Automatically Adaptive Conformal Risk Control

Vincent Blot
Paris-Saclay University, CNRS,
Laboratoire Interdisciplinaire des Sciences du Numerique,
91405, Orsay, France,
Capgemini Invent France

Anastasios N. Angelopoulos
University of California, Berkeley

Michael I. Jordan
University of California, Berkeley,
INRIA Paris

Nicolas J-B. Brunel
ENSIIE,
1 Square de la Resistance,
91000, Evry-Courcouronnes, France,
Capgemini Invent France

Abstract

Science and technology have a growing need for effective mechanisms that ensure reliable, controlled performance from black-box machine learning algorithms. These performance guarantees should ideally hold *conditionally on the input*—that is the performance guarantees should hold, at least approximately, no matter what the input. However, beyond stylized discrete groupings such as ethnicity and gender, the right notion of conditioning can be difficult to define. For example, in problems such as image segmentation, we want the uncertainty to reflect the intrinsic difficulty of the test sample, but this may be difficult to capture via a conditioning event. Building on the recent work of Gibbs et al. (2023), we propose a methodology for achieving approximate conditional control of statistical risks—the expected value of loss functions—by adapting to the difficulty of test samples. Our framework goes beyond traditional conditional risk control based on user-provided conditioning events to the algorithmic, data-driven determination of appropriate function classes for conditioning. We apply this framework to various regression and segmentation tasks, enabling finer-grained control over model performance and demonstrating that by continuously monitoring and adjusting these parameters, we can achieve superior precision compared to conventional risk-control methods.

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

1 INTRODUCTION

Conformal prediction (Vovk et al., 2005) has emerged over the last several years as a promising solution for quantifying uncertainty in black-box machine learning models via prediction sets. Conformal risk control (Angelopoulos et al., 2024) extends the conformal methodology to high-dimensional and structured data tasks, such as image segmentation, where the standard notion of coverage does not naturally apply. These techniques are especially attractive due to their model- and distribution-agnostic nature; their validity does not rely on any assumptions about the model class at hand or the particular data distribution (Vovk et al., 2005). A limitation of classical conformal techniques, however, is its inability to provide conditional guarantees. Thus, the quality of the uncertainty quantification can depend on the input covariates and degrade in some parts of the input space, especially where data is scarce, even if the average quality of uncertainty quantification is controlled.

While conditional guarantees are impossible in full generality for any algorithm (Vovk, 2012), recent progress has been made on tractable relaxations of conditional coverage. In particular, Gibbs et al. (2023) introduce an extension of conformal prediction that gives exact coverage conditionally on overlapping groups, and additionally, can provide a relaxed form of conditional coverage against certain covariate shifts parameterized

by a user-chosen function class $\mathcal{F}$. For a non-expert user, however, specifying $\mathcal{F}$ can be hard, and in many prediction tasks, there is no clear choice even for the expert user. Indeed, there are many tasks for which users do not have any conditioning events in mind, but rather, simply want their uncertainty to adapt automatically to the difficulty of the test sample.

In this paper, we introduce a procedure— *automatically adaptive conformal risk control (AA-CRC)*—that involves two innovations: (1) it obviates the need to pick a function class $\mathcal{F}$ in Gibbs et al. (2023) by providing a theoretically motivated algorithm for selection of $\mathcal{F}$, and (2) it extends the arguments of Gibbs et al. (2023) for conformal prediction to conformal risk control. We also extend Gibbs et al. (2023) to handle label-conditional coverage. Auto-adaptive CRC thus adapts more carefully to the difficulty of the input sample, and the resulting uncertainty better reflects the true errors of the model. As an important practical side effect, AA-CRC generally has substantially better statistical power than conformal prediction or conformal risk control alone. For an example of this improved performance, see Figure 1. The code is available at `https://github.com/vincentblot28/multiaccurate-cp` and all datasets used for the experiments are open source.

1.1 Problem statement

Consider a dataset of exchangeable feature-label pairs, $(X_1, Y_1), \ldots, (X_{n+1}, Y_{n+1}) \in \mathcal{X} \times \mathcal{Y}$, where the last label Y_{n+1} is our target (an unknown quantity we want to predict). Consider a set-valued predictor $\mathcal{C}_\lambda(x)$, indexed by λ. We would like this set to have a low risk—or expected loss—as measured by a loss function $\ell(\mathcal{C}_\lambda(x), y)$. An example of a loss function is the false negative rate in multilabel classification: $\ell(\mathcal{C}_\lambda(x), y) = \frac{|y \setminus \mathcal{C}_\lambda(x)|}{|y|}$. Conformal risk control, as defined in Angelopoulos et al. (2024), offers guarantees of the form

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

provided that ℓ is monotone nonincreasing when viewed as a function of λ. The goal of our work is to extend the above guarantee analogously to (2.3) of Gibbs et al. (2023):

$$\mathbb{E}\left[\frac{\lambda(X_{n+1})}{\mathbb{E}[\lambda(X_{n+1})]}\left(\ell(\mathcal{C}_{\hat{\lambda}(X_{n+1})}(X_{n+1}), Y_{n+1}) - \alpha\right)\right] \leq 0, \tag{2}$$

for any non-negative $\lambda \in \Lambda$, where Λ is some class of functions that map $\mathcal{X}$ to $\mathbb{R}$. Following Gibbs et al. (2023), the choice of function class Λ will determine the type of multiaccuracy guarantees we are able to achieve.

To better understand the guarantee in (2), we give several examples.

1. When $\Lambda = \{x \mapsto 1\}$, we recover standard, marginal conformal risk control, and (2) becomes equivalent to (1).

2. Let Φ map $\mathcal{X}$ to a d-dimensional binary vector. One can think of $\Phi(x)$ as a vector of group indicators. When $\Lambda = \{\Phi(x)^\top \theta : \theta \in \mathbb{R}^d\}$, we obtain group-conditional conformal risk control with overlapping groups:

$$\mathbb{E}\left[\ell(\mathcal{C}_{\hat{\lambda}(X_{n+1})}(X_{n+1}), Y_{n+1}) \mid \Phi(X_{n+1}) = j\right] \leq \alpha, \quad \forall j \in [d]. \tag{3}$$

3. Let Φ be a d-dimensional neural network embedding of X, and let $\Lambda = \{\Phi(x)^\top \theta : \theta \in \mathbb{R}^d\}$. Then our method provides a risk control guarantee over a set of covariate shifts:

$$\mathbb{E}_\lambda\left[\ell(\mathcal{C}_{\hat{\lambda}(X_{n+1})}(X_{n+1}), Y_{n+1})\right] \leq \alpha, \qquad \forall \lambda \in \Lambda, \tag{4}$$

where $\mathbb{E}_\lambda$ is defined as the expected value when tilted by $\frac{\lambda(X_{n+1})}{\mathbb{E}[\lambda(X_{n+1})]}$, as in (2). In other words, our guarantee is robust to all covariate shifts that are linear in embedding space. It is by learning Φ to provide features that result in a high-accuracy linear predictor of the error on X_{n+1} that we achieve our "automatic" notion of conditional validity.

