n,m}(t^* - \delta) > C_n(\alpha)\} \cap \{\bar{L}_{n,m}(t^* + \delta) < C_n(\alpha)\} \implies t'_m \in [t^* - \delta,\, t^* + \delta]. \tag{30}$$

Hence,

$$\mathbb{P}\left(|t'_m - t^*| \le \delta\right) \to 1, \quad \text{i.e., } t'_m \xrightarrow{p} t^*. \tag{31}$$

For the test sample (X_{n+1}, Y_{n+1}), since $L_{n+1,m}(t)$ is non-increasing, it holds that

$$L_{n+1,m}(t^* + \delta) \le L_{n+1,m}(t'_m) \le L_{n+1,m}(t^* - \delta) \tag{32}$$

when $|t'_m - t^*| \le \delta$. As $m \to \infty$, the lower and upper bounds converge in probability:

$$L_{n+1,m}(t^* + \delta) \xrightarrow{p} L_{n+1}^*(t^* + \delta), \tag{33}$$

$$L_{n+1,m}(t^* - \delta) \xrightarrow{p} L_{n+1}^*(t^* - \delta). \tag{34}$$

Recall the definition $t^* = \inf\left\{t \mid \bar{L}_n^*(t) \le C_n(\alpha)\right\}$. Since $\{(X_i, Y_i)\}_{i=1}^{n+1}$ are i.i.d., the random variable t^* is independent of the finite jump points of $L_{n+1}^*(t)$. Thus, $L_{n+1}^*(t)$ is continuous at t^* almost surely. Then, letting $\delta \to 0$, we obtain:

$$L_{n+1}^*(t^* + \delta) \to L_{n+1}^*(t^*) \quad \text{and} \quad L_{n+1}^*(t^* - \delta) \to L_{n+1}^*(t^*) \quad a.s., \tag{35}$$

which implies

$$L_{n+1,m}(t'_m) \xrightarrow{p} L_{n+1}^*(t^*). \tag{36}$$

Since

$$L_{n+1}^*(t^*) \le L_{n+1}^*(0) = \alpha \quad a.s., \tag{37}$$

for any $\varepsilon > 0$ we have $\mathbb{P}\left(L_{n+1,m}(t'_m) \le \alpha + \varepsilon\right) \to 1$. Equivalently,

$$\mathbb{P}\left(\frac{|\widehat{C}(X_{n+1}) \cap Y_{n+1}|}{|Y_{n+1}|} \;\ge\; 1 - \alpha - \varepsilon\right) \;\to\; 1. \tag{38}$$

□

A.3 DATASETS

We utilized three distinct image segmentation datasets to evaluate the robustness of our proposed algorithms: polyp segmentation, skin lesion segmentation, and flame segmentation. These tasks are particularly critical for FNR control, as missing parts of the region of interest can lead to severe consequences.

For each dataset, we followed a specific data partitioning strategy. The initial training set was used for training the base segmentation models (UNet (Ronneberger et al., 2015), DeepLab v3+ (Chen et al., 2018), PSPNet (Zhao et al., 2017), SINet (Fan et al., 2020a)). The remaining data was designated as the test set. This test set was then further partitioned. One half of the test set was reserved for final performance evaluation. The other half was designated as a calibration set. For AA-CRC, AT and COAT methods, this calibration set was further equally divided into a training subset (for training the adaptive threshold prediction model f_D) and a calibration subset (for determining the final calibrated threshold). For the standard Conformal Risk Control (CRC) method, the entire calibration set was utilized for its calibration procedure.

