Two-stage Conformal Risk Control with Application to Ranked Retrieval

Yunpeng Xu[1], Mufang Ying[2], Wenge Guo*[3], and Zhi Wei[1]

[1]Department of Computer Science, New Jersey Institute of Technology
[2]Department of Statistics, Rutgers University - New Brunswick
[3]Department of Mathematical Sciences, New Jersey Institute of Technology

November 2, 2024

Abstract

Many practical machine learning systems, such as ranking and recommendation systems, consist of two concatenated stages: retrieval and ranking. These systems present significant challenges in accurately assessing and managing the uncertainty inherent in their predictions. To address these challenges, we extend the recently developed framework of conformal risk control, originally designed for single-stage problems, to accommodate the more complex two-stage setup. We first demonstrate that a straightforward application of conformal risk control, treating each stage independently, may fail to maintain risk at their pre-specified levels. Therefore, we propose an integrated approach that considers both stages simultaneously, devising algorithms to control the risk of each stage by jointly identifying thresholds for both stages. Our algorithm further optimizes for a weighted combination of prediction set sizes across all feasible thresholds, resulting in more effective prediction sets. Finally, we apply the proposed method to the critical task of two-stage ranked retrieval. We validate the efficacy of our method through extensive experiments on two large-scale public datasets, MSLR-WEB and MS MARCO, commonly used for ranked retrieval tasks.

1 Introduction

As machine learning models and systems become increasingly integrated into our daily lives, there is a growing demand for the transparency and reliability in their predictions. Rather than accepting black-box predictions, we urgently seek to understand the uncertainty of these models and systems. Consequently, it is more important than ever to accurately measure and control the uncertainty of their predictions.

Among many solutions to the uncertainty problem, one received significant attention in the machine learning field recently, which is conformal prediction (Vovk et al., 2005). It is distribution-free, statistically rigorous, and easy to integrate with existing machine learning models, with the goal to create uncertainty sets for predictions made by these models while satisfying a certain coverage requirement. Particularly, the recently developed conformal risk control framework (Angelopoulos et al., 2024) extends conformal prediction from traditional miscoverage control to a more general setting, where it controls the expected value of any loss function. This significantly broadens its applicability to various new contexts.

*Author e-mail addresses: yx8@njit.edu, mufang.ying@rutgers.edu, wenge.guo@njit.edu, zhi.wei@njit.edu

Most existing research on conformal prediction assumes a single-stage process for the machine learning system under study, where the system processes the input to produce the prediction in a single step. However, this assumption may not hold for many real-world machine learning systems, which often consist of two or more concatenated stages. For example, consider the ranked retrieval problem, which involves retrieving and ranking documents from a repository based on their relevance to a user's query. Due to the large volume of documents in the repository, search engines typically employ a two-stage or even multi-stage process (Yin and et al, 2016; Khattab et al., 2020). In this setup, the initial stage retrieves a small number of candidate documents from the repository, while subsequent stages refine and rank these documents before presenting the final list to the user. Furthermore, different stages may have distinct optimization goals, and errors in one stage can cascade to subsequent stages. This multi-stage setup adds complexity and presents a unique challenge in accurately measuring and controlling uncertainty in ranked retrieval problems.

In this paper, we propose a *two-stage* conformal prediction method as our approach to quantify and control the uncertainty inherent in these problems. Specifically, we extend the recently developed single-stage conformal risk control framework to a two-stage setup, where each stage has its own risk control requirements. Control is achieved by identifying parameters that satisfy the risk constraints for both stages. Furthermore, to address the specific purpose of the two stages in ranked retrieval problems, we define the *retrieval risk* and the *ranking risk*, respectively, and then apply our proposed two-stage conformal risk control method to derive their corresponding prediction sets while respectively controlling the retrieval risk and the ranking risk at pre-specified levels. This method does not rely on assumptions about the underlying ranking model and can easily be integrated into an existing ranked retrieval system. It is also easily adaptable to similar multi-stage risk control problems.

Our major contributions are as follows:

- We extended the conformal risk control framework to a two-stage setup, where there exists an inter-stage dependency.
- We developed a novel two-stage conformal risk control method to control the risk of both stages within guaranteed risk bounds.
- We properly formulated the uncertainty measurement for a typical two-stage ranked retrieval problem using the extended conformal risk control framework.
- We thoroughly tested our proposed method on two datasets for real-life ranked retrieval tasks and demonstrated the validity of the method.

1.1 Related Work

Conformal prediction, originally developed by Vladimir Vovk and his collaborators, has recently emerged as a prominent and widely embraced approach for addressing uncertainty quantification in statistical machine learning (Vovk et al., 1999, 2005). Angelopoulos and Bates provide a comprehensive survey of this field, highlighting its significance and applications (Angelopoulos and Bates, 2021). Our research aligns with the category of split conformal prediction (Papadopoulos et al., 2002; Lei et al., 2015), specifically relating to the recently developed conformal risk control framework (Angelopoulos et al., 2024). This framework enhances standard conformal prediction by extending specific coverage measurement to a general risk assessment setting, offering a more comprehensive approach to evaluating prediction reliability, and therefore greatly improves its adoption in real problems.

The application discussed in our work bears some resemblance to Angelopoulos et al. (2023), which employs the Learn-then-Test technique (Angelopoulos et al., 2021) to control the false discovery rate in recommendation systems and optimize for recommendation diversity. However, despite tackling the same ranked retrial problems, our work and that of Angelopoulos et al. differ in both their objectives and methodologies.

The ranked retrieval problems have been extensively studied over the past few decades, with a broad spectrum of ranking models introduced, ranging from conventional Information Retrieval (IR) models like BM25 (Baeza-Yates and Ribeiro-Neto, 1999; Stephen and K., 1976) to modern learning-to-rank algorithms (Liu, 2009). Recently, as deep learning has led to breakthroughs in various machine learning problems, it has also been successfully adopted in the field of ranked retrieval (Severyn and Moschitti, 2015; Guo et al., 2016). Depending on how the loss functions are defined, these methods can generally be categorized into three types: pointwise algorithms (Crammer and Singer, 2001; Chu and Ghahramani, 2005), pairwise algorithms (Burges et al., 2005; Freund et al., 2003), and listwise algorithms (Burges et al., 2006; Cao et al., 2007).

2 Two-stage Conformal Risk Control

2.1 Problem Formulation

We investigate a learning system that consists of two consecutive stages, where the second stage depends on the output of the first stage. This setup is common in many real-world machine learning systems, such as ranking systems, recommendation systems, question & answer systems and object detection systems.

Let $D = \{(X_i, Y_i, Z_i)\}_{i=1}^n$ be a set of exchangeable calibration data, with $(X_i, Y_i, Z_i) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, where Y_i and Z_i are the labeled responses of feature vector X_i corresponding to the two stages respectively. For example, in a ranked retrieval problem, X_i is a query and its associated candidate documents, Y_i denotes the retrieval status of these documents, and Z_i corresponds to their ranking positions. We let $(X_{n+1}, Y_{n+1}, Z_{n+1}) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ denote the test point, where both Y_{n+1} and Z_{n+1} are unknown. Between the two response variables y and z in D, we assume that there exists a *link function* $S : \mathcal{Y} \to 2^{\mathcal{Z}}$ such that for any $(x, y, z) \in D \subset \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, we have $z \in S(y)$.