Related work

We study the topic of conformal prediction (Vovk et al., 2005) under relaxed notions of conditional coverage. There is a large volume of work on conformal prediction and conditional coverage, most notably the foundational works of Vovk (2012), Barber et al. (2021). Also foundational are the works of Jung et al. (2021, 2022); Bastani et al. (2022), which introduce the relationship between conformal prediction, quantile regression, and conditional coverage on discrete but overlapping groups. These papers introduce the idea in a high-probability, split-conformal sense; it is developed by Gibbs et al. (2023) in a full-conformal, marginal sense over possibly continuous groups, as in the present manuscript. We remark that the guarantee in (2) resembles the multi-accuracy guarantee in (1) of Kim et al. (2019), although the mathematical tools we use are unrelated, as far as we know. The closest ancestors of our work are Gibbs et al. (2023) and Angelopoulos et al. (2024). Our paper combines the guarantees from these two lines of work. As we will soon see, combining these approaches is not trivial, and stems from a new reframing of conformal risk control as the solution to an

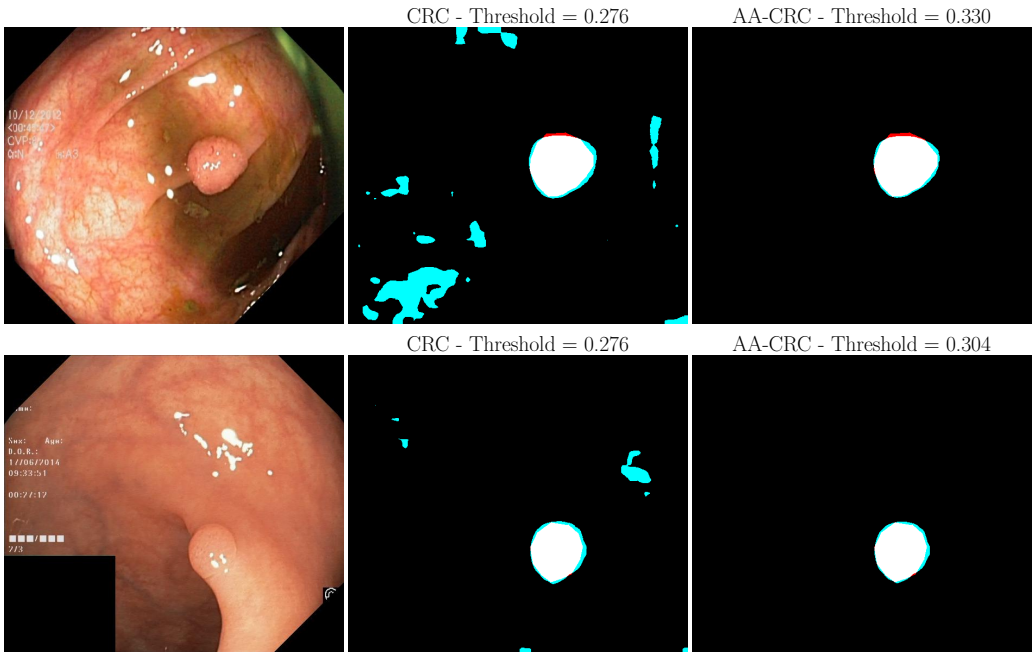

Figure 1: Example of polyp segmentations with conformal risk control (CRC) and our methodology (AA-CRC), where the true positive pixels are in white and the false positives in blue. To guarantee the recall on the image, our method outputs a threshold equal to 0.304 and 0.330 while the constant threshold of the CRC methodology is 0.276. This difference implies a higher precision for our methodology.

implicit optimization problem. An additional novelty as compared to Gibbs et al. (2023) is suggesting an automatic algorithm for selecting the function class Λ in order to achieve better general purpose conditional performance—this is critical, as there is no clear choice of Λ in many practical problems, so this improves upon the practical value of Gibbs et al. (2023) even in the standard conformal setup. Along the same lines, we extend the guarantee of Gibbs et al. (2023) to handle label-conditional coverage, and more generally allow $\mathcal{F}$ to be a class of mappings that depend on both the covariate and the label. Finally, we refer the reader to the related concurrent work of Zhang et al. (2024) on fair risk control; their problem setting is similar to ours, while their algorithms and guarantees are different but complementary to ours.

2 THEORY

2.1 Background

We begin by reinterpreting conformal prediction in the language of the first-order optimality conditions of standard quantile regression (Koenker and Bassett Jr, 1978). Let $D^y = ((X_1, Y_1), \ldots, (X_n, Y_n), (X_{n+1}, y))$ denote a putative dataset where the $(n+1)$st label is replaced with the putative label y. Let $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a conformal score. Also, let $\mathcal{C}_\lambda(x) = \{y : s(x, y) \leq \lambda\}$ be the split-conformal prediction set formed using threshold λ. Finally, let $\rho(u) = (1-\alpha)u\mathbb{1}\{u \geq 0\} - \alpha u\mathbb{1}\{u < 0\}$ be the pinball loss.

The first step to understanding our approach is to reframe conformal prediction as a form of intercept-only quantile regression. Let

$$J(\lambda, D^y) = \sum_{i=1}^{n} \rho(s(X_i, Y_i) - \lambda) + \rho(s(X_{n+1}, y) - \lambda), \tag{5}$$

and let $\hat{\lambda}^y = \text{argmin}_{\lambda \in \Lambda} J(\lambda, D^y)$. As in Gibbs et al. (2023), it is straightforward to verify that the standard split-conformal prediction set is formed as

$$\mathcal{C}(X_{n+1}) = \{y : s(X_{n+1}, y) \leq \hat{\lambda}^y\}. \tag{6}$$

In addition to the procedure being equivalent to a form of quantile regression, the coverage guarantee can be rephrased in the language of the first-order conditions of quantile regression as well. As in any optimization problem, the first-order optimality condition states that $0 \in \partial J(\hat{\lambda}^y, D^y)$. Accordingly, for all $i \in [n+1]$, we define g_i to be the subgradient function of $\lambda \mapsto \rho(s(X_i, Y_i) - \lambda)$ characterized as follows:

1. $g_i(\lambda) = \mathbb{1}\{Y_i \notin \mathcal{C}_\lambda(X_i)\} - \alpha$ if $\lambda \neq s(X_i, Y_i)$.