- **Polyp Dataset**: Following similar setups as Angelopoulos *et al.* (Angelopoulos et al., 2024), blot *et al.* (Blot et al., 2025) and Fan *et al.* (Fan et al., 2020b), this dataset (Jha et al., 2019; Bernal et al., 2017; Vázquez et al., 2017; Tajbakhsh et al., 2015; Silva et al., 2014) comprised 1450 images for training the base segmentation models and 798 images for the test set.
- **Skin Lesion Dataset**: We employed the HAM10000 skin image dataset (Tschandl et al., 2018). This dataset was split with 50% (5007 images) allocated for training the base models and the remaining 50% (5008 images) for the test set.
- **Fire Dataset**: For image fire segmentation experiments, we used the dataset provided by Aktaş (Aktaş, 2023). This dataset was partitioned with 80% (21968 images) for training the base models and 20% (5492 images) for the test set.

Additionally, for both the AT and COAT methods, the threshold predictor f_D was implemented using a ResNet-50 (He et al., 2016) architecture. The training epochs for AT were set to 30, while for COAT, they were set to 60. For the COAT method, the temperature parameter T used in the sigmoid function for the soft mask calculation was set to 0.05.

A.4 Experimental details

Reproducibility Statement: The source code and experiment scripts used to generate the results in this paper will be made publicly available upon publication of the paper.

Implementation Details: All experiments were conducted on a server equipped with an NVIDIA RTX 4090 GPU (24GB of RAM), running Ubuntu 24.04. Our models were implemented using Python 3.10, PyTorch 2.3.0, and the system was configured with CUDA 12.6. The learning rate of the COAT method is consistently set to $5e^{-4}$, and the batch size is 64. All the basic segmentation models underwent 20 epochs of training. Additionally, a unified learning rate of $1e^{-4}$ and a batch size of 24 were employed for all models. The random seed was set to 42 for all experiments.

A.5 Fire results

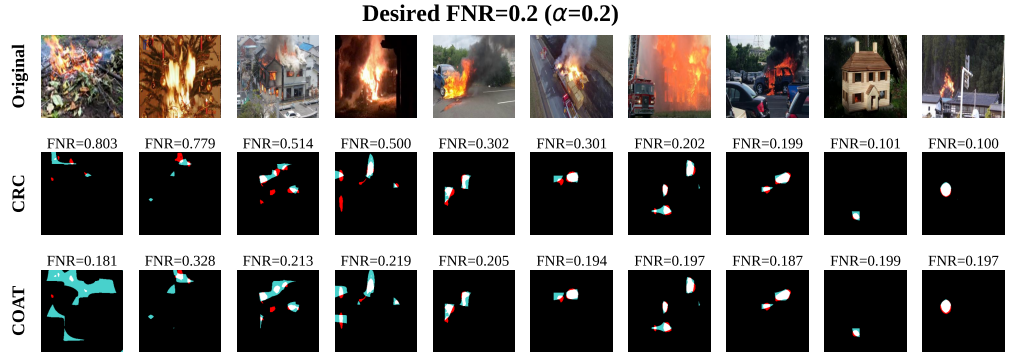

Figure 7: Qualitative comparison of CRC and COAT prediction sets at a significance level of $\alpha = 0.2$.

Due to space constraints, we have included the qualitative analysis of the fire dataset in the appendix. It is evident that COAT consistently maintains target coverage, albeit with a certain degree of false

positive rate. However, when compared to CRC, COAT demonstrates a lower and more stable false negative rate.

A.6 Sensitivity Analysis

In this section, we employed the SINet basic segmentation model on various datasets and conducted a grid search sensitivity analysis with parameters set as $\alpha = 0.1$ and $T = [0.001, 0.01, 0.05, 0.1, 1, 10, 100]$. Similarly, each experiment was carried out with 20 random splits, and the results are presented as the mean values and standard deviations.