In the first stage, given a trained model $f : \mathcal{X} \times \mathcal{Y} \to [0, 1]$, we post-process the model's raw output to construct a prediction set $C_\lambda(x) = \{y : f(x, y) \geq 1 - \lambda\}$ for $x \in \mathcal{X}$. Different values of the parameter λ determine a sequence of prediction sets $\{C_\lambda(x) : \lambda \in [0, 1]\}$. To measure the quality of the prediction set C_λ, we consider a bounded loss function $l(\lambda) := l(C_\lambda(x), y) \in [0, 1]$ for $(x, y) \in \mathcal{X} \times \mathcal{Y}$. The function $l(\lambda)$ is assumed to be non-increasing and right-continuous in λ with $l(0) = 1$ and $l(1) = 0$. Let $L_i(\lambda) = l(C_\lambda(X_i), Y_i)$ be the first-stage loss of the i^{th} data point where $i = 1, \ldots, n + 1$. The goal of this stage is to determine the threshold value of the parameter $\hat{\lambda} \in [0, 1]$ such that

$$\mathbb{E}[L_{n+1}(\hat{\lambda})] \leq \alpha, \tag{1}$$

where $\alpha \in [0, 1]$ is the pre-specified first-stage risk bound.

Similarly, given a trained model $g : \mathcal{X} \times \mathcal{Z} \to [0, 1]$ for the second stage, we construct a prediction set $\tilde{C}_{\lambda,\gamma}(x) = \{z \in S(y) : f(x, y) \geq 1 - \lambda,\ g(x, z) \geq 1 - \gamma\}$ for $x \in \mathcal{X}$. Note that the parameter λ is present since the output of the second stage depends on the output of the first stage. Therefore, different values of the parameter pair (λ, γ) determine a sequence of prediction sets $\{\tilde{C}_{\lambda,\gamma}(x) : (\lambda, \gamma) \in [0, 1] \times [0, 1]\}$. Consider a bounded loss function $\tilde{l}(\lambda, \gamma) := \tilde{l}(\tilde{C}_{\lambda,\gamma}(x), z) \in [0, 1]$ for $(x, z) \in \mathcal{X} \times \mathcal{Z}$ and assume it is non-increasing and

right-continuous in both λ and γ, particularly $\tilde{l}(0,0) = 1$ and $\tilde{l}(1,1) = 0$. Let $\tilde{L}_i(\lambda, \gamma) = \tilde{l}(\tilde{C}_{\lambda,\gamma}(X_i), Z_i))$ be the second-stage loss of the i^{th} data point where $i = 1, \ldots, n+1$. The goal of this stage is to determine the threshold value $\hat{\gamma} \in [0,1]$ of γ given the first-stage determined value $\hat{\lambda} \in [0,1]$ of λ such that

$$\mathbb{E}[\tilde{L}_{n+1}(\hat{\lambda}, \hat{\gamma})] \leq \beta, \tag{2}$$

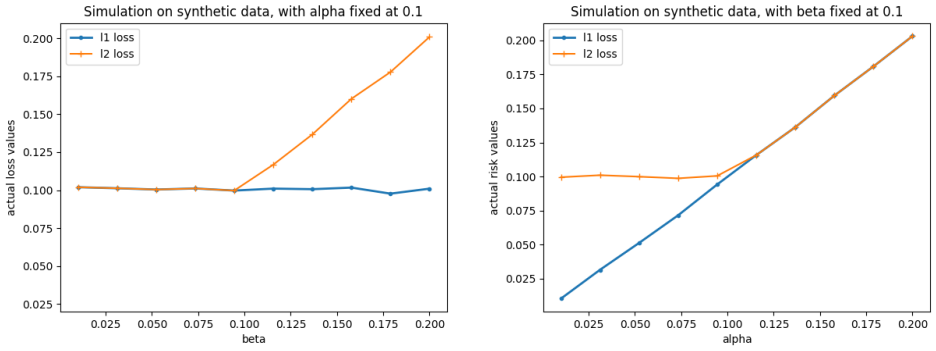
where $\beta \in [0,1]$ is the pre-specified second-stage risk bound.

2.2 Risk Control

To address a two-stage risk control task, a straightforward or ad hoc approach is to decompose it into two independent one-stage tasks. This naturally leads to framing it as two separate one-stage conformal risk control problems, with the standard conformal risk control algorithm applied to each stage— a method referred to as AdHoc CRC in this paper. While this approach may seem intuitive, it often fails to adequately control risk at the second stage, as demonstrated in a simple simulation study.

In the simulation, we generate a 2D synthetic dataset with two classes $\mathcal{C}_1$ and $\mathcal{C}_2$. Each data point is associated with two response variables: one binary variable representing the class label and one continuous variable. We design a two-stage learning task: Stage 1 identifies samples from $\mathcal{C}_1$, and Stage 2 predicts the range of the continuous response values. The risk for Stage 1 is defined by the false negative rate (FNR) of $\mathcal{C}_1$ samples, while the risk for Stage 2 is defined by the miscoverage rate of the true continuous response by the predicted range for $\mathcal{C}_1$ samples. For this purpose, we build two MLPs for the two stages respectively, one for classification, the other for regression. More details on the simulation can be found in the Technical Appendix.

The actual loss values for both stages, calculated using the AdHoc method, are presented in Figure 1. Panel (a) shows that when $\alpha = 0.1$, the risk of Stage 2 can be well controlled around 0.1 for $\beta \geq 0.1$.



(a) Fix $\alpha = 0.1$ while changing β value (b) Fix $\beta = 0.1$ while changing α value

Figure 1: Risk control on the synthetic dataset.

However, when $\beta < 0.1$, the risk of Stage 2 becomes unmanageable. Similarly, panel (b) illustrates that when $\beta = 0.1$, the risk of Stage 2 can also be controlled around 0.1 for $\alpha \leq 0.1$. However, when $\alpha > 0.1$, the risk of Stage 2 exceeds control. This simulation illustrates that treating the two stages independently can lead to inadequate risk control. This finding motivates us to develop an integrated approach that considers both stages simultaneously.

For a joint risk control of both stages, we need to carefully determine the feasible set of the conformal parameters values (λ, γ) so that it satisfies risk requirements of both stages simultaneously. This entails the following criteria:

1. For any pre-specified risk level $\alpha \in [0,1]$, a feasible λ needs to satisfy the first stage risk requirement $\mathbb{E}[L_{n+1}(\lambda)] \leq \alpha$. Denote the subset of such λ by Λ_1^*. Since $L_i(\lambda)$ is non-increasing and right-continuous in λ, then $\Lambda_1^* = [\lambda_1, 1]$, where $\lambda_1 = \inf\{\lambda : \mathbb{E}[L_{n+1}(\lambda)] \leq \alpha\}$.

2. For any pre-specified risk level $\beta \in [0,1]$, a feasible λ also needs to satisfy the second stage risk requirement $\mathbb{E}[\tilde{L}_{n+1}(\lambda, 1)] \leq \beta$. Here, γ is set to 1 to achieve the minimal risk value, as $\tilde{L}_{n+1}(\lambda, \gamma)$ is non-increasing in γ for a fixed λ. Denote the subset of such λ by Λ_2^*. Therefore, $\Lambda_2^* = [\lambda_2, 1]$, where $\lambda_2 = \inf\{\lambda : \mathbb{E}[\tilde{L}_{n+1}(\lambda, 1)] \leq \beta\}$. Following the first criteria, the feasible set of λ becomes the intersection of Λ_1^* and Λ_2^*, i.e., $\Lambda^* = \Lambda_1^* \cap \Lambda_2^* = [\max\{\lambda_1, \lambda_2\}, 1]$.

