

# Learn then Test: Calibrating Predictive Algorithms to Achieve Risk Control

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, Lihua Lei

April 27, 2022

## Abstract

We introduce a framework for calibrating machine learning models so that their predictions satisfy explicit, finite-sample statistical guarantees. Our calibration algorithm works with any underlying model and (unknown) data-generating distribution and does not require model refitting. The framework addresses, among other examples, false discovery rate control in multi-label classification, intersection-over-union control in instance segmentation, and the simultaneous control of the type-1 error of outlier detection and confidence set coverage in classification or regression. Our main insight is to reframe the risk-control problem as multiple hypothesis testing, enabling techniques and mathematical arguments different from those in the previous literature. We use our framework to provide new calibration methods for several core machine learning tasks with detailed worked examples in computer vision and tabular medical data.

## 1 Introduction

Learning algorithms are used increasingly in real-world systems despite insufficient testing and poor understanding of likely failure modes. They are often vetted only by measuring their accuracy on stylized benchmarks of limited scope. Trustworthy decision making requires more extensive notions of reliability: assessing when, how often, and how badly predictions will fail.

A major challenge in this regard is that contemporary automated systems are based on ever-evolving learning algorithms, are being deployed in novel architectural configurations, and make use of new kinds of data formats. The statistical literature is rich with time-tested modeling tools backed up by formal guarantees and mature diagnostics, and these tools have formed the backbone of scientific data analysis for half a century. However, this toolbox is mainly developed for tabular data with relatively simple models. Trying to mimic these successful analysis strategies in our many modern data problems with state-of-the-art predictive models is fraught with difficulty; it would require systems designers to proceed with challenging, case-by-case theoretical derivations in many complex, high-dimensional problems. Absent such analysis, current deployments often resort to rough rules of thumb. For example, state-of-the-art vision systems for object detection are currently being deployed without a statistically calibrated notion of the quality of a model prediction.

Thus, these new data problems demand new statistical methods that endow the entire spectrum of predictive models with performance and uncertainty guarantees. Importantly, the methods should not interfere with the engineering choices that drive performance gains, and must be able to handle complex inputs and outputs, such as video, 3D volumes, natural language data, and so on. In short, we need statistical methods to put modern predictive models on firm ground.

This paper introduces a framework called *Learn then Test* (LTT) that bestows such finite-sample guarantees on any predictive model, without assumptions on the inner workings of the model or the true distribution underlying the dataset. See Figure 1 for a real example of an object detector annotated with representations of the statistical guarantees that we will develop herein. Our method is intuitive and simple: a model learns to make predictions which we modify with multiple hypothesis testing to rigorously select a set of parameter values to control any user-chosen statistical error rate. Put plainly, we learn a base model and then test which parameter values lead to risk control. Thus, we link the problem of distribution-free predictive inference with the well-developed ideas in multiple hypothesis testing, allowing us to bring the latter to bear to ensure the reliability of machine learning models.



Figure 1: **Object detection with simultaneous distribution-free guarantees** on the expected intersection-over-union, recall, and coverage rate is possible with our methods; see Section 7 for details.

## 1.1 Motivating examples

We begin with an informal definition of our setting and goals, then give three practical instances of its application before introducing formal notation in Section 1.3. Abstractly, we receive a pre-trained model  $\hat{f}$  that takes inputs  $X$  and predicts an output  $Y$ . The model also has a low-dimensional parameter  $\lambda$  that affects its predictions, which we are free to choose, such as the binarization threshold of a segmentation algorithm. The procedures we describe give the user many possible choices for  $\lambda$  that are guaranteed to control a risk function  $R$ —i.e. a well defined performance metric—with high probability, ensuring that the model’s prediction is reliable. If the risk cannot be controlled at the requested level because the underlying model is not good enough, the procedure will say so. The choices of  $\lambda$  are determined via a multiple testing procedure described in detail in Section 2. We give three examples of this abstract selection process below, in order of increasing intricacy.

- *Multi-label classification:* A **learned** multiclass classification model  $\hat{f}$  outputs a probability that each of  $K$  classes is in an image. The parameter  $\lambda$  is a threshold for the inclusion of a class in the predicted set of classes. Using fixed-sequence **testing** we select a  $\lambda$  that is guaranteed to control the false discovery rate (FDR) risk  $R$  of the predicted set of classes at the user’s chosen level.
- *Segmentation:* A **learned** segmentation model  $\hat{f}$  outputs a probability that each pixel in the image comes from an object. The parameter  $\lambda$  is the binarization threshold that ultimately controls the size of the predicted mask. Using a specialized form of sequential graphical **testing**, we can certify a set of  $\lambda$  guaranteed to control the intersection-over-union (IOU) risk  $R$  of the predicted mask with the ground truth mask at the user’s chosen level.
- *Instance segmentation:* A **learned** object detector  $\hat{f}$  proposes a set of objects in an image, along with segmentation masks and a prediction set of classes for each object. The parameter  $\lambda$  is a vector thresholding which predictions are confident enough to be displayed, the binarization level of the masks, and how many classes are included in the prediction set. We can find a set of  $\lambda$  guaranteed to control the recall, IOU, and coverage jointly at the user’s chosen levels via multiple **testing**.