2. $g_{n+1}(\lambda) = \mathbb{1}\{y \notin \mathcal{C}_\lambda(X_{n+1})\} - \alpha$ if $\lambda \neq s(X_i, y)$.
3. $g_i(\lambda) \in [-\alpha, 1-\alpha]$.
4. $\sum_{i=1}^{n+1} g_i(\hat{\lambda}^y) = 0$.

Using these constraints, and defining $\mathcal{I} = \{i : \hat{\lambda} = s(X_i, y)\}$, we have:

$$\sum_{i=1}^{n+1} g_i(\hat{\lambda}^{Y_{n+1}}) = 0$$
$$\iff \frac{1}{n+1}\sum_{i=1}^{n+1} \mathbb{1}\left\{Y_i \notin \mathcal{C}_{\hat{\lambda}}(X_i)\right\} - \alpha = \frac{1}{n+1}\sum_{i\in\mathcal{I}}(\mathbb{1}\left\{Y_i \notin \mathcal{C}_{\hat{\lambda}}(X_i)\right\} - \alpha - g_i(\hat{\lambda}^{Y_{n+1}})). \tag{7}$$

Note that for all $i \in \mathcal{I}$, $\mathbb{1}\left\{Y_i \notin \mathcal{C}_{\hat{\lambda}}(X_i)\right\} = 0$, and $g_i(\hat{\lambda}^{Y_{n+1}}) \geq -\alpha$. Thus, the right-hand side of the displayed equation is nonpositive, implying that $\frac{1}{n+1}\sum_{i=1}^{n+1} \mathbb{1}\left\{Y_i \notin \mathcal{C}_{\hat{\lambda}}(X_i)\right\} \leq \alpha$. The standard conformal argument completes the proof:

$$\mathbb{P}(Y_{n+1} \notin \mathcal{C}(X_{n+1})) = \mathbb{P}(Y_{n+1} \notin \mathcal{C}_{\hat{\lambda}^{Y_{n+1}}}(X_{n+1})) = \mathbb{E}\left[\frac{1}{n+1}\sum_{i=1}^{n+1} \mathbb{1}\left\{Y_i \notin \mathcal{C}_{\hat{\lambda}^{Y_{n+1}}}(X_i)\right\}\right] \leq \alpha. \tag{8}$$

The work of Gibbs et al. (2023) extends the above argument beyond intercept-only quantile regression; roughly, the idea is to define a vector space of functions Λ whose elements map $\mathcal{X}$ to $\mathbb{R}$, and then repeat the argument above. We omit the details here, since the argument will be clear from the proof of our main theorem.

2.2 Main results

We now build up to our main result, which is analogous to Theorem 3 of Gibbs et al. (2023). The main difference is that we do not provide a conditional coverage guarantee, but rather, a conditional risk control guarantee. Furthermore, we handle function classes that depend on both the input and output: $\Lambda = \{(x, y) \mapsto \lambda(x, y)\}$. This allows the methodology to capture both group-conditional and label-conditional coverage.

Consider a nested family of sets, $\mathcal{C}_u(x)$, indexed by u. Let $\ell : \mathcal{Y} \times 2^{\mathcal{Y}} \to [0, 1]$ be a left-continuous and monotone nonincreasing loss function:

$$\mathcal{C}_1 \subseteq \mathcal{C}_2 \implies \ell(y, \mathcal{C}_1) \geq \ell(y, \mathcal{C}_2). \tag{9}$$

For convenience we will abuse notation to write $\ell(x, y, u) = \ell(y, \mathcal{C}_u(x))$. We also define a related indefinite integral (or antiderivative), $I : \mathcal{X} \times \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$, as

$$I(x, y, u) = \int (\ell(x, y, u') - \alpha) du' \Big|_u. \tag{10}$$

In the above equation, u' is a dummy variable used for the purpose of calculating the antiderivative, and we evaluate the antiderivative at the value u. Because ℓ is a monotone loss function, we are guaranteed that I is a quasiconvex function; this will pose some interesting challenges for forming the prediction set, as we will soon see. As an additional challenge, unlike the case of the pinball loss, this indefinite integral can not be computed analytically in general.

We now define the following functions, analogously to the previous section:

$$J^y(\lambda) = \frac{1}{n+1}\sum_{i=1}^{n} I(X_i, Y_i, \lambda(X_i, Y_i)) + \frac{1}{n+1} I(X_{n+1}, y, \lambda(X_{n+1}, y)) + \mathcal{R}(\lambda), \tag{11}$$

where $\mathcal{R} : \Lambda \to \mathbb{R}$ is a regularizer,

$$\hat{\lambda}^y = \operatorname*{argmin}_{\lambda\in\Lambda} J^y(\lambda), \tag{12}$$

and

$$\mathcal{C}(x) = \mathcal{C}_{\sup_{y\in\mathcal{Y}} \hat{\lambda}^y(x)}(x). \tag{13}$$

Then the set $\mathcal{C}(X_{n+1})$ has the following guarantee.