3. Furthermore, for any pre-specified risk level $\beta \in [0,1]$ and a specified $\lambda \in \Lambda^*$, a feasible γ needs to satisfy the second stage risk control, i.e., $\mathbb{E}[\tilde{L}_{n+1}(\lambda, \gamma)] \leq \beta$. Denote the subset of such γ by $\Gamma^*(\lambda)$. Since $\tilde{L}_i(\lambda, \gamma)$ is non-increasing in γ for a fixed λ, then $\Gamma^*(\lambda) = [\gamma_0(\lambda), 1]$, where $\gamma_0(\lambda) = \inf\{\gamma : \mathbb{E}[L_{n+1}(\lambda, \gamma)] \leq \beta\}$. Particularly, $\gamma_0(\lambda)$ takes its minimal value $\bar{\gamma}_0$ when $\lambda = 1$, i.e., $\gamma_0(\lambda) \geq \bar{\gamma}_0 = \inf\{\gamma : \mathbb{E}[L_{n+1}(1, \gamma)] \leq \beta\}$.

Combining the above three criteria, we have the feasible set of both parameters (λ, γ) as follows:

$$\Omega^* = \{(\lambda, \gamma) : \lambda \in \Lambda^*, \gamma \in \Gamma^*(\lambda)\}. \tag{3}$$

The set Ω^* consists of all possible values of (λ, γ) such that Equation (1) and (2) are both satisfied.

Proposition 1. For any pre-specified risk level $\alpha \in [0,1]$ and $\beta \in [0,1]$, assuming the loss functions $l(\lambda)$ and $\tilde{l}(\lambda, \gamma)$ that are both bounded, right-continuous and non-increasing in their respective arguments, if the prediction sets of the two stages are constructed using $(\lambda, \gamma) \in \Omega^*$, then it holds that $\mathbb{E}[L_{n+1}(\lambda)] \leq \alpha$ and $\mathbb{E}[\tilde{L}_{n+1}(\lambda, \gamma)] \leq \beta$.

The above proposition suggests the following three steps for a two-stage conformal risk control problem:

- Determine a subset $\Lambda_1^* = [\lambda_1, 1]$ such that for any $\lambda \in \Lambda_1^*$, $\mathbb{E}[L_{n+1}(\lambda)] \leq \alpha$;
- Determine a subset $\Lambda_2^* = [\lambda_2, 1]$ such that for any $\lambda \in \Lambda_2^*$, $\mathbb{E}[\tilde{L}_{n+1}(\lambda, 1)] \leq \beta$;
- For any $\lambda \in \Lambda_1^* \cap \Lambda_2^* = [\max\{\lambda_1, \lambda_2\}, 1]$, determine a subset $\Gamma^*(\lambda) = [\gamma_0(\lambda), 1]$ such that for any $\gamma \in \Gamma^*(\lambda)$, $\mathbb{E}[\tilde{L}_{n+1}(\lambda, \gamma)] \leq \beta$.

From a practical perspective, within the feasible set Ω^*, a solution that produces smaller prediction sets at both stages is preferred. Therefore, after achieving the goal of the two-stage risk control, we aim to identify $(\lambda^*, \gamma^*) \in \Omega^*$ that minimizes the expected value of a weighted combination of the sizes of the prediction sets derived at both stages, i.e.,

$$(\lambda^*, \gamma^*) = \underset{(\lambda,\gamma)\in\Omega^*}{\operatorname{argmin}} \mathbb{E}[w|C_\lambda(X_{n+1})| + (1-w)|\tilde{C}_{\lambda,\gamma}(X_{n+1})|] \tag{4}$$

where $|C_\lambda(X_{n+1})|$ and $|\tilde{C}_{\lambda,\gamma}(X_{n+1})|$ are the prediction set sizes for X_{n+1} for the two stages, respectively, w is a weight parameter to balance the importance between the two stages, particularly, if $w = 1$, we minimizes the prediction size of the first stage, while if $w = 0$, we minimizes the prediction size of the second stage.

To find the threshold values of the unknown parameters λ_1, λ_2, and $\gamma_0(\lambda)$, assume a calibration dataset $\mathcal{D} = \{(X_i, Y_i, Z_i)\}_{i=1}^n$ is in force, then λ_1 can be easily calculated using Equation (5) derived from the

standard conformal risk control algorithm Angelopoulos et al. (2024):

$$\hat{\lambda}_1 = \inf\left\{\lambda : \sum_{i=1}^{n} L_i(\lambda) \leq (n+1)\alpha - 1\right\}, \tag{5}$$

while λ_2, and $\gamma_0(\lambda)$ can be similarly calculated as the following:

$$\begin{aligned} \hat{\lambda}_2 &= \inf\left\{\lambda : \sum_{i=1}^{n} \tilde{L}_i(\lambda, 1) \leq (n+1)\beta - 1\right\}, \\ \hat{\gamma}(\lambda) &= \inf\left\{\gamma : \sum_{i=1}^{n} \tilde{L}_i(\lambda, \gamma) \leq (n+1)\beta - 1\right\}, \end{aligned} \tag{6}$$

where α and $\beta \in [\frac{1}{n+1}, 1]$.

Correspondingly, the feasible set Ω^* can be estimated as:

$$\hat{\Omega}^* = \{(\lambda, \gamma) : \lambda \in [\max\{\hat{\lambda}_1, \hat{\lambda}_2\}, 1], \gamma \in [\hat{\gamma}_0(\lambda), 1]\} \tag{7}$$

After identifying the estimated feasible set $\hat{\Omega}^*$ of (λ, γ), we aim to identify $(\hat{\lambda}, \hat{\gamma}) \in \hat{\Omega}^*$ that minimizes the a weighted average of the sizes of the prediction sets derived at both stages across n calibration points, i.e.,

$$(\hat{\lambda}, \hat{\gamma}) = \underset{(\lambda,\gamma)\in\hat{\Omega}^*}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left[w|C_\lambda(X_i)| + (1-w)|\tilde{C}_{\lambda,\gamma}(X_i)| \right] \tag{8}$$

A greedy algorithm to determine $(\hat{\lambda}, \hat{\gamma})$ is presented in Alg 1. In the algorithm, we use a linear grid search to find the optimal $(\hat{\lambda}, \hat{\gamma})$. It is worth mentioning that even with a linear search, the algorithm runs fast because the steps that determine the feasible set significantly reduce the search space. Furthermore, once the values of $(\hat{\lambda}, \hat{\gamma})$ are determined from the validation set, the inference on the test set requires only constant time.

Remark 1. We need to emphasize that our developed data-driven conformal method does not have a proven risk control for the second stage. However, in the Technical Appendix, we present a data-splitting approach that offers a finite sample guarantee for a two-stage risk control.

3 Conformal Ranked Retrieval

We employ the above two-stage risk control method to a ranked retrieval problem, which typically comprises a retrieval stage (referred to as candidate generation or $L1$) followed by a ranking stage (referred to as $L2$). This setup is necessary due to the sheer volume of documents, which surpasses the capacity of a single ranking model, particularly if it is resource-intensive. Consequently, a two-stage or even multi-stage process is devised to handle this complexity (Yin and et al, 2016; Khattab et al., 2020). For a given query and a document repository, in the retrieval stage, a set of candidate document is fetched. Subsequently, in the ranking stage, we rank these documents in descending order based on their relevance to the query. The primary objective is to ensure that the most relevant documents to the query are retrieved and ranked at high positions.