We give a detailed treatments of these three specific problems in this work. Note that these examples are uniquely possible using our method, when compared to existing literature, because all previous work in distribution-free uncertainty quantification, such as conformal prediction [1] and risk-controlling prediction sets [2], have required that  $\lambda$  is one-dimensional and the risk function be monotonic in  $\lambda$ . We do not make this assumption, and therefore we can simultaneously control many risks.

## 1.2 Our contribution

Our main contribution in this work is a calibration procedure for machine learning models that gives statistical control of any risk. For bounded risk functions, our procedure has explicit finite-sample guarantees. In particular,

we recast risk control as a multiple testing problem, and then use ideas from the rich field of multiple testing to create powerful testing algorithms for our setting. The multiple testing problem underlying risk control is challenging in that the many tests are dependent and the hypothesis space has subtle structure, so we put care into constructing a powerful multiple testing procedure. We specialize our framework to four core problems in machine learning: multi-label classification, selective classification, selective regression, out-of-distribution (OOD) detection, and instance segmentation. Our code, available at <https://github.com/aangelopoulos/lrt>, allows exact reproduction of our experiments.

### 1.3 Formalizing our goal

Let  $(X_i, Y_i)_{i=1, \dots, n_{\text{full}}}$  be an independent and identically distributed (i.i.d.) set of variables, where the feature vectors  $X_i$  take values in  $\mathcal{X}$  and the responses  $Y_i$  take values in  $\mathcal{Y}$ . To begin, we split our data into a *training set*  $\mathcal{I}_{\text{train}}$  and a *calibration set*  $\mathcal{I}_{\text{cal}}$  that partition  $\{1, \dots, n_{\text{full}}\}$  and let  $n = |\mathcal{I}_{\text{cal}}|$ . Without loss of generality, we take  $\mathcal{I}_{\text{cal}} = \{1, \dots, n\}$ . We fit a predictive model on the training set  $\mathcal{I}_{\text{train}}$  using an arbitrary procedure, resulting in a function  $\hat{f}$  mapping from  $\mathcal{X}$  to some space  $\mathcal{Z}$  (for example,  $\hat{f}$  may be a neural network with a softmax layer at the end, in which case  $\mathcal{Z}$  is a simplex). The remainder of this paper shows how to create predictions using  $\hat{f}$  that control a risk (i.e., a statistical error rate), regardless of the quality of  $\hat{f}$  or the distribution of the data.

At a high level, we accomplish this by introducing a low-dimensional parameter  $\lambda \in \Lambda$  controlling the model’s predictions. Then, we test the model’s performance on the calibration points  $(X_1, Y_1), \dots, (X_n, Y_n)$  for different values of the parameter  $\lambda$ . Formally, we consider functions  $\mathcal{T}_\lambda : \mathcal{X} \rightarrow \mathcal{Y}'$  for some space  $\mathcal{Y}'$ . We will often choose  $\mathcal{Y}'$  to be  $\mathcal{Y}$  or  $2^{\mathcal{Y}}$  in our applications, though our theory works for other spaces. The reader should think of  $\mathcal{T}_\lambda$  as a post-processing of the base model  $\hat{f}$  that makes predictions in  $\mathcal{Y}'$ . For such a function  $\mathcal{T}_\lambda$ , we define a *risk*  $R(\mathcal{T}_\lambda) \in \mathbb{R}$  that captures a problem-specific notion of the statistical error rate. For convenience, we often use the shorthand  $R(\lambda)$  to stand for  $R(\mathcal{T}_\lambda(X))$ . Our goal in this work is to train a function  $\hat{\mathcal{T}}_\lambda$  based on  $\hat{f}$  and the calibration data in such a way that it achieves the following error-control property:

**Definition 1** (Risk-controlling prediction). *Let  $\hat{\lambda}$  be a random variable taking values in  $\Lambda$ . We say that  $\hat{\mathcal{T}}_\lambda$  is a  $(\alpha, \delta)$ -risk-controlling prediction (RCP) if  $\mathbb{P}(R(\hat{\mathcal{T}}_\lambda) \leq \alpha) \geq 1 - \delta$ .*

The error level  $(\alpha, \delta)$  is chosen in advance by the user. The reader can think of 10% as a representative value of  $\delta$ ; the choice of  $\alpha$  will vary with the choice of risk function. In our work,  $\hat{\lambda}$  will be a function of the calibration data, so the probability in the above definition will be over the randomness in the sampling of the calibration data  $(X_1, Y_1), \dots, (X_n, Y_n)$ .