Theorem 1. *Consider a vector space Λ equipped with the standard addition operation, and assume that for all $\lambda, \lambda' \in \Lambda$, the derivative $\epsilon \mapsto \mathcal{R}(\lambda + \epsilon\lambda')$ exists. If λ is nonnegative and $\mathbb{E}[\lambda(X_{n+1}, Y_{n+1})] > 0$, then*

$$\mathbb{E}_\lambda[\ell(Y_{n+1}, \mathcal{C}(X_{n+1}))] \leq \alpha - \frac{1}{\mathbb{E}[\lambda(X_{n+1}, Y_{n+1})]}\mathbb{E}\left[\frac{d}{d\epsilon}\mathcal{R}(\hat{\lambda}^{Y_{n+1}} + \epsilon\lambda)\Big|_{\epsilon=0}\right]. \tag{14}$$

Proof. Pick any $\lambda \in \Lambda$ and $\epsilon \in [0, 1]$. For all $y \in \mathcal{Y}$, because $\hat{\lambda}^y$ is a minimizer of J^y and Λ, we have that the first-order optimality condition is satisfied. Thus, for Y_{n+1},

$$0 \in \partial_\epsilon J^{Y_{n+1}}(\hat{\lambda}^{Y_{n+1}} + \epsilon\lambda)\Big|_{\epsilon=0}. \tag{15}$$

Now we will define any subgradients $g_i(\lambda)$ that satisfy a similar list of conditions as we previously defined in the proof of conformal prediction. We start by defining some useful functions. Let $L_i(\lambda) = \lambda(X_i, Y_i)(\ell(X_i, Y_i, \hat{\lambda}^{Y_{n+1}}(X_i, Y_i)) - \alpha)$, $U_i(\lambda) = \lim_{\epsilon\to 0^+} \lambda(X_i, Y_i)(\ell(X_i, Y_i, (\hat{\lambda}^{Y_{n+1}} +$

$\epsilon\lambda)(X_i, Y_i)) - \alpha)$, and $r(\lambda) := \frac{d}{d\epsilon}\mathcal{R}(\hat{\lambda}^{Y_{n+1}} + \epsilon\lambda)\Big|_{\epsilon=0}$, for $i \in [n+1]$. Importantly, $U_i \geq L_i$ deterministically, since $(\hat{\lambda}^{Y_{n+1}} + \epsilon\lambda)(X_i, Y_i) = \hat{\lambda}^{Y_{n+1}}(X_i, Y_i) + \epsilon\lambda(X_i, Y_i)$ and $\lambda(X_i, Y_i) \geq 0$. Returning to the conditions for the subgradients, we pick any subgradients $g_1, \ldots, g_{n+1}$ satisfying

1. $g_i(\lambda) \in [L_i(\lambda), U_i(\lambda)]$.
2. $\frac{1}{n+1}\sum_{i=1}^{n+1} g_i(\hat{\lambda}^{Y_{n+1}}) + r(\lambda) = 0$.

Let

$$\mathcal{I}_1 = \{i \in [n+1] : L_i(\lambda) \neq U_i(\lambda)\}. \tag{16}$$

Then, we can write

$$\frac{1}{n+1}\sum_{i=1}^{n+1} g_i(\hat{\lambda}^{Y_{n+1}}) + r(\lambda) = 0 \tag{17}$$

$$\iff \frac{1}{n+1}\sum_{i=1}^{n+1} \lambda(X_i, Y_i)(\ell(X_i, Y_i, \hat{\lambda}^{Y_{n+1}}(X_i, Y_i)) - \alpha) \tag{18}$$

$$= \frac{1}{n+1}\sum_{i\in\mathcal{I}} \left(L_i - g_i(\hat{\lambda}^{Y_{n+1}})\right) - r(\lambda). \tag{19}$$

But for all $i \in \mathcal{I}$, $g_i(\hat{\lambda}^{Y_{n+1}}) \geq L_i(\lambda)$. Thus, the right-hand side of the displayed equation is no greater than $-r(\lambda)$, implying that $\frac{1}{n+1}\sum_{i=1}^{n+1} \ell(X_i, Y_i, \hat{\lambda}^{Y_{n+1}}(X_i, Y_i)) - \alpha \leq -r(\lambda)$.

Now we apply our standard exchangeability arguments. By exchangeability and the symmetry of $\hat{\lambda}^{Y_{n+1}}$, we have that

$$\mathbb{E}[\lambda(X_i, Y_i)(\ell(X_i, Y_i, \hat{\lambda}^{Y_{n+1}}(X_i, Y_i)) - \alpha)] \tag{20}$$

$$= \mathbb{E}\left[\frac{1}{n+1}\sum_{i=1}^{n+1} \lambda(X_i, Y_i)(\ell(X_i, Y_i, \hat{\lambda}^{Y_{n+1}}(X_i, Y_i)) - \alpha)\right] \tag{21}$$

$$\leq -\mathbb{E}[r(\lambda)]. \tag{22}$$

Thus, rearranging terms, $\mathbb{E}_\lambda[\ell(X_i, Y_i, \hat{\lambda}^{Y_{n+1}}(X_i, Y_i))] \leq \alpha - \frac{1}{\mathbb{E}[\lambda(X_i, Y_i)]}\mathbb{E}[r(\lambda)]$.

For the final conclusion, $\mathcal{C}(X_{n+1}) \supseteq \mathcal{C}_{\hat{\lambda}^{Y_{n+1}}}$, by definition we have that

$$\mathbb{E}_\lambda[\ell(Y_{n+1}, \mathcal{C}(X_{n+1}))] \leq \mathbb{E}_\lambda[\ell(Y_{n+1}, \mathcal{C}_{\hat{\lambda}^{Y_{n+1}}}(X_{n+1}))] \leq \alpha - \frac{1}{\mathbb{E}[\lambda(X_i, Y_i)]}\mathbb{E}[r(\lambda)]. \tag{23}$$

□

2.3 Efficient computation of $\hat{\lambda}$

The procedure we have outlined thus far is analogous to full conformal risk control (see Angelopoulos (2024)), in that we must loop over all values of $y \in \mathcal{Y}$ to calculate $\hat{\lambda}$. We have avoided including the model retraining as part of this procedure, but regardless, it may be impossible or infeasible to loop through $\mathcal{Y}$.

However, this can be avoided. In particular, assume for all $\lambda \in \Lambda$ and all (x, y) that $\lambda(x, y) \leq \nu(x)$. For example, if Λ is a class of bounded functions, then $\nu(x)$ can be set to the bound. Another special case is when $\lambda(x, y)$ does not depend on y, in which case ν exists trivially. The following optimization problem also provides risk control, but does not require looping through $y \in \mathcal{Y}$:

$$\tilde{\lambda} = \operatorname*{argmin}_{\lambda\in\Lambda} \tilde{J}(\lambda) = \frac{1}{n+1}\sum_{i=1}^{n} I(X_i, Y_i, \lambda(X_i)) + \frac{1-\alpha}{(n+1)}\lambda(X_{n+1}) + \mathcal{R}(\lambda). \tag{24}$$