For clarity, we use superscripts for queries and subscripts for documents related to a query in this paper. For $i \in \{1, \ldots, n\}$, denote a query by q^i and its associated documents by $\{d^i_j\}_{j=1}^{m^i}$, where m^i is the number of documents associated with the query q^i. Denote the corresponding input feature vector by $X_i = \{x^i_j\}_{j=1}^{m^i}$,

Algorithm 1 Determine $(\hat{\lambda}, \hat{\gamma})$ given risk levels α and β

Input: data $\{(X_i, Y_i, Z_i)\}_{i=1}^n$, first-stage risk level α, second-stage risk level β, weight parameter w

Parameter: step size η

Output: $(\hat{\lambda}, \hat{\gamma})$

Calculate $\hat{\lambda}_1$ and $\hat{\lambda}_2$ using Equation (5) and (6).
$\lambda' \leftarrow \max\{\hat{\lambda}_1, \hat{\lambda}_2\}$, $S_{\min} \leftarrow \infty$
while $\lambda' \leq 1$ **do**
  Calculate $\gamma' = \hat{\gamma}(\lambda')$ using Equation (6)
  $S' \leftarrow \sum_{i=1}^n w|C_{\lambda'}(X_i)| + (1-w)|\tilde{C}_{\lambda',\gamma'}(X_i)|$
  if $S' < S_{\min}$ **then**
    $S_{\min} \leftarrow S'$, $\hat{\lambda} \leftarrow \lambda'$, $\hat{\gamma} \leftarrow \gamma'$
  end if
  $\lambda' \leftarrow \lambda' + \eta$
end while
return $(\hat{\lambda}, \hat{\gamma})$

where each x_j^i is generated by a specific feature extractor Φ and is given by $x_j^i = \Phi(q^i, d_j^i) \in \mathcal{F}_\Phi$. We assume that each pair of q^i and d_j^i is associated with a ground truth relevance score $r_j^i \in \{0, 1, \ldots, R\}$, where a higher value of r_j^i indicates a higher relevance of d_j^i to q^i. This score is only observable for the training data and is hidden for the test data. Here we denote the relevant documents in the retrieval results of q^i by $Y_i = \{d_j^i : r_j^i > 0\}$. In the ranking stage, with a focus on the ranking quality of documents with a ground truth relevance level r_0 or above, we denote the r_0-relevant documents for q^i as $Z_i = \{d_j^i : r_j^i \geq r_0\}$. Here, $r_0 \in \{1, \ldots, R\}$ is pre-selected for specific problems. Documents with relevance score equal or higher than r_0 are considered interchangeable in terms of their ranking positions.

Each stage is associated with a model learned on the training data for all queries, denoted by f for the retrieval model and by g for the ranking model, respectively. The models are applied to the feature vector x_j^i, and produce a predicted relevance score between d_j^i and q^i. Formally, $f, g : \mathcal{F}_\Phi \to [0, 1]$. The model's form, however, is not fixed; it could range from a simple Okapi BM25 model that counts word occurrences to a more complex large language model that generates embeddings for embedding-based retrieval. Typically, the model in the retrieval stage is more efficient but less powerful than the one in the ranking stage, thereby achieving a balance between efficiency and effectiveness.

To measure the quality of the inference results, we define a loss function for each stage in the following sections. For the retrieval stage, we assess the coverage of the ground truth document set by the retrieved document set. For the ranking stage, we evaluate the difference in ranking order between these two sets. Our goal is to control risks at both stages while maintaining a relatively small prediction set size. Keep in mind that the prediction set size quantifies the uncertainty of the retrieval or ranking results—a larger set size indicates higher uncertainty in the inference, and vice versa.

3.1 Conformal Retrieval Control

As defined in the previous section, Y_i is the set of relevant documents with respect to the query q^i, i.e., the set of documents with ground truth relevance scores greater than 0. To represent the set of documents fetched by

the model used in the retrieval stage, we define the retrieved document set $\hat{D}_\lambda(X_i)$ for the retrieval stage as:

$$\hat{D}_\lambda(X_i) = \{d_j^i : f(x_j^i) \geq 1 - \lambda\}, \tag{9}$$

where $\lambda \in [0, 1]$ and $f(x_j^i)$ denotes the model score of x_j^i calculated by the retrieval model f. Note that a good retrieved document set $\hat{D}_\lambda(X_i)$ associated with the query q^i should aim to cover as many relevant documents in Y_i as possible. To measure the miscoverage of Y_i by $\hat{D}_\lambda(X_i)$, we define retrieval loss function l as:

$$l(\hat{D}_\lambda(X_i), Y_i) = 1 - \frac{|Y_i \cap \hat{D}_\lambda(X_i)|}{|Y_i|}. \tag{10}$$

The goal of conformal retrieval control is to ensure $\mathbb{E}[l(\hat{D}_{\hat{\lambda}}(X_{n+1}), Y_{n+1})] \leq \alpha$ for a predefined first-stage risk level $\alpha \in [0, 1]$, by determining the threshold value of the parameter $\hat{\lambda} \in [0, 1]$. Observe that a higher value of λ results in a larger retrieval set $\hat{D}_\lambda(X_i)$ and, consequently, a smaller retrieval risk, indicating the retrieval risk is non-increasing in λ. Furthermore, it is evident that the loss function defined in Equation (10) is right-continuous in λ and bounded by $[0, 1]$. By verifying the conditions outlined in Theorem 1 in (Angelopoulos et al., 2024), it becomes apparent that if the retrieval set is constructed using the threshold $\hat{\lambda}$ as calculated by Equation (5), the expected retrieval risk for a new query can be controlled at level α.

3.2 Conformal Ranking Control

We treat documents in Z_i interchangeably for their ranking positions. To obtain a prediction set for Z_i, one can employ the ranking model g on the retrieved document set $\hat{D}_\lambda(X_i)$ and ranking positions for the prediction set can be obtained through their model scores from g in descending order. Now, for a parameter pair $(\lambda, \gamma) \in [0, 1] \times [0, 1]$, we define the prediction set $\hat{D}_{\lambda,\gamma}$ as:

$$\hat{D}_{\lambda,\gamma}(X_i) = \{d_j^i : f(x_j^i) \geq 1 - \lambda, g(x_j^i) \geq 1 - \gamma\}, \tag{11}$$

where $g(x_j^i)$ is the model score of x_j^i obtained from the ranking model g. To measure the ranking quality, we used the popular ranking metric nDCG (Järvelin and Kekäläinen, 2000), though other metrics may be considered. We slightly adapted the traditional definition of nDCG to fit for our setting. The Discounted Cumulative Gain (DCG) is defined as:

$$\text{DCG}\left(\hat{D}_{\lambda,\gamma}(X_i), Z_i\right) = \sum_{d_{(j)} \in \hat{D}_{\lambda,\gamma}(X_i)} \frac{\mathbf{1}(d_{(j)} \in Z_i)}{\log(j+1)}, \tag{12}$$

where $d_{(j)}$ is the document ranked at the j^{th} position in $\hat{D}_{\lambda,\gamma}(X_i)$. Notably, when $\hat{D}_{\lambda,\gamma}(X_i)$ equals the whole retrieved document set $\hat{D}_\lambda(X_i)$, DCG attains its maximum value. The Ideal Discounted Cumulative Gain (iDCG) for query q^i is defined as:

$$\text{iDCG}\left(Z_i\right) = \sum_{j=1}^{|Z_i|} \frac{1}{\log(j+1)}, \tag{13}$$

where $|Z_i|$ is the size of Z_i. Note that this value is only achievable in a perfect retrieval scenario where all relevant documents for relevance level r_0 and above are retrieved. Correspondingly, the Normalized Discounted Cumulative Gain (nDCG) is defined as follows:

$$\text{nDCG}\left(\hat{D}_{\lambda,\gamma}(X_i), Z_i\right) = \frac{\text{DCG}\left(\hat{D}_{\lambda,\gamma}(X_i), Z_i\right)}{\text{iDCG}(Z_i)}. \tag{14}$$

We can verify that $\text{nDCG}(\hat{D}_{\lambda,\gamma}(X_i), Z_i) \in [0,1]$. Accordingly, we define the loss function as:

$$\tilde{l}(\hat{D}_{\lambda,\gamma}(X_i), Z_i) = 1 - \text{nDCG}(\hat{D}_{\lambda,\gamma}(X_i), Z_i). \tag{15}$$

Given λ specified in the first stage, the ranking loss is a function of γ. We note that this loss function is bounded in $[0,1]$ and non-increasing in γ. Recall that the aim of conformal ranking control is to ensure $\mathbb{E}[\tilde{l}(\hat{D}_{\lambda,\gamma}(X_{n+1}), Z_{n+1})] \leq \beta$ for a specified second-stage risk level $\beta \in [0,1]$.