**Remark 1** (Not all risks can be controlled at any requested level). *Note that some risks cannot be controlled at every level  $\alpha$  for every data-generating distribution. For example, it may be impossible to have a classifier with 90% accuracy due to the noise level in the data-generating distribution or the poor quality of the base model. To accommodate such cases, in our setting we may abstain from returning a function  $\mathcal{T}$  if no risk-controlling function can be certifiably found.*

As a concrete example, consider multi-label classification, where  $Y \subset \{1, \dots, K\}$  for some  $K$ . A relevant notion of risk is the FDR of the prediction  $\mathcal{T}_\lambda(X) \subset \{1, \dots, K\}$ ,

$$\text{FDR}(\mathcal{T}_\lambda) = \mathbb{E} \left[ \frac{|\mathcal{T}_\lambda(X) \setminus Y|}{|\mathcal{T}_\lambda(X)|} \right],$$

with the convention that  $0/0 = 0$ . Note that the FDR can always be controlled by taking  $\mathcal{T}_\lambda(X) = \emptyset$ . Turning to the construction of  $\mathcal{T}_\lambda$ , suppose our base model  $\hat{f}$  outputs a vector in  $[0, 1]^K$ . We take the parameter  $\lambda$  to be the probability threshold above which a class is included in the prediction set:  $\mathcal{T}_\lambda(x) = \{k : \hat{f}_k(x) \geq \lambda\}$ , where  $\hat{f}_k(x)$  is the  $k$ -th entry of  $\hat{f}(x)$ . In this case,  $\lambda$  controls the size of the set, which affects the FDR. Our calibration procedure will identify values of  $\lambda$  that control the FDR at the desired level. We will develop this example in full detail in Section 3 after we introduce our core methodology.

### 1.4 Related work

Predictions with statistical guarantees have been heavily explored in the context of set-valued predictions. This approach dates back at least to tolerance regions (sets that cover a pre-specified fraction of the population distri-

bution) in the 1940s [3–6]. See [7] for a review of this topic. Recently, tolerance regions have been used to form prediction sets with deep learning models [8, 9]. In parallel, conformal prediction [1, 10, 11] has been developed as a way to produce prediction sets with finite-sample statistical guarantees. One convenient, widely-used form of conformal prediction, known as split conformal prediction [12, 13], uses data splitting to generate prediction sets in a computationally efficient way; see also [14, 15] for generalizations that re-use data for improved statistical efficiency. Conformal prediction is a generic approach, and much recent work has focused on designing specific conformal procedures to have good performance according to additional desiderata such as small set sizes [16], coverage that is approximately balanced across regions of feature space [17–23], and errors balanced across classes [16, 24–26]. Recent extensions also address topics such as distribution estimation [27], causal inference [28], survival analysis [29], differential privacy [30], outlier detection [31], speeding up the test-time evaluation of complex models [32, 33], the few-shot setting [34], handling dependent data [35, 36], and handling of testing distribution shift [31, 37–41].

Most closely related to the present work is the technique of Risk-Controlling Prediction Sets [2] which extends tolerance regions and conformal prediction to give prediction sets that control other notions of statistical error. The present work goes beyond that work to consider prediction with risk control more generally, without restricting the scope to confidence sets. This is possible because we solve a key technical limitation of that earlier work—the restriction to monotonic risks—so that the techniques herein apply to any notion of statistical error. The present work moves in the direction of decision-making, which has been only lightly explored in the context of conformal prediction [42, 43], with the existing literature taking a very different approach.

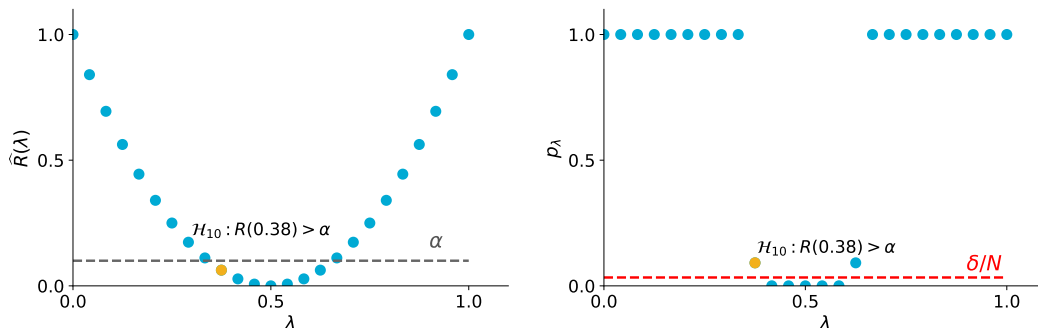
## 2 Risk Control in Prediction

This section introduces our proposed method for controlling the risk of a prediction. Section 2.2 introduces our core methodology, which reframes risk control as a multiple testing problem. In Section 2.3, we make our methodology concrete with a powerful multiple testing procedure that works well in practice for our machine learning examples. In Section 2.4, we show that our framework also handles the case where one seeks to control multiple risks simultaneously. Finally, Section 2.5 describes an alternative approach utilizing uniform concentration which we find to be more conservative than the multiple testing strategy. We begin by grounding our discussion with an informal preview of the LTT procedure.

### 2.1 Overview of the Learn then Test procedure

Recalling Definition 1, our goal is to find a set function whose risk is less than some user-specified threshold  $\alpha$ . To do this, we search across the collection of functions  $\{\mathcal{T}_\lambda\}_{\lambda \in \Lambda}$  and estimate their risk on the calibration data  $(X_1, Y_1), \dots, (X_n, Y_n)$ . The output of the procedure will be a set of  $\lambda$  values,  $\hat{\Lambda} \subseteq \Lambda$  which are all guaranteed to control the risk, abbreviated as  $R(\lambda) = R(\mathcal{T}_\lambda)$ . The Learn then Test procedure is outlined and illustrated below.