To see why this algorithm provides risk control, assume for convenience that ℓ is continuous in its last argument. Then, the first-order optimality of $\tilde{\lambda}$ implies that for all λ,

$$\begin{aligned}
&\frac{d}{d\epsilon}\tilde{J}(\tilde{\lambda} + \epsilon\lambda)\Big|_{\epsilon=0} = 0 \\
\implies &\frac{d}{d\epsilon}\left(\frac{1}{n+1}\sum_{i=1}^{n} I(X_i, Y_i, \tilde{\lambda}(X_i) + \epsilon\lambda(X_i)) \right. \\
&\quad + \frac{1-\alpha}{(n+1)}(\tilde{\lambda}(X_{n+1}) + \epsilon\lambda(X_{n+1})) \\
&\quad \left. + \mathcal{R}(\tilde{\lambda} + \epsilon\lambda)\right)\Bigg|_{\epsilon=0} = 0 \\
\implies &\frac{1}{n+1}\sum_{i=1}^{n} \lambda(X_i)(\ell(X_i, Y_i, \tilde{\lambda}(X_i)) - \alpha) \\
&\quad + \frac{1-\alpha}{n+1}\lambda(X_{n+1}) \\
&\quad = -\frac{\partial}{\partial\epsilon}\mathcal{R}(\tilde{\lambda} + \epsilon\lambda)\Big|_{\epsilon=0} \\
\implies &\frac{1}{n+1}\sum_{i=1}^{n+1} \lambda(X_i)(\ell(X_i, Y_i, \tilde{\lambda}(X_i)) - \alpha) \\
&\quad \leq -\frac{\partial}{\partial\epsilon}\mathcal{R}(\tilde{\lambda} + \epsilon\lambda)\Big|_{\epsilon=0}.
\end{aligned}$$

from which we can then continue on with the same exchangeability arguments in Theorem 3 to prove a risk-control bound.

We make some final observations about solving this optimization problem. In the absence of regularization, solving an optimization problem over $J(\lambda)$ is a quasiconvex optimization problem, and standard first-order methods can get stuck in saddle points or local minima. However, a saddle point or local minimum is fine from the purpose of risk control—our analyses rely only on local first-order optimality conditions. For maximum performance, it is best to escape the saddle points to find the global minimum; noisy gradient descent has been shown to be an effective method for this purpose (Jin et al., 2017). All that said, in our experiments, we have never encountered a problem with the standard SciPy automatic solvers, such as `scipy.optimize.minimize`.

3 RESULTS

We have not yet discussed one of our main contributions: how do we pick Λ automatically? Our answer is straightforward: the preceding sections have shown that we should think of $\hat{\lambda}(X_i, Y_i)$ as an error-prediction algorithm, much like the scorecaster of Angelopoulos et al. (2023). We use this perspective to parameterize the function class to yield the best predictor possible.

For semantic segmentation tasks, we use a standard deep-learning approach in which we train a convolutional neural network that predicts the highest threshold (on the softmax of pixels) such that the risk (on this single image) is lower than α. We then slice off the last fully connected layer, and the resulting feature extractor becomes our $\Phi(x)$. The class of functions $\Lambda = \{x \mapsto \Phi(x)^\top \theta | \theta \in \mathbb{R}^d\}$ is defined as the space of linear functions of this embedding. This procedure is presented in Figure 3. This essentially amounts to a rigorous method for fine-tuning a fully-connected layer on a pretrained network backbone to provide risk-controlled estimates.

For tabular regression tasks, where neural networks are not the tool of choice (Shwartz-Ziv and Armon, 2022), we create an embedding with a Random Forest (RF) (Breiman, 2001), building on the work of Amoukou and Brunel (2023) for computing adaptive predictive intervals. Our Algorithm 1 is analogous to theirs. The main idea is to train a RF model to learn quantiles of the error distribution of the base model, and then to consider each leaf a group. This procedure assigns to each observation as many groups as there are trees in the RF. We then set $\Phi(x)$ to be the vector of group indicators, and Λ to be the space of linear functions of $\Phi(x)$.

3.1 Regression task

Our first example is a simple simulation in the context of prediction intervals. This experiment is primarily meant to visually showcase the automatic selection of groups. In this setting, let f be any regression model that takes as input $x \in \mathbb{R}$ and predicts $y \in \mathbb{R}$. The goal is to control the coverage of our prediction intervals; hence, our loss function will be defined as follows:

$$\ell(x, y, \lambda) = \mathbb{1}\{y \in \mathcal{C}_{\lambda(x,y)}(x)\}, \text{ where } \mathcal{C}_u(x) = [\hat{f}(x) \pm u] \tag{25}$$

for all $u \in \mathbb{R}$.

Algorithm 1 Random Forest Training and inference for automatic group creation

Require: $\mathcal{D}_{res} = \{(X_1, |y_1 - \hat{y}_1|), \ldots, (X_N, |y_N - \hat{y}_N|)\}$, $\mathcal{D}_{cal} = \{(X_1, y_1), \ldots, (X_m, y_m)\}$

```
Initialize RF ← RandomForest
Train RF on D_res
for all element x in D_cal do
    G ← [0]^|RF|  {|RF| is the number of leafs in the forest}
    for all tree T ∈ RF do
        for all leaf L ∈ T do
            if x ∈ L then
                G[L] ← 1
            end if
        end for
    end for
end for
```

We use a simulated dataset from Romano et al. (2019). We used 2000 points for training, 1000 for the residual RF training, and 9000 for calibration and 5000 test points. Results are reported in Figure 2. The obtained marginal coverage is 0.897 with a target coverage $1 - \alpha = 0.9$ and the coverage of each group varies between 0.886 and 0.913, which is within the expected fluctuations for a test set of this size.

3.2 Semantic segmentation

In this setting, let f be any semantic segmentation model which takes as input $x \in \mathbb{R}^{d_1 \times d_2 \times c}$ and predicts sigmoids $f(x) \in [0, 1]^{d_1 \times d_2}$. Our target is a binary segmentation mask in $\mathcal{Y} = \{0, 1\}^{d_1 \times d_2}$, and for any $y \in \mathcal{Y}$, we abuse notation and refer to $|y| = \mathbb{1}^\top y \mathbb{1}$ as the sum of all the pixels. The objective here is to control the recall of the segmentation model. In particular, we index our final segmentation with threshold $u \in [0, 1]$ as $\mathcal{C}_u(x) \in \mathcal{Y}$, and $\mathcal{C}_u(x)_{i,j} = \mathbb{1}\{f(x)_{i,j} \geq u\}$. With this in hand, the loss function ℓ is defined as follows:

$$\ell(x, y, \lambda) = 1 - \frac{y \cap \mathcal{C}_{\lambda(x,y)}(x)}{|y|}. \tag{26}$$