4 Experiments and Results

We validate our methods on two widely-used public datasets for ranked retrieval tasks: the MSLR-WEB dataset (10K)[1](Qin and Liu, 2013), and the MS MARCO Question Answering dataset (V2.1)[2] (Bajaj et al., 2016). Due to space constraints, we present the results on the MSLR-WEB dataset in this section, while leaving the results on the MS MARCO dataset in the technical appendix.

For both tasks, models are trained on a hold-out training set, where we do not particularly tune the model for optimal ranking results, as it is not the primary focus of this work. Subsequently, we randomly split the remaining labeled data into validation and test sets using a 50-50 ratio. We run the experiment for 100 times and report the averaged results. Our code is available at https://github.com/git4review/conformal_ranked_retrieval.

In the experiments, to investigate the impact of different values of α and β, we firstly fixed $\beta = 0.2$ and varied the α values, then fixed $\alpha = 0.3$ and varied the β values. For each pair of (α, β), we compare the following four methods:

- CRC: our proposed two-stage conformal risk control method
- LTT: the multi-risk version of the Learn-Then-Test method, as described by Angelopoulos et al. (2021) in section 2.4
- AdHoc CRC: applying the conventional conformal risk control method independently for each of the two stages
- AdHoc LTT: applying the conventional Learn-Then-Test method sequentially for the two stages

More detailed setup information about these methods can be found in the technical appendix: Experiments Setup.

4.1 MSLR-WEB dataset

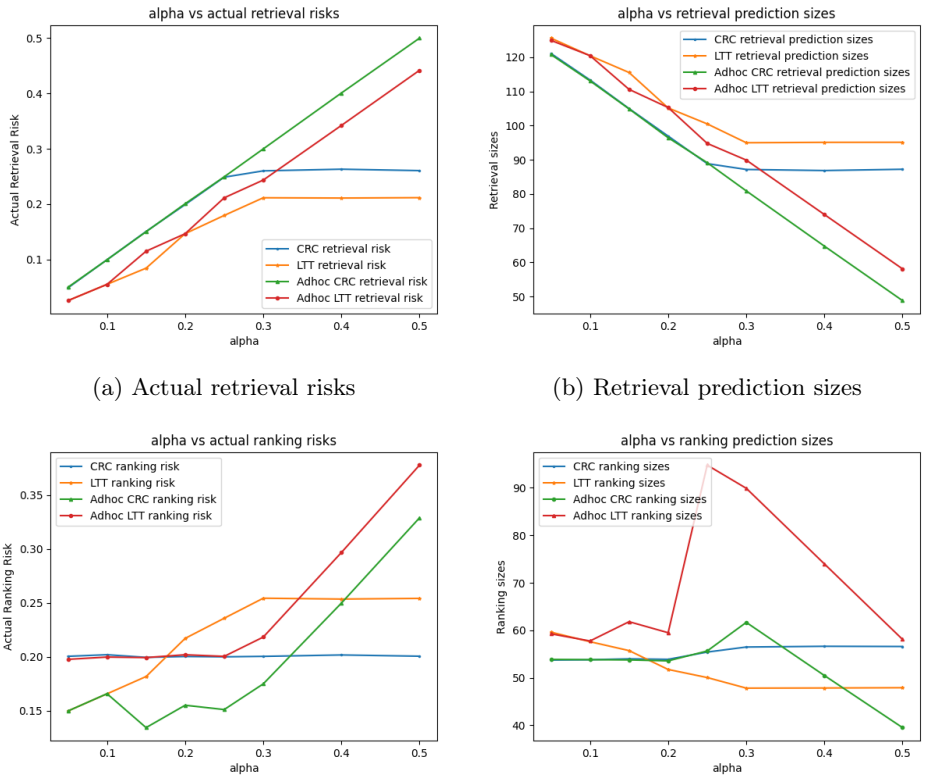
The MSLR-WEB dataset, a large-scale Learning-to-Rank dataset released by Microsoft Research, is curated through a commercial web search engine (Microsoft Bing). It comprises 10K queries, each associated with an average of 120 documents. The dataset consists of 136-dimensional feature vectors extracted from query-url pairs, accompanied by human-assigned relevance judgment labels ranging from 0 (irrelevant) to 4 (perfectly relevant).

For the L1 model, i.e., the retrieval model, we train a 3-layer MLP, with 128 and 32 neurons in the hidden layers. For the L2 model, i.e., the ranking model, we utilize the LambdaRank model (Burges et al., 2006) implemented by the open-source Pytorch package PT-Rank (Yu, 2020). This choices of the model emulates

[1]Available at www.microsoft.com/en-us/research/project/mslr
[2]Available at https://microsoft.github.io/msmarco/

real search engine practices, where the retrieval stage employs a lightweight model for efficiency, while the ranking stage adopts a more sophisticated model for quality.

Figure 2 shows the results on different α values while fixing $\beta = 0.2$. It can be seen that the retrieval risks for both CRC and LTT firstly increase with α when α is small, then become flattened when α is large, in order to control their ranking risks. The Ad-hoc CRC and Ad-hoc LTT retrieval risks, on the other hand, monotonically increase with α throughout the range, resulting the uncontrolled ranking risks. Between CRC and LTT, the former controls the ranking risk at its exact target value 0.2, while the latter provides a more strict retrieval risk control than the former, therefore produces a larger retrieval size. LTT also misses the ranking risk control target by a considerable margin.



(a) Actual retrieval risks

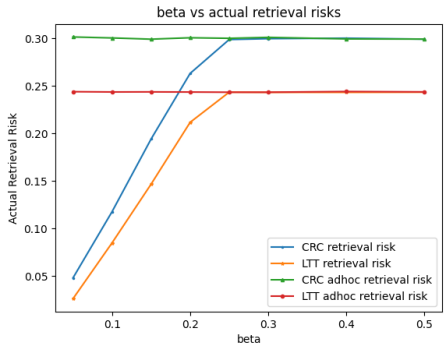
(b) Retrieval prediction sizes

(c) Actual ranking risks

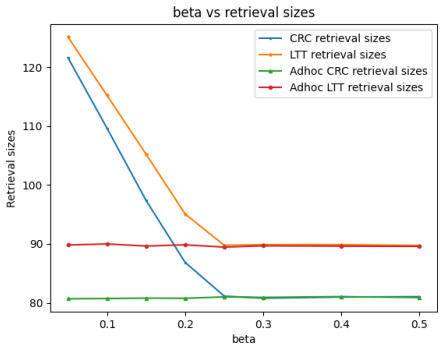
(d) Ranking prediction sizes

Figure 2: Risk and prediction sizes for different α at a fixed $\beta = 0.2$ on MSLR-WEB dataset.

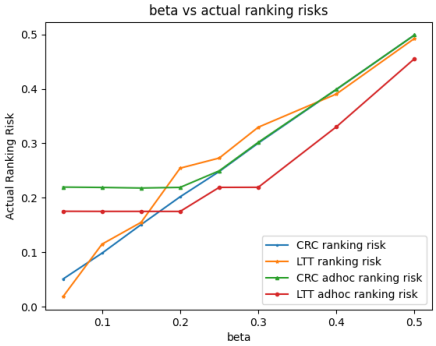
Figure 3 shows the results on different β values while fixing $\alpha = 0.3$. It can be seen that the retrieval risks for both CRC and LTT firstly increase with β, before becoming flattened at the fixed α value 0.3. Between the two methods, LTT has a smaller retrieval risk than CRC, resulting in a larger prediction size. Consequently, the ranking risks of CRC is well controlled by their target values as shown by the blue diagonal line, while the ranking risk of LTT is loosely controlled. On the other hand, however, both Ad-hoc methods control their retrieval risks at the targeted value 0.3, while fail to control their ranking risks for small β values.