1. For each  $\lambda_j$  in a discrete set  $\Lambda = \{\lambda_1, \dots, \lambda_N\}$ , associate the null hypothesis  $\mathcal{H}_j : R(\lambda_j) > \alpha$ . Notice that *rejecting*  $\mathcal{H}_j$  corresponds to selecting  $\lambda_j$  as a point where the risk is controlled.
2. For each null hypothesis, compute a finite-sample valid p-value using a concentration inequality. We discuss how to pick good p-values in Section 2.3.1.



- Return  $\widehat{\Lambda} = \mathcal{A}(\{p_j\}_{j \in \{1, \dots, \Lambda\}})$ , where  $\mathcal{A}$  is an algorithm that controls the familywise-error rate (FWER). For example, the Bonferroni correction yields  $\widehat{\Lambda} = \{\lambda_j : p_j < \frac{\delta}{|\Lambda|}\}$ . In the image above, this is all values of  $\lambda \in \Lambda$  whose p-value falls below the red dotted line. We define the FWER and preview ways to design good FWER-controlling procedures in Sections 2.3.2 - 2.3.4.

The LTT procedure defined above outputs a set  $\widehat{\Lambda}$  satisfying Definition 1. It decomposes risk control into two subproblems: computing p-values and combining them with multiple testing. We will now begin a formal development of LTT, going into detail about both of these subproblems.

## 2.2 Risk control as multiple testing

Recall that we have a family of set-valued predictors  $\mathcal{T}_\lambda : \mathcal{X} \rightarrow \mathcal{Y}'$ , with the parameter  $\lambda$  living in a closed index set  $\Lambda$ . We slightly abuse the notation hereafter by rewriting  $R(\mathcal{T}_\lambda)$  as  $R(\lambda)$ . The goal of this section is to identify a subset  $\widehat{\Lambda} \subseteq \Lambda$  such that all elements of  $\widehat{\Lambda}$  control the risk simultaneously with high probability, so that  $\mathcal{T}_{\widehat{\lambda}}$  is an RCP for any  $\widehat{\lambda} \in \widehat{\Lambda}$ . Eventually, we will select an element  $\lambda \in \widehat{\Lambda}$  to deploy in our predictions. Unlike [2], our method allows  $\Lambda$  to be multidimensional, we require no special structure of  $\mathcal{T}_\lambda$ , and  $\widehat{\Lambda}$  can have more than one element. In practice, there will often be structure on  $\Lambda$ , and so we also design multiple testing methods to leverage various kinds of structure for improved power. Note however, that the methods *always* control error whether or not the motivating structure holds in practice.

Our techniques will build on the concept of family-wise error rate (FWER) control from the multiple testing literature [e.g., 44], which we formalize next. Consider a list of null hypotheses,  $H_j$ ,  $j = 1, \dots, N$ , with associated p-values  $p_j$ . That is,  $p_j$  has a distribution that stochastically dominates the uniform distribution on  $[0, 1]$  for each  $j$  such that  $H_j$  holds. Let the indices of the true nulls be  $J_0 \subset \{1, \dots, N\}$  and those of the non-nulls be  $J_1 = \{1, \dots, N\} \setminus J_0$ . The goal of FWER control is to use the p-values to reject as many of the  $H_j$  (i.e., identify them as non-nulls) as possible while limiting the probability of making any false rejections to be less than a pre-specified level  $\delta$  (e.g.,  $\delta = 0.05$ ). We formalize FWER-controlling algorithms next.

**Definition 2** (FWER-controlling algorithm). *An algorithm  $\mathcal{A}(\{p_j\}_{j=1}^N) \subseteq \{1, \dots, N\}$  producing a family of rejections is an FWER-controlling algorithm at level  $\delta$  if*

$$\mathbb{P}(\mathcal{A}(p_1, \dots, p_N) \subseteq J_1) \geq 1 - \delta,$$

whenever  $p_j$  has a marginal distribution stochastically dominating the uniform distribution  $[0, 1]$  for all  $j \in J_0$ .

Note that the p-values in the above definition may be dependent; we require that the algorithm works even in this case.

Connecting this to our setting, we will associate a p-value to many discrete values of  $\lambda$  and use a FWER-controlling procedure to identify those that control the risk. For each  $\lambda_j \in \Lambda = \{\lambda_1, \dots, \lambda_N\}$ , we consider the null hypothesis that it does *not* control the risk at level  $\alpha$ :

$$H_j : R(\lambda_j) > \alpha, \tag{1}$$

and its associated p-value  $p_j$ . Then taking  $\widehat{\Lambda}$  to be the output of a procedure that controls the FWER will yield risk control, as we formalize next.

**Theorem 1.** *Suppose  $p_j$  has a distribution stochastically dominating the uniform distribution for all  $j$  under  $H_j$ . Let  $\mathcal{A}$  be a FWER-controlling algorithm at level  $\delta$ . Then  $\widehat{\Lambda} = \mathcal{A}(p_1, \dots, p_N)$  satisfies the following:*

$$\mathbb{P}\left(\sup_{\lambda \in \widehat{\Lambda}} \{R(\lambda)\} \leq \alpha\right) \geq 1 - \delta,$$

where the supremum over an empty set is defined as  $-\infty$ . Thus, selecting any  $\lambda \in \widehat{\Lambda}$ ,  $\mathcal{T}_\lambda$  is an  $(\alpha, \delta)$ -RCP.