Polyp segmentation dataset. For this experiment, we used a PraNet (Fan et al., 2020) model for the semantic segmentation and a ResNet-50 (He et al., 2016) for the embedding learning. We chose an embedding

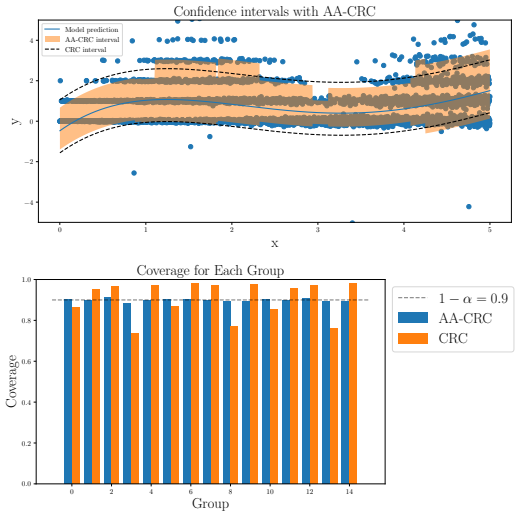

Figure 2: **Top figure.** The blue curve is the model prediction, blue dots are test data points, and prediction intervals are shown in orange. The dotted lines represent the CRC prediction intervals which have constant width. **Bottom figure.** It shows the within-group coverage for each of the adaptively selected groups. The red line is the target coverage level. The coverage is almost exact for all groups for AA-CRC while almost all groups are either undercovered or overcovered this the standard CRC method.

size of 1024. Both models were trained on Kvasir-SEG (Jha et al., 2020) and CVC-ClinicDB (Bernal et al., 2017), and the calibration and testing were performed on the CVC-300 (Vazquez et al., 2017), CVC-ClinicDB, CVC-ColonDB (Tajbakhsh et al., 2015), ETIS-LaribPolypDB (Silva et al., 2014), and Kvasir datasets. In total, 1450 images were used for the training and embedding learning, and 798 images were used for calibration and testing. Results are reported in Figure 5 with $\alpha = 0.1$.

The mean and standard deviation of the recall over 100 random splits are 0.906 and 0.021 respectively. The average precision of the AA-CRC method is 0.457 versus 0.395 for standard CRC, showing a significant improvement of the precision while guaranteeing the same level of recall. The reason we showed the improvement in precision is that the improved conditional performance essentially allows us to improve the overall performance of the method: that is, subject to the same type-1 error α, the adaptive procedure gets better type-2 error. This does imply better conditional performance, since more correct segmentations are being made, allowing us to allocate the error budget more efficiently.

Moreover, as shown in Figure 4, when performing a PCA on the embeddings of the images and plot the two first components, and then attribute to each point the value of the threshold returned AA-CRC, we can visualize the adaptiveness of the methodology by seeing that the closer the embeddings, the closer the thresholds.

We also evaluated the conditional performance of our method by analyzing the relationship between image recall at a fixed threshold of 0.5 and the adaptive thresholds assigned by the AA-CRC method. Recall at a fixed threshold reflects image difficulty: low recall indicates a challenging image requiring a lower threshold, while high recall suggests the threshold could be increased without degrading performance. To quantify this adaptiveness, we computed the Spearman correlation between recall at the fixed threshold and the AA-CRC-assigned thresholds. A positive correlation would indicate that AA-CRC assigns higher thresholds to easier images, demonstrating adaptive behavior. On the Polyp dataset, we observed a Spearman correlation of 0.41 with a p-value of 4×10^{-23}, confirming a positive relationship where higher recall generally corresponds to higher thresholds. To further illustrate this trend, we grouped images into recall bins from 0 to 1 in increments of 0.1, computing the average AA-CRC threshold and its standard deviation for each bin. A bar (Figure 4 visualizes this relationship, showing an overall increase in thresholds as recall improves, reinforcing the adaptive nature of the AA-CRC method.

Fire segmentation dataset. We next perform experiments on fire segmentation from image data, using the dataset of (Aktaş, 2023). We used a UNet (Ronneberger et al., 2015) for the segmentation backbone and a ResNet-50 to caculate the score embedding. We chose the last hidden layer to have size 1024. We used 11671 images to train the UNet and ResNet-50 models, and respectively, 3432 and 6865 images for calibration and testing. To achieve better results in terms of precision, we performed a PCA (Wold et al., 1987) on the embedding and added an intercept. The number of components was chosen such that explained variance ratio was equal to 0.85. Results are reported in Figure 6 with $\alpha = 0.1$. The mean and standard deviation of recall over 100 random splits of the data are 0.898 and 0.003 respectively. The average precision of our method is 0.403, versus 0.363 for standard CRC, again improving the precision at the same recall level.

4 CONCLUSION AND FUTURE WORK

We have presented a generalization of Gibbs et al. (2023), AA-CRC that handles monotonic risks and adaptively chosen groups. We demonstrated the benefits of AA-CRC through the improvement of the preci-

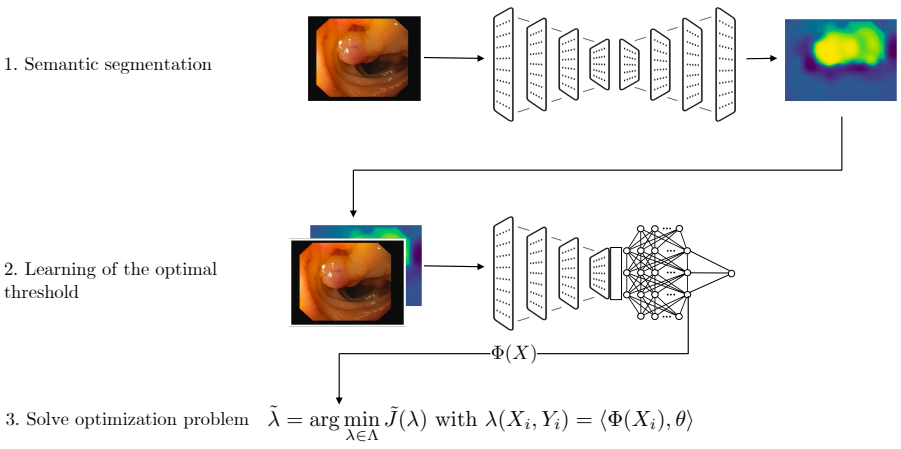

Figure 3: Procedure to create the embedding of the images. The **first step** is the training of the segmentation model on the $\mathcal{D}_{train}$ dataset. The **second step** is the learning of the embedding based on the segmentation output on the $\mathcal{D}_{res}$ dataset. The **third step** is the solving of the optimization procedure and the $\mathcal{D}_{cal}$ dataset.

sion in semantic segmentation tasks while controlling the recall. Moreover, we proposed a systematic methodology, for both tabular and image data, to construct adaptive function classes Λ without needing any *a priori* knowledge. Future work will focus on the extension of this methodology to multiclass semantic segmentation, as well as using exploring additional choices of function spaces Λ. Proving extensions of the remaining theorems in Gibbs et al. (2023)—such as the bound in their (3.3), would also be available via standard analysis (e.g., via analyzing the same jump function as in Angelopoulos et al. (2024)).