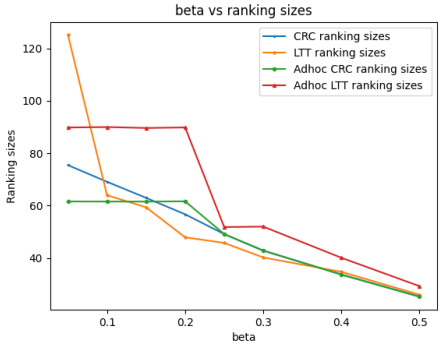
(a) Actual retrieval risks



(b) Retrieval prediction sizes



(c) Actual ranking risks



(d) Ranking prediction sizes

Figure 3: Risk and prediction sizes for different β at a fixed $\alpha = 0.3$ on MSLR-WEB dataset.

5 Conclusion & Discussion

We extended the recently developed single-stage conformal risk control framework to a more complex and widely applicable two-stage setting, creating a new two-stage conformal risk control method. We then applied this method to a typical ranked retrieval problem. The method adjusts its prediction set sizes for different queries, while ensuring that retrieval and ranking risks remain within predefined bounds. Empirical results from real-world tasks underscore the effectiveness of the proposed method.

Several pertinent questions warrant further exploration. In our current research, we exclusively employed nDCG to measure ranking quality, given its prevalence in numerous practical ranked retrieval tasks. However, there exist various other ranking metrics, such as MAP (mean average precision) and MRR (mean reciprocal rank), among others. Some metrics may not adhere to the monotonic risk condition required for the application of our control algorithm. It would be interesting to explore how to define proper risk functions for those metrics, or to adapt the risk control framework so that it is applicable to more general risk functions. We intend to investigate these questions in our future work.

References

A.N. Angelopoulos, K. Krauth, S. Bates, Y. Wang, and M.I. Jordan. Recommendation systems with distribution-free reliability guarantees. In *Symposium on Conformal and Probabilistic Prediction with Applications (COPA), 2023*, 2023.

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv:2107.07511*, 2021.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *arXiv preprint arXiv:2110.01052*, 2021.

Anastasios N Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. *ICLR*, 2024.

R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

C. J. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, 2005.

Christopher J.C. Burges, Robert Ragno, and Quoc Viet Le. Learning to rank with nonsmooth cost functions. In *Proceedings of NIPS conference, 2006*, 2006.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *MSR-TR-2007-40*, 2007.

W. Chu and Z. Ghahramani. Preference learning with gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, 2005.

Koby Crammer and Yoram Singer. Pranking with ranking. In *Proceedings of NIPS conference*, 2001.

Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. In *Journal of Machine Learning Research*, 2003.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 39th International ACM SIGIR conference*, 2016.

Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. *Proceedings of the 23rd international ACM SIGIR conference*, 2000.

Omar Khattab, Mohammad Hammoud, and Tamer Elsayed. Finding the best of both worlds: Faster and more robust top-k document retrieval. *Proceedings of the 43rd International ACM SIGIR Conference*, 2020.

Jing Lei, Alessandro Rinaldo, and Larry Wasserman. A conformal prediction approach to explore functional data. *Annals of Mathematics and Artificial Intelligence*, 2015.

Tie-Yan Liu. Learning to rank for information retrieval. *Proceedings of the 33rd international ACM SIGIR conference*, 2009.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *ECML*, 2002.

Tao Qin and Tie-Yan Liu. Introducing LETOR 4.0 datasets. *CoRR*, abs/1306.2597, 2013. URL http://arxiv.org/abs/1306.2597.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference*, 2015.

Robertson. Stephen and Jones. K., Sparck. Relevance weighting of search terms. journal of the association for information science and technology. *27(3):129-146. doi: 10.1002/ASI.4630270302*, 1976.

Vladimir Vovk, Alex Gammerman, and Craig Saunders. Machine-learning applications of algorithmic randomness. *Sixteenth International Conference on Machine Learning (ICML-1999)*, 1999.

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

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Dawei Yin and et al. Ranking relevance in Yahoo search. *Proceedings of the ACM SIGKDD Conference*, 2016.

Hai-Tao Yu. PT-Ranking: A benchmarking platform for neural learning-to-rank, 2020.

A Theoretic Result

A.1 Data splitting approach

To obtain finite sample risk control guarantee, we consider randomly splitting dataset $\mathcal{D} = \{(L_i(\lambda), \tilde{L}_i(\lambda, \gamma))\}_{i\in[n]}$ into $\mathcal{D}_1 = \{(L_i(\lambda), \tilde{L}_i(\lambda, \gamma))\}_{i\in\mathcal{I}_1}$ and $\mathcal{D}_2 = \{(L_i(\lambda), \tilde{L}_i(\lambda, \gamma))\}_{i\in\mathcal{I}_2}$ with $|\mathcal{I}_1| = n_1$, $|\mathcal{I}_2| = n_2$, and $n_1 + n_2 = n$. Furthermore, we assume $\min\{\alpha, \beta\}(n_1 + 1) \geq 1$ and $\beta(n_2 + 1) \geq 1$. Accordingly, define

$$\hat{\lambda}_1(\mathcal{I}_1) = \inf\left\{\lambda : \sum_{i\in\mathcal{I}_1} L_i(\lambda) \leq (n_1 + 1)\alpha - 1\right\},$$
$$\hat{\lambda}_2(\mathcal{I}_1) = \inf\left\{\lambda : \sum_{i\in\mathcal{I}_1} \tilde{L}_i(\lambda, 1) \leq (n_1 + 1)\beta - 1\right\}.$$

Additionally, if $\sum_{i\in\mathcal{I}_2} \tilde{L}_i(\lambda, 1) \leq (n_2 + 1)\beta - 1$, define

$$\hat{\gamma}(\lambda, \mathcal{I}_2) = \inf\left\{\gamma : \sum_{i\in\mathcal{I}_2} \tilde{L}_i(\lambda, \gamma) \leq (n_2 + 1)\beta - 1\right\},$$

and set $\hat{\gamma}(\lambda, \mathcal{I}_2) = 2$ otherwise. Subsequently, we can define the corresponding feasible set as

$$\begin{aligned}\hat{\Omega}^*_{\mathrm{S}} = \{(\lambda, \gamma) :&\lambda \in [\max\{\hat{\lambda}_1(\mathcal{I}_1), \hat{\lambda}_2(\mathcal{I}_1)\}, 1],\\ &\hat{\gamma}(\lambda, \mathcal{I}_2) \leq 1, \gamma \in [\hat{\gamma}(\lambda, \mathcal{I}_2), 1]\}.\end{aligned} \tag{16}$$

Theorem 1. *Under the same set of assumptions as in Proposition 1, for any parameter pair $(\lambda, \gamma) \in \hat{\Omega}^*_S$, we have*

$$\mathbb{E}L_{n+1}(\lambda) \leq \alpha \quad \text{and} \quad \mathbb{E}\tilde{L}_{n+1}(\lambda, \gamma) \leq \beta.$$