All proofs are presented in Appendix B. This result is straightforward but can be used to great effect: the user can use any FWER-controlling procedure to find  $\widehat{\Lambda}$ , and then may pick any  $\lambda \in \widehat{\Lambda}$  as their chosen RCP (even in a

data-driven way). For example, we can choose the one that maximizes another performance metric estimated using the same data without concern about double-dipping. For example, in the FDR case, we would want to choose  $\hat{\lambda} = \sup \hat{\Lambda}$ , which yields the most discoveries and hence the lowest false negative rate (FNR).

Theorem 1 reduces the problem of risk control into two subproblems: first, generating a p-value for each hypothesis, and second, combining the hypotheses to discover the largest set that controls the risk at level  $\alpha$ . To generate p-values, we leverage the hybridized Hoeffding-Bentkus bound from [2]. This solves the first subproblem. To solve the second, we apply two FWER-controlling methods for selecting  $\hat{\Lambda}$ , which improve upon Bonferroni’s and Holm’s procedure [45] for our purpose, in Algorithms 1 and 2.

The multiple testing problem that arises here presents unique challenges. First, nearby p-values in most settings are highly dependent since many of the functions adjacent  $T_{\lambda_j}$  are similar to each other and the p-values are computed with the same set of calibration data. Second, the set of hypothesis  $\Lambda$  is structured, but not always in a simple way. For example  $\Lambda$  may be a two-dimensional grid, where we expect that the risks are approximately increasing along the first axis when holding the second coordinate fixed. Third, the power of FWER-controlling procedures in our settings is not measured by the number of rejections as in the traditional multiple testing literature. Instead, a procedure is powerful even if it only rejects one hypothesis provided that the corresponding  $\lambda$  yields tight risk control. These three aspects make our multiple testing problem unique.

## 2.3 Multiple testing mechanics

We next outline three different multiple testing subroutines to generate the set  $\hat{\Lambda}$  from Theorem 1, each of which leads to risk-controlling predictions. We begin with the simplest such method—Bonferroni—and progress to the most complex—sequential graphical testing (SGT). Each of these three methods takes p-values as input, so we first explain how valid finite-sample p-values are obtained, before moving on to the multiple testing subroutines.

### 2.3.1 Calculating valid p-values

To carry out the multiple testing procedure, we will rely on p-values, so we first explain how these are obtained. We consider the special case where the risk function is the expectation of a *loss* function  $L$ :  $R(\mathcal{T}) = \mathbb{E}[L(\mathcal{T}(X), Y)]$ . For example, the false discovery rate is the expectation of the false discovery proportion. This restriction is *only* for the purpose of computing p-values, and our overall framework is not limited to this case; there are many other cases for which p-values are available.

Beginning with the bounded case where  $L(\mathcal{T}(X), Y) \in [0, 1]$ , we apply the hybridized Hoeffding-Bentkus (HB) inequality from [2], which combines the celebrated results from Hoeffding [46] and Bentkus [47]. The Hoeffding-Bentkus inequality will be a function of the empirical risk on our calibration set,

$$\hat{R}_j = \frac{1}{n} \sum_{i=1}^n L_{i,j}, \quad L_{i,j} = L(T_{\lambda_j}(X_i), Y_i).$$

**Proposition 1** (Hoeffding-Bentkus inequality p-values). *Define the quantity*

$$p_j^{\text{HB}} = \min \left( \exp\{-nh_1(\hat{R}_j \wedge \alpha, \alpha)\}, e\mathbb{P}(\text{Bin}(n, \alpha) \leq \lceil n\hat{R}_j \rceil) \right), \quad (2)$$

where

$$h_1(a, b) = a \log \left( \frac{a}{b} \right) + (1 - a) \log \left( \frac{1 - a}{1 - b} \right).$$

Then  $p_j^{\text{HB}}$  is a valid p-value for  $H_j$ : for all  $u \in [0, 1]$ ,  $\mathbb{P}(p_j \leq u) \leq u$ .

The Hoeffding-Bentkus inequality provides finite-sample statistical results with surprising empirical effectiveness, as we will see later.

For the unbounded case where the Hoeffding-Bentkus inequality no longer applies, one can choose to apply the central limit theorem, which requires a second-moment assumption and does not provide finite-sample validity, and

uses the empirical standard deviation  $\hat{\sigma}_j = \sqrt{\sum_{i=1}^n (L_{i,j} - \hat{R}_j)^2 / (n - 1)}$ .