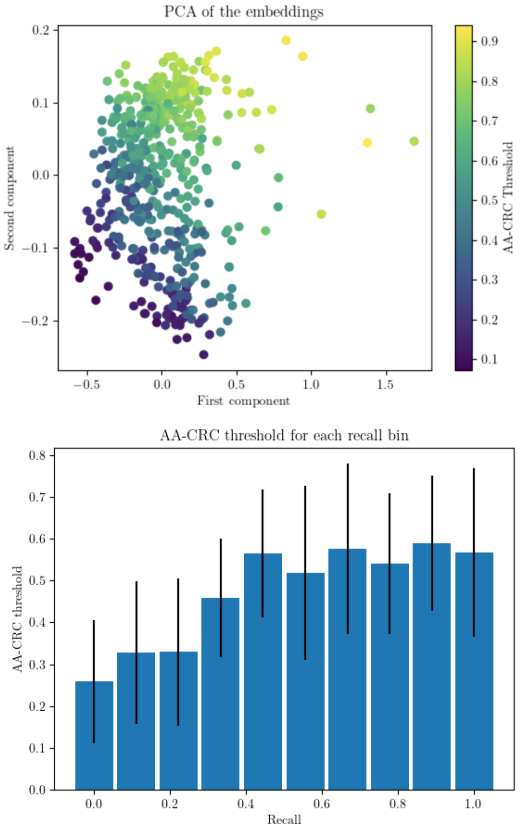

Figure 4: **Top figure.** Plot of the two first components of the PCA on the embeddings of the images. The color of each point correspond to the threshold returned by AA-CRC. **Bottom figure.** The right figure shows the within-group coverage for each of the adaptively selected groups. The red line is the target coverage level. The coverage is almost exact for all groups for AA-CRC while almost all groups are either undercovered or overcovered this the standard CRC method.

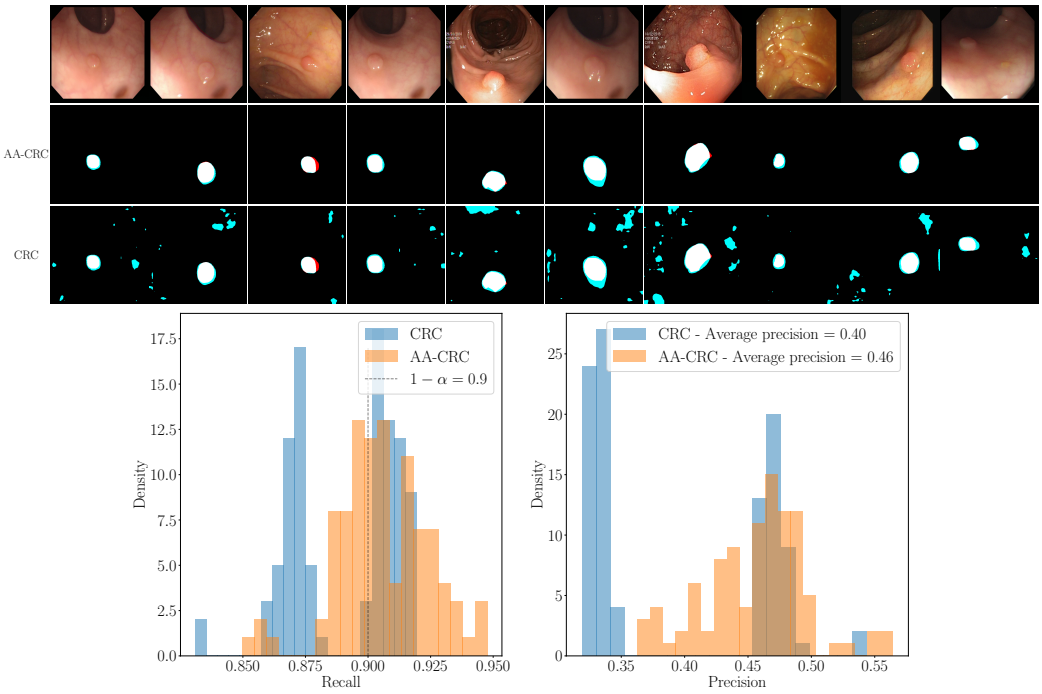

Figure 5: **Recall control for polyp segmentation.** The top figure compares the control of the recall made with our method (AA-CRC) to the control done with CRC. White pixels are true positives, blue pixels are false positives and red pixels are false negatives. The bottom figures represents the distribution of the recall of our procedure and distribution of the precision for both CRC and AA-CRC over 100 independent random data split.