Dataset	T	$\alpha = 0.1$	
		Marginal Coverage	Coverage Gap
Polyp	100	0.901 (0.016)	0.151 (0.012)
	10	0.899 (0.023)	0.153 (0.018)
	1	0.899 (0.023)	0.156 (0.017)
	0.1	0.899 (0.012)	0.114 (0.015)
	0.05	0.896 (0.016)	**0.102 (0.010)**
	0.01	0.900 (0.020)	0.148 (0.013)
	0.001	0.901 (0.026)	0.147 (0.018)
Fire	100	0.900 (0.003)	0.071 (0.001)
	10	0.901 (0.003)	0.070 (0.001)
	1	0.900 (0.003)	0.070 (0.001)
	0.1	0.900 (0.003)	0.062 (0.001)
	0.05	0.900 (0.003)	**0.059 (0.001)**
	0.01	0.900 (0.004)	0.060 (0.001)
	0.001	0.900 (0.003)	0.066 (0.014)
Skin	100	0.900 (0.001)	0.081 (0.001)
	10	0.901 (0.001)	0.081 (0.001)
	1	0.900 (0.002)	0.081 (0.001)
	0.1	0.901 (0.004)	0.056 (0.001)
	0.05	0.899 (0.003)	**0.055 (0.001)**
	0.01	0.900 (0.004)	0.056 (0.002)
	0.001	0.899 (0.003)	0.058 (0.009)

Table 2: Marginal Coverage and Coverage Gap Results at $\alpha = 0.1$ Across Different Models and T. Each dataset result is the mean and standard deviation of 20 random splits.

A.7 COAT Core Code Implementation

To enhance the reproducibility of our work, this appendix provides the core PyTorch implementation details for our COAT (Conditional Optimization for Adaptive Thresholding) method. We have split the implementation into two parts, mirroring the two main stages of the framework: the end-to-end training of the threshold predictor, and the subsequent post-hoc calibration and inference process.

A.7.1 Algorithm for COAT Training Phase

This first algorithm details the training procedure for the adaptive threshold predictor, f_D. The core components are the network architecture itself and the novel differentiable miscoverage loss function, L_{COAT}, which enables direct, gradient-based optimization towards the conditional coverage target.

Algorithm 2 Python Pseudo-Code for COAT: Training Phase. This code outlines the end-to-end training of the `ThresholdPredictor` network. It includes the network architecture, the differentiable TPR approximation (`calculate_differentiable_tpr`), and the main training loss computation within a single training step function (`coat_training_step`).

```python
import torch
import torch.nn as nn
from torchvision.models import resnet50, ResNet50_Weights

class ThresholdPredictor(nn.Module):
    """The adaptive threshold predictor network, f_D in the paper."""
    def __init__(self, pretrained: bool = True):
        super().__init__()
        self.resnet = resnet50(weights=ResNet50_Weights.DEFAULT)
        # Modify input layer for 4 channels (Image RGB + Probability Map)
        original_conv1 = self.resnet.conv1
        self.resnet.conv1 = nn.Conv2d(4, 64, kernel_size=7, stride=2, padding=3,bias=False)
        with torch.no_grad():
            self.resnet.conv1.weight[:, :3, :, :] = original_conv1.weight.clone()
            self.resnet.conv1.weight[:, 3, :, :] = original_conv1.weight.clone().mean(dim=1,
            keepdim=True)
        # Modify output layer for a single threshold value
        self.resnet.fc = nn.Linear(self.resnet.fc.in_features, 1)

    def forward(self, image: torch.Tensor, prob_map: torch.Tensor) -> torch.Tensor:
        input_tensor = torch.cat([image, prob_map.unsqueeze(1)], dim=1)
        return torch.sigmoid(self.resnet(input_tensor))

def calculate_differentiable_tpr(
    prob_map: torch.Tensor, true_mask: torch.Tensor, pred_tau: torch.Tensor,
    T: float, epsilon: float) -> torch.Tensor:
    """Computes the differentiable TPR'."""
    soft_mask = torch.sigmoid((prob_map - pred_tau) / T)
    intersection = torch.sum(soft_mask * true_mask)
    true_mask_size = torch.sum(true_mask)
    return intersection / (true_mask_size + epsilon)

def coat_training_step(
    model: ThresholdPredictor, optimizer: torch.optim.Optimizer,
    image_batch: torch.Tensor, prob_map_batch: torch.Tensor, true_mask_batch: torch.Tensor,
    alpha: float, T: float, epsilon: float):
    """Performs a single training step for the COAT model."""
    model.train()
    optimizer.zero_grad()

    # Predict image-specific thresholds
    pred_taus = model(image_batch, prob_map_batch)

    total_loss = 0.0
    target_coverage = 1 - alpha
    # Compute loss for each sample in the batch
    for i in range(image_batch.size(0)):
        tpr_prime = calculate_differentiable_tpr(
            prob_map_batch[i], true_mask_batch[i], pred_taus[i], T, epsilon)
        # L_COAT = (TPR' - (1 - alpha))^2
        loss = (tpr_prime - target_coverage)**2
        total_loss += loss

    # Update parameters
    avg_loss = total_loss / image_batch.size(0)
    avg_loss.backward()
    optimizer.step()
    return avg_loss.item()
```