**Proposition 2** (CLT p-values). *Suppose  $L(T_{\lambda_j}(X), Y)$  has a finite mean and variance. Then,*

$$p_j^{\text{CLT}} = 1 - \Phi\left(\frac{\alpha - \widehat{R}_j}{\widehat{\sigma}_j}\right)$$

*is an asymptotically valid p-value: for all  $u \in [0, 1]$ ,  $\limsup_{n \rightarrow \infty} \mathbb{P}(p_j^{\text{CLT}} \leq u) \leq u$ .*

### 2.3.2 The Bonferroni correction

With p-values in hand, we now turn to the question of forming the rejection set  $\widehat{\Lambda}$ . As a first step, we could consider forming  $\widehat{\Lambda}$  with the Bonferroni correction [48]:

$$\widehat{\Lambda}^{(\text{Bf})} = \{\lambda_j : p_j \leq \delta/|\Lambda|\}.$$

It is well known that this satisfies the requirements of Theorem 1, leading to valid risk control, which we state below for completeness.

**Proposition 3** (Bonferroni controls FWER). *Let  $\mathcal{A}^{(\text{Bf})}(p_1, \dots, p_N)$  be the function returning the set  $\widehat{\Lambda}^{(\text{Bf})}$  from the preceding display. Then,  $\mathcal{A}^{(\text{Bf})}$  is an FWER-controlling algorithm.*

Forming  $\widehat{\Lambda}^{(\text{Bf})}$  is very simple, which makes it attractive. There is a small improvement of this procedure due to Holm [45] that we use in our experiments, although the difference between the two versions is small when  $N$  is moderately large as in our examples. The Bonferroni method works reasonably well in our experiments, although for large  $|\Lambda|$  the multiplicity correction does degrade performance. We will report on this method in experiments in the sequel.

### 2.3.3 Fixed sequence testing

We next discuss a refinement of Bonferroni for our specific setting: *fixed sequence testing* [44, 49]. Notice that our multiple testing problem has several pieces of structure. First, our tests are parameterized by  $\lambda \in \Lambda$ , and the p-values will be highly dependent for nearby  $\lambda$  so that they often vary smoothly with  $\lambda$ . Second, we expect the non-nulls to appear in clusters with similar values of  $\lambda$ . Lastly, our goal is to reject based on a value of  $\lambda$  that guarantees the error control with reasonably good performance. Unlike in the settings motivating most of the multiple testing literature, we do not necessarily care about rejecting as many as possible.

In order to leverage this structure to avoid a severe multiplicity correction (and hence make more rejections), we instead carry out our tests in sequence. In particular, for a predefined ordering of the hypotheses, we walk through the sequence from beginning to end until we fail to reject for the first time. Then, we stop and return all hypothesis rejected so far. This procedure controls the type-1 error [44, 49]. More generally, we can initialize this procedure at several different points of the ordering, provided we adjust the significance level accordingly. This general procedure, stated formally in Algorithm 1, is also guaranteed to control the type-1 error, as recorded in Proposition 4.

In this algorithm, the user should choose an ordering of the values of  $\lambda$  roughly proceeding from the most likely to least likely to be rejected, possibly based on the training set but not based on the calibration data. Any ordering leads to valid error control, but some orderings will lead to better power than others. When  $\Lambda \subset \mathbb{R}$  we can often take the natural ordering: for the FDR control example in the introduction, we would start by considering  $\lambda = 1$  (where the FDR is controlled by definition) and then proceed to consider progressively smaller values. When  $\Lambda$  is higher dimensional, we discuss a general approach in Appendix C that learns an ordering using data splitting and apply that on instance segmentation detailed in Section 7.

---

**Algorithm 1** Fixed sequence testing

---

```
1: Input: error level  $\delta \in (0, 1)$ , parameter grid  $\Lambda = \{\lambda_1, \dots, \lambda_N\}$ , p-values  $(p_1, \dots, p_N)$ , initializations  $\mathcal{J} \subset \{1, \dots, N\}$  (e.g., a coarse equi-spaced grid)
2:  $\widehat{\Lambda} \leftarrow \emptyset$ 
3: for  $j \in \mathcal{J}$  do
4:   if  $j \notin \widehat{\Lambda}$  then ▷ Avoid repeating values of  $j$ .
5:     while  $p_j \leq \delta/|\mathcal{J}|$  do
6:        $\widehat{\Lambda} \leftarrow \widehat{\Lambda} \cup \{\lambda_j\}$ 
7:        $j \leftarrow j + 1$ 
8: Return: rejection set  $\widehat{\Lambda}$ 
```

---

Note that  $\mathcal{J}$  should be a coarse grid within  $\Lambda$ , with 20-100 elements.

**Proposition 4** (Fixed sequence testing controls FWER). *Algorithm 1 is an FWER-controlling algorithm; i.e., it satisfies Definition 2.*

Like Bonferroni, fixed sequence testing is simple to implement. In our experiments, we find that it offers moderate power improvements over Bonferroni. In the experiments, when  $|\mathcal{J}| > 1$  we call this method *multi-start* fixed sequence testing, and when  $|\mathcal{J}| = 1$ , we simply call it fixed sequence testing.