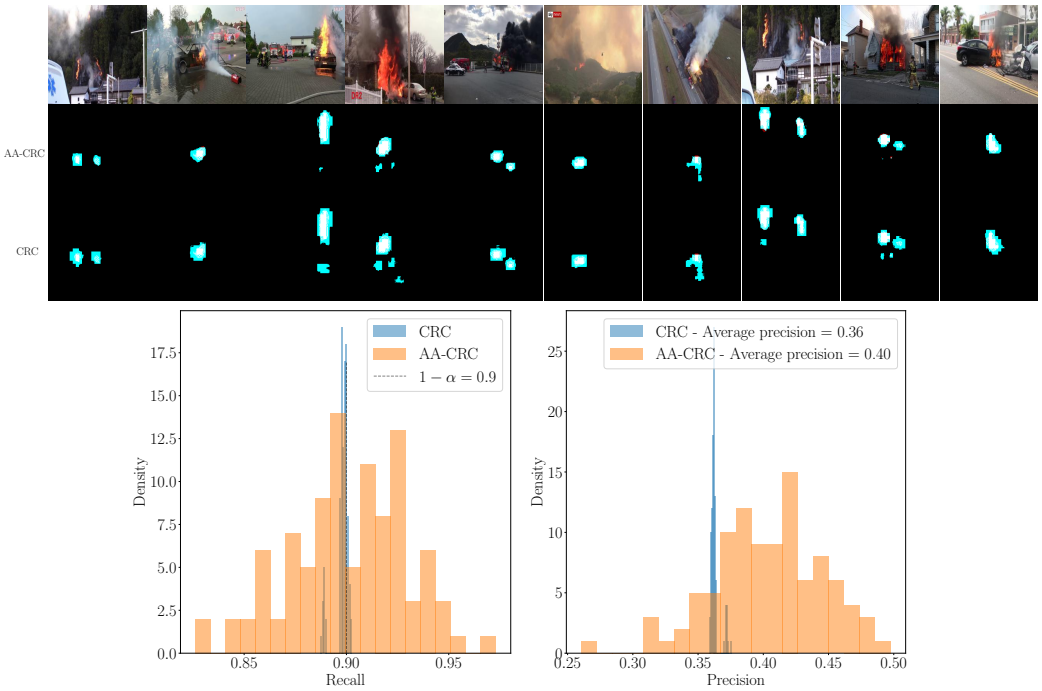

Figure 6: **Recall control for fire segmentation.** The top figure compares the control of the recall made with our method to the control done with CRC. White pixels are true positives, blues are false positives and reds are false negatives. The bottom figures represents the distribution of the recall of our procedure and distribution of the precision for both CRC and AA-CRC over 100 independent random data split.

References

Aktaş, M. (2023). Fire Segmentation dataset - Kaggle.

Amoukou, S. I. and Brunel, N. J. (2023). Adaptive conformal prediction by reweighting nonconformity score. *arXiv:2303.12695*.

Angelopoulos, A. N. (2024). Note on full conformal risk control.

Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., and Schuster, T. (2024). Conformal risk control. In *12th International Conference on Learning Representations*.

Angelopoulos, A. N., Candes, E., and Tibshirani, R. (2023). Conformal PID control for time series prediction. In *Neural Information Processing Systems*.

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

Bastani, O., Gupta, V., Jung, C., Noarov, G., Ramalingam, R., and Roth, A. (2022). Practical adversarial multivalid conformal prediction. *Advances in Neural Information Processing Systems*, 35:29362–29373.

Bernal, J., Tajkbaksh, N., Sánchez, F., Matuszewski, B., Chen, H., Yu, L., Angermann, Q., Romain, O., Rustad, B., Balasingham, I., Pogorelov, K., Choi, S., Debard, Q., Maier-Hein, L., Speidel, S., Stoyanov, D., Brandao, P., Cordova, H., Sánchez-Montes, C., Gurudu, S., Fernández-Esparrach, G., Dray, X., Liang, J., and Histace, A. (2017). Comparative validation of polyp detection methods in video colonoscopy: Results from the MICCAI 2015 endoscopic vision challenge. *IEEE Transactions on Medical Imaging*, 99.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Fan, D.-P., Ji, G.-P., Zhou, T., Chen, G., Fu, H., Shen, J., and Shao, L. (2020). Pranet: Parallel reverse attention network for polyp segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 263–273.

Gibbs, I., Cherian, J. J., and Candès, E. J. (2023). Conformal prediction with conditional guarantees. *arXiv:2305.12616*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Jha, D., Smedsrud, P. H., Riegler, M. A., Halvorsen, P., De Lange, T., Johansen, D., and Johansen, H. D. (2020). Kvasir-SEG: A segmented polyp dataset. In *26th International Conference on Multimedia Modeling*, pages 451–462.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. (2017). How to escape saddle points efficiently. In *International Conference on Machine Learning*, pages 1724–1732. PMLR.

Jung, C., Lee, C., Pai, M., Roth, A., and Vohra, R. (2021). Moment multicalibration for uncertainty estimation. In *Conference on Learning Theory*, pages 2634–2678.

Jung, C., Noarov, G., Ramalingam, R., and Roth, A. (2022). Batch multivalid conformal prediction. *arXiv preprint arXiv:2209.15145*.

Kim, M. P., Ghorbani, A., and Zou, J. (2019). Multiaccuracy: Black-box post-processing for fairness in classification. In *AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254.

Koenker, R. and Bassett Jr, G. (1978). Regression quantiles. *Econometrica: Journal of the Econometric Society*, 46(1):33–50.

Romano, Y., Patterson, E., and Candès, E. J. (2019). Conformalized quantile regression. In *Neural Information Processing Systems*.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241.

Shwartz-Ziv, R. and Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90.

Silva, J., Histace, A., Romain, O., Dray, X., and Granado, B. (2014). 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.

Tajbakhsh, N., Gurudu, S. R., and Liang, J. (2015). Automated polyp detection in colonoscopy videos using shape and context information. *IEEE Transactions on Medical Imaging*, 35(2):630–644.

Vazquez, D., Bernal, J., Sánchez, F. J., Fernández-Esparrach, G., Lopez, A. M., Romero, A., Drozdzal, M., and Courville, A. (2017). A benchmark for endoluminal scene segmentation of colonoscopy images. *Journal of Healthcare Engineering*, pages 1–9.

Vovk, V. (2012). Conditional validity of inductive conformal predictors. In *Asian Conference on Machine Learning*, volume 25, pages 475–490.

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic Learning in a Random World*. Springer.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52. Multivariate Statistical Workshop for Geologists and Geochemists.

Zhang, L., Roth, A., and Zhang, L. (2024). Fair risk control: A generalized framework for calibrating multigroup fairness risks. *arXiv preprint arXiv:2405.02225*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes. Detailed description in paragraph 2.2**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **No**
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes. Given in the anonymized repository**

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes. Detailed description in paragraph 2.2**
   (b) Complete proofs of all theoretical results. **Yes. Proof in paragraph 2.2**
   (c) Clear explanations of any assumptions. **Yes. Detailed description in paragraph 2.2**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes. Given in the anonymized repository**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen).**Yes. Given in the anonymized repository**
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes. number of splits given to create the distributions of the precision and recall**
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **No.**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **Yes**
   (b) The license information of the assets, if applicable. **Not applicable.**
   (c) New assets either in the supplemental material or as a URL, if applicable. **Not Applicable**
   (d) Information about consent from data providers/curators. **No**
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not Applicable**

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