A.7.2 Algorithm for COAT Calibration and Inference Phase

Once the 'ThresholdPredictor' is trained, this second algorithm details the procedure for calibration and inference. A held-out calibration set, $\mathcal{D}_{cal}$, is used to compute a single, global correction term t'. This correction is then applied to the predicted thresholds for all test images to generate the final prediction sets, $C(X)$, which are guaranteed to satisfy the marginal coverage property.

Algorithm 3 Python Pseudo-Code for COAT: Calibration and Inference Phase. This code shows the post-training procedure. First, `calibrate_coat` uses the trained model and a calibration set to find the optimal correction term `t_prime`. Then, `apply_calibrated_coat` uses this term to generate final prediction sets on the test data.

```python
import torch
import numpy as np

def calibrate_coat(
    model: torch.nn.Module, cal_loader: torch.utils.data.DataLoader, alpha: float) -> float:
    """Performs calibration to find the correction term t'."""
    model.eval()
    base_taus_cal = []

    # Compute base thresholds for the calibration set D_cal
    with torch.no_grad():
        for (images, prob_maps, _) in cal_loader:
            base_taus_cal.extend(model(images, prob_maps).cpu().numpy())

    # Define the empirical coverage function R(t)
    def get_empirical_coverage(t: float) -> float:
        coverages = []
        cal_dataset = cal_loader.dataset
        for i in range(len(cal_dataset)):
            _, prob_map, true_mask = cal_dataset[i]
            # Apply correction: tau_i - t
            adjusted_tau = np.clip(base_taus_cal[i] - t, 0, 1)
            pred_set = prob_map >= adjusted_tau
            if true_mask.sum() > 0:
                coverage = (pred_set & true_mask).sum() / true_mask.sum()
                coverages.append(coverage.item())
        return np.mean(coverages) if coverages else 0.0

    # Find minimal correction t' via search (e.g., binary search)
    # Target is R(t) >= (|D_cal|+1)/|D_cal| * (1-alpha)
    n = len(cal_loader.dataset)
    target_cal_coverage = (n + 1) / n * (1 - alpha)
    # A placeholder for a binary search function to find t'
    t_prime = binary_search_for_t(get_empirical_coverage, target_cal_coverage)

    return t_prime

def apply_calibrated_coat(
    model: torch.nn.Module, test_loader: torch.utils.data.DataLoader, t_prime: float) -> list:
    """Applies the calibrated model to the test set."""
    model.eval()
    prediction_sets = []
    with torch.no_grad():
        for (images, prob_maps, _) in test_loader:
            # Compute base thresholds for test images
            base_taus_test = model(images, prob_maps)
            for i in range(len(images)):
                # Calculate the adjusted threshold
                adjusted_tau = torch.clamp(base_taus_test[i] - t_prime, 0, 1)
                # Generate the final prediction set
                pred_set = prob_maps[i] >= adjusted_tau
                prediction_sets.append(pred_set)

    return prediction_sets # Line 23: Output C(X_i)
```