**Remark 2.** *Fixed sequence testing with  $|\mathcal{J}| = 1$  is essentially the same procedure as the upper confidence bound procedure in [2], though the latter allows  $\Lambda$  to be a continuous space. However, the multiple testing correction allows it to work for non-monotone risk functions. Nonetheless, we expect the fixed-testing procedure to be most powerful for risks that are nearly monotone before hitting the target level, such as the FDR. In fact, this procedure is not (asymptotically) conservative if the p-values are not (asymptotically) conservative.*

**Proposition 5.** *Let  $j^*$  be the index of the first null in the sequence. Then, for Algorithm 1 with  $|\mathcal{J}| = 1$ ,*

$$\text{FWER} = \mathbb{P}(p_{j^*} \leq \delta).$$

*As a result, if the null p-values are (asymptotically) uniform, the FWER is (asymptotically)  $\delta$  as well.*

### 2.3.4 A general recipe for FWER control

Lastly, we introduce a general, more powerful framework for FWER control due to Bretz, Maurer, Brannath, and Posch [50], called the graphical approach. We will call it *sequential graphical testing* (SGT) herein to emphasize that it is for the “Test” part in the Learn then Test framework. The SGT approach encodes information about the space of hypotheses  $\Lambda$  via a directed graph, where the null hypotheses indexed by  $\lambda \in \Lambda$  are the nodes, and the edges determine the way the error budget percolates through the graph. The procedure then sequentially tests the hypotheses indexed by  $\lambda \in \Lambda$  at iteratively updated significance levels, while guaranteeing that the final  $\widehat{\Lambda}$  controls the FWER. The basic idea is simple: when a hypothesis is rejected, its error budget gets distributed among adjacent hypotheses in the graph, which allows them to be rejected more easily. The fixed sequence testing procedure (Algorithm 1) is a special case of SGT [50].

Formally, the SGT procedure is parameterized by a directed graph  $\mathcal{G}$  comprising a node set  $\Lambda$ , and edge weights  $g_{i,j} \in [0, 1]$  for each pair  $i, j \in \Lambda$  obeying  $g_{i,i} = 0$  and  $\sum_{j=1}^n g_{i,j} \leq 1$ . In addition, each node  $i$  is allocated an initial error budget  $\delta_i$  such that  $\sum_i \delta_i = \delta$ . From here, the algorithm tests each hypothesis  $i \in \Lambda$  at level  $\delta_i$  (i.e., checks if  $p_i \leq \delta_i$ ). If any  $i$  is rejected, the procedure reallocates the error budget from node  $i$  to the rest of the nodes according to the edge weights; see Algorithm 2 for details. That is, each time a hypothesis is rejected, the algorithm is permitted to take the error budget and spend it elsewhere (i.e., increase the level of the remaining tests) leading to more rejections than weighted Bonferroni that assigns  $\delta_i$  error budget to the hypothesis  $i$ . Indeed, one can see that weighted Bonferroni corresponds the case with initial budget  $\delta_i$  and edge weights of zero. Note that this can be uniformly improved by using any set of nonzero edge weights, leading to at least as many rejections for any input set of p-values. Thus, Bonferroni is clearly suboptimal, and in some structured cases it can be greatly improved.

This procedure, outlined in Algorithm 2, controls the family-wise error rate, as stated next for completeness.



---

**Algorithm 2** Sequential graphical testing [50]

---

- 1: **Input:** error level  $\delta \in (0, 1)$ , parameter grid  $\Lambda = \{\lambda_1, \dots, \lambda_N\}$ , p-values  $(p_1, \dots, p_N)$ , graph  $\mathcal{G}$ , initial error budget  $\delta_i$  such that  $\sum_i \delta_i = \delta$
- 2:  $\widehat{\Lambda} \leftarrow \emptyset$
- 3: **while**  $\exists i : p_i \leq \delta_i$  **do**
- 4:     Choose any  $i$  such that  $p_i \leq \delta_i$
- 5:      $\widehat{\Lambda} \leftarrow \widehat{\Lambda} \cup \{\lambda_i\}$  ▷ Reject hypothesis  $i$
- 6:     Update the error levels and the graph:

$$\delta_j \leftarrow \begin{cases} \delta_j + \delta_i g_{i,j} & \lambda_j \in \Lambda \setminus \widehat{\Lambda} \\ 0 & \text{otherwise} \end{cases}$$

$$g_{k,j} \leftarrow \begin{cases} \frac{g_{k,j} + g_{k,i} g_{i,j}}{1 - g_{k,i} g_{i,k}} & \lambda_k, \lambda_j \in \Lambda \setminus \widehat{\Lambda}, \quad k \neq j \\ 0 & \text{otherwise} \end{cases}$$

- 7: **Return:** rejection set  $\widehat{\Lambda}$
- 

**Proposition 6** (SGT controls FWER [50]). *Algorithm 2 is an FWER-controlling algorithm; i.e., it satisfies Definition 2.*

The choices of the graph  $\mathcal{G}$  and initial error budget  $\{\delta_i\}_{i \in \Lambda}$  are critical for the power of the procedure. We next outline some basic principles guiding the design of these two parameters; see [50] for further discussion. The general idea is that we want the error budget to be initially concentrated on a small number of hypotheses most likely to reject. If these promising hypotheses are indeed rejected, then the error budget should flow to the their adjacent hypotheses. In this way, we accrue the error budget for rejected hypotheses and give the other ones a higher chance to be rejected. We explicitly develop a good design of  $\mathcal{G}$  and the initial error budget for the case where  $\Lambda$  is a two-dimensional grid in Section 6; see Figure 6 therein.