Proof. Consider $(\lambda, \gamma) \in \hat{\Omega}^*_{\mathrm{S}}$. By definition, $\lambda \geq \hat{\lambda}_1(\mathcal{I}_1) \vee \hat{\lambda}_2(\mathcal{I}_1)$. According to Theorem 1 in Angelopoulos et al. (2024), we have

$$\mathbb{E}L_{n+1}(\hat{\lambda}_1(\mathcal{I}_1)) \leq \alpha \quad \text{and} \quad \mathbb{E}\tilde{L}_{n+1}(\hat{\lambda}_2(\mathcal{I}_1), 1) \leq \beta.$$

Consequently, we have $\mathbb{E}L_{n+1}(\lambda) \leq \alpha$ and $\mathbb{E}\tilde{L}_{n+1}(\lambda, 1) \leq \beta$ by the monotonicity of the loss functions. Moreover, by the tower property of conditional expectation, we have

$$\mathbb{E}\tilde{L}_{n+1}(\lambda, \hat{\gamma}(\lambda, \mathcal{I}_2)) = \mathbb{E}\left\{\mathbb{E}\left\{\tilde{L}_{n+1}(\lambda, \hat{\gamma}(\lambda, \mathcal{I}_2))|\mathcal{D}_1\right\}\right\}.$$

By conditioning on the $\mathcal{D}_1$, λ can be viewed as a constant. With a direct application of Theorem 1 in Angelopoulos et al. (2024), we have

$$\mathbb{E}\left\{\tilde{L}_{n+1}(\lambda, \hat{\gamma}(\lambda, \mathcal{I}_2))|\mathcal{D}_1\right\} \leq \beta.$$

Therefore, by the monotonicity of the loss function, we conclude

$$\mathbb{E}\tilde{L}_{n+1}(\lambda, \gamma) \leq \mathbb{E}\tilde{L}_{n+1}(\lambda, \hat{\gamma}(\lambda, \mathcal{I}_2)) \leq \beta. \tag{17}$$

B Details on Simulation & Experiments Setup

This section provides detail of settings used for the simulation and experiments.

B.1 Simulation Setup

We present a simulation[3] of a simple two-stage risk control task on a 2D synthetic dataset consisting of two classes: $\mathcal{C}_1$ and $\mathcal{C}_2$. Each class contains 2,000 data points sampled from distinct two-dimensional Gaussian distributions with an identity covariance matrix $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Class $\mathcal{C}_1$ has a mean vector $(-1, 0)$, while class $\mathcal{C}_2$ has a mean vector $(1, 0)$, as shown in Figure 4. For each data $x_i = (x_{i1}, x_{i2}) \in \mathcal{C}_1$, except for its class label y_i, we further associate a continuous response value z_i that equals to its distance to the centroid $(-1, 0)$, i.e., $z_i = \sqrt{(x_{i1} + 1)^2 + x_{i2}^2}$, as indicated by its color. Data points in $\mathcal{C}_2$ are depicted in gray color.

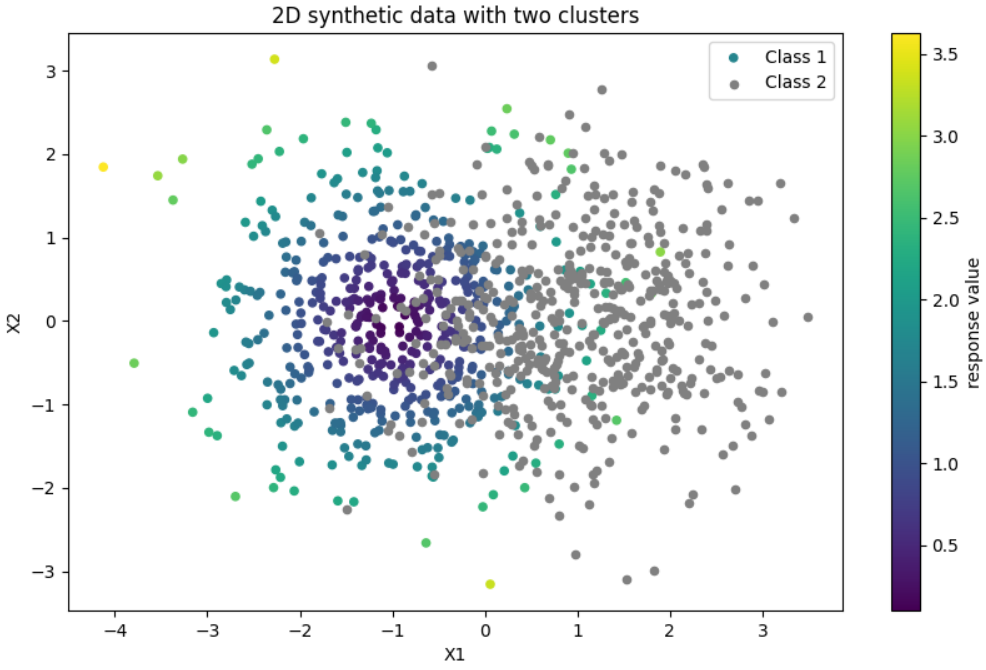


Figure 4: 2D synthetic data

For the first stage, to classify the two classes, we train a MLP with three layers, where each of the two linear hidden layers has 16 neurons. The model is trained using cross entropy loss with 10 iterations at a learning rate 0.001.

For the second stage, to predict the response value, we train a MLP with three layers, where each of the two linear hidden layers has 64 neurons. The model is trained using the mean squared error loss with 20 iterations at a learning rate 0.001.

To control the first stage risk, during the test, a test data is classified as the class $\mathcal{C}_1$ if its model score $f(x_i) \geq 1 - \lambda$, where λ is the first stage conformal parameter. The first stage risk is the false negative rate of class $\mathcal{C}_1$.

To control the second stage risk, for a test data x_i, in addition to predict its response value $\hat{z}_i$, we also produce a predicted response range $\tilde{C}(x_i) = [\hat{z}_i - \gamma, \hat{z}_i + \gamma]$, where γ is the second stage conformal parameter. The risk used is the mis-coverage rate of the labeled response z_i by $\tilde{C}(x_i)$.

[3]A notebook file for this simulation can be found at https://github.com/git4review/conformal_ranked_retrieval.

We use a 7:3 ratio to randomly split the data into validation and test datasets, and the results are the average over 100 random runs.

B.2 Experiments Setup

In our experiments with real-world data, we compared our method, CRC, with three other approaches: LTT, AdHoc CRC, and AdHoc LTT. This section provides details on the setup of these three methods.

The LTT method, developed by Angelopoulos et al. (2021), is a multiple hypothesis testing framework for risk control. To adapt it for a two-stage setting, we follow the "Multiple risks and multi-dimensional" formulation discussed in section 2.4 of their work, where we define the null hypothesis as

$$\mathcal{H}_j = \{L(\lambda_j) > \alpha \textbf{ or } \tilde{L}(\lambda_j, \gamma_j) > \beta\},$$

and apply a family-wise error rate (FWER) controlling procedure to test $\mathcal{H}_j$.

Specifically, we employed the fixed sequence testing approach to control the family-wise error rate (FWER). This involved performing a linear search on a two-dimensional parameter grid, where the range $[0, 1]$ for each parameter was discretized into 100 equi-spaced elements. We then evaluated the two-stage risks at each of the 10,000 points on the grid. To calculate the p-value, we use the Hoeffding-Bentkus inequality, and for FWER control, we use the Bonferroni correction.

For the two Ad-hoc methods, we essentially treat them as two sequential single-stage risk control problems. Specifically, we firstly calculate the $\hat{\lambda}$ that controls the first stage risk. With this value, we build the prediction set $C_{\hat{\lambda}}(X_i)$ for each $(X_i, Y_i, Z_i) \in D$ as the validation input for the second stage, then we apply the same control algorithm to calculate $\hat{\gamma}$. Similarly, for Ad-hoc LTT, we discritized the range $[0, 1]$ into 100 equi-spaced elements for both λ and γ to faciliate the fixed sequence testing, and we also use the Hoeffding-Bentkus inequality to calculate the p-value, as well as the Bonferroni correction for FWER control.

C Experiment Results on MS MARCO Dataset

MS MARCO (MicroSoft MAchine Reading COmprehension) (Bajaj et al., 2016) is a collection of large-scale datasets focused on machine reading comprehension. For our experiment, we opted for the Question and Answering v2.1 task within this dataset. This dataset comprises approximately 100K queries and 1M passages, with each query has 10 candidate passages. The objective is to select the most relevant passage from the provided candidates to answer the corresponding query. Each passage is associated with a binary flag is_selected, where 1 denotes it is a good answer and 0 otherwise.

In the retrieval (L1) stage, we utilized Okapi BM25 (Stephen and K., 1976), which provides a TF-IDF like score function to measure the relevance between the query and the document. In our work, we follow the definition by (Robertson and Zaragoza, 2009). Specifically, for a given query q with keywords $w_1, \ldots, w_n$, the BM25 score of a candidate document D is defined as:

$$\text{BM25}(q, D) = \sum_{i=1}^{n} \frac{\text{IDF}(w_i) f(w_i, D) \cdot (k_1 + 1)}{f(w_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgdl})},$$

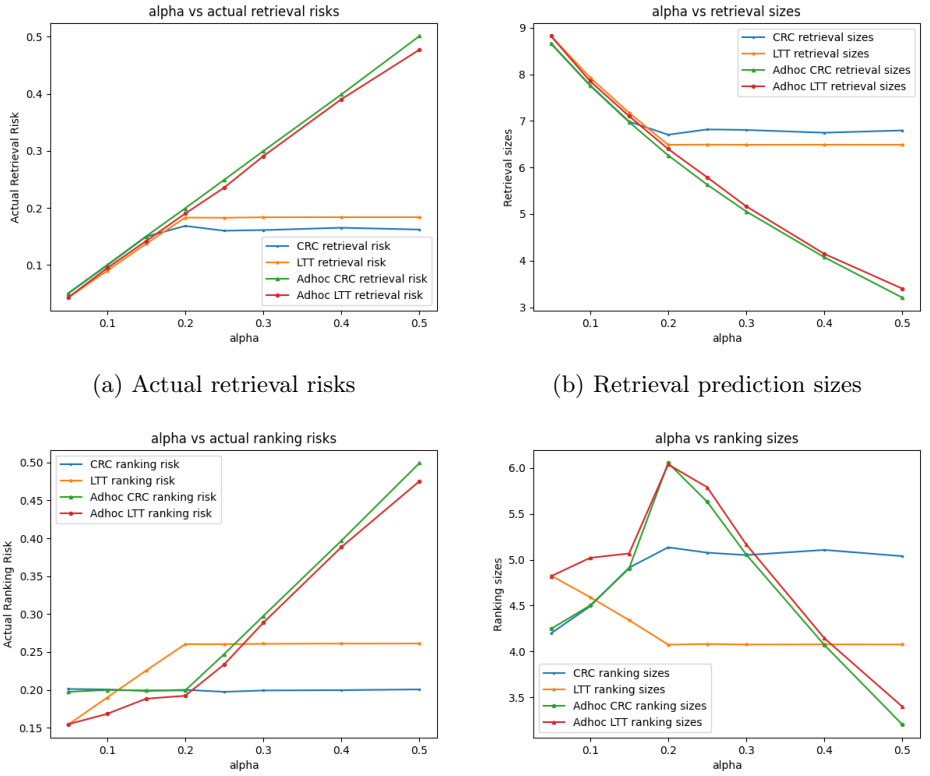
where $f(w_i, D)$ is the frequency of w_i in D, $|D|$ is the total number of words in D, $avgdl$ is the average length of the documents in the repository, k_1 and b are hyper-parameters. In our experiments, we choose $k_1 = 2$, $b = 0.75$. The $IDF(w_i)$ is the inverse document frequency of w_i, defined as

$$\text{IDF}(w_i) = \ln\left(\frac{N - n(w_i) + 0.5}{n(w_i) + 0.5} + 1\right),$$

where N is the total number of documents in the repository, and $n(w_i)$ is the number of documents containing w_i.

For the ranking (L2) stage, we choose the e5-base model, a 12-layer encoder large language model trained through a weakly supervised constrastive learning on a large scale text pair dataset CCPairs (Wang et al., 2022). This model is much more resource-intensive to deploy and run compared to a simple score function such as BM25. In our experiment, we leverage the HuggingFace transformer package[4] to encode both the query and the document. We choose a maximum window size 512 for tokenization, and use an average pooling on the last hidden layer to produce the final 768 dimentional embedding vectors. Subsequently, we calculate the cosine similarity between the query embedding and the document embedding to obtain the relevance score for ranking the documents.

Figure 6 shows the results on different β values while fixing $\alpha = 0.3$, while Figure 5 shows the results on different α values while fixing $\beta = 0.2$, with similar observations as on the MSLR-WEB dataset.



(a) Actual retrieval risks

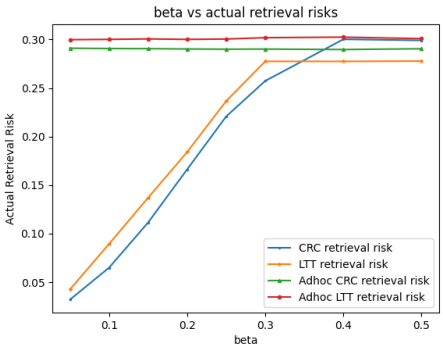
(b) Retrieval prediction sizes

(c) Actual ranking risks

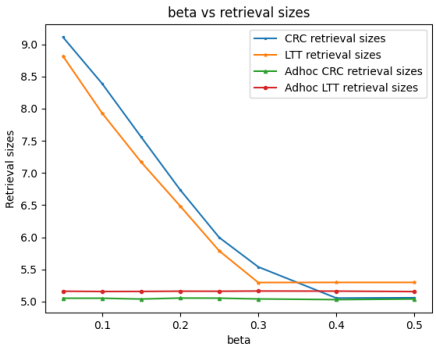
(d) Ranking prediction sizes

Figure 5: Risk and prediction sizes for different α at a fixed $\beta = 0.2$ on MS MARCO dataset.

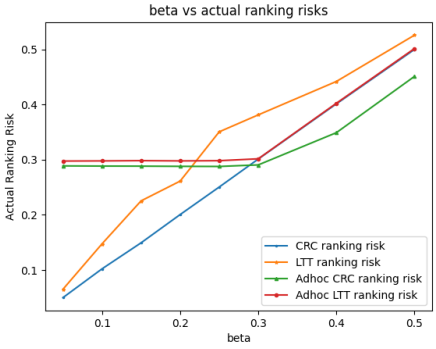
[4]Available at https://huggingface.co/intfloat/e5-base.



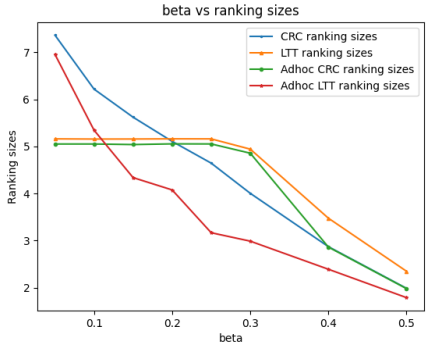
(a) Actual retrieval risks



(b) Retrieval prediction sizes



(c) Actual ranking risks



(d) Ranking prediction sizes

Figure 6: Risk and prediction sizes for different β at a fixed $\alpha = 0.3$ on MS MARCO dataset.