This approach toward FWER control will typically be the most powerful, since it allows the user to leverage structural information about the relationships between the hypotheses. We recommend that this approach be employed whenever it is feasible.

## 2.4 Multiple risks

The previous sections focused on controlling one statistical error rate by using multiple testing to select risk-controlling predictions from a one-dimensional family. However, this outlook restricts the user to simple families of predictions—generally ones that grow whenever  $\lambda$  does. Furthermore, these sets satisfy only one error control property, while we might want them to satisfy many. In this section, we show that our techniques also apply to multiple risks and more complicated set constructions with multi-dimensional  $\lambda$ .

Formally, we only have to slightly modify the theory in Section 2.2. Again, we consider a family of set-valued predictors  $\mathcal{T}_\lambda$  with a discrete index set  $\Lambda$ , which may have multiple dimensions. However we now seek to control  $m$  risks  $R_1, \dots, R_m$  at levels  $\alpha_1, \dots, \alpha_m$ . We would like to control all risks simultaneously, so we define the null hypothesis corresponding to  $\lambda_j$  as

$$H_j : R_l(\lambda_j) > \alpha_l, \quad \text{for some } l \in 1, \dots, m.$$

To test this null hypothesis, we must examine the finer null hypotheses,

$$H_{j,l} : R_l(\lambda_j) > \alpha_l.$$

Specifically,  $H_j$  holds if and only if there exists a  $l \in 1, \dots, m$  such that  $H_{j,l}$  holds, which allows us to apply an FWER-controlling procedure to test  $H_j$ , as we now summarize.

**Proposition 7.** *Let  $p_{j,l}$  be a p-value for  $H_{j,l}$ , for each  $l = 1, \dots, m$ . Define  $p_j := \max_l p_{j,l}$ . Then, for all  $j$  such that  $H_j$  holds, we have*

$$\mathbb{P}(p_j \leq u) \leq u,$$

for all  $u \in [0, 1]$ .

Notice that now we have a conservatively valid p-value,  $p_j$ , for the null hypothesis  $H_j$ . Having calculated valid p-values for each  $\lambda_j$ , we can now directly use the techniques from the previous section to select a set  $\hat{\Lambda}$  that controls the FWER.

## 2.5 An alternative approach: uniform concentration

An alternative approach that may seem natural is to use a *uniform concentration bound* to control the risk. In short, the idea of uniform concentration is to upper bound the risk simultaneously for all  $\lambda$ —i.e., the upper bound lies above the true risk for *all*  $\lambda$  at once with high probability. With the bound in hand, one can produce  $\hat{\Lambda}$  by including all  $\lambda$  where the bound falls below  $\alpha$ , as we state next.

**Proposition 8** (Uniform bounds give risk control). *Let  $R^+$  be a  $1 - \delta$  uniform upper confidence bound such that*

$$\mathbb{P}(R(\mathcal{T}_\lambda) \leq R^+(\mathcal{T}_\lambda) \text{ for any } \lambda \in \Lambda) \geq 1 - \delta.$$

*For any desired risk level  $\alpha$ , consider any  $\hat{\lambda}$  such that  $R^+(\mathcal{T}_{\hat{\lambda}}) \leq \alpha$ . Then,*

$$\mathbb{P}\left(R(\mathcal{T}_{\hat{\lambda}}) \leq \alpha\right) \geq 1 - \delta.$$

We find that despite the elegance of uniform concentration, its practical performance is quite poor. (See Figure 3 and Figure 4.) This is not for lack of trying; we develop a novel and carefully optimized concentration bound in Appendix E. This bound consistently and significantly underperforms even the most naive multiple testing procedure, the Bonferroni correction. Nonetheless, we include it as a yardstick for our multiple testing procedures in our forthcoming experiments, and run a simulation in Appendix A.1 to compare it against multiple testing procedures in a controllable testbed.

To be clear, it should be expected that uniform concentration will fare worse than multiple testing because it solves a harder problem. First, uniform bounds operate on a continuous  $\Lambda$ , whereas testing procedures operate on a discrete grid. Second, generic uniform bounds do not take the problem structure into account, while multiple testing procedures can build it into the sequence of hypotheses. This second point matters even more in high dimensions, as the metric entropy scales poorly while testing procedures can essentially cut down the problem dimension by leveraging structure (e.g., via SGT). Thus, multiple testing had the upper hand from the outset; the experiments confirm this.

## 3 Example: FDR Control for Multi-Label Classification

Next, we consider four different computer vision tasks and a tabular medical expenditure prediction task, using our framework to give procedures with finite-sample statistical guarantees. Each of these examples highlights a new and useful form of error control.

We first consider multi-label classification, where each input  $X$  (in our case, an image) may have multiple corresponding correct labels; i.e., the response  $Y$  is a subset of  $\{1, \dots, K\}$ . Here, we seek to return predictions that control the FDR at level  $\alpha$ ,

$$R(\mathcal{T}) = 1 - \mathbb{E} \left[ \frac{|Y \cap \mathcal{T}(X)|}{|\mathcal{T}(X)|} \right],$$

where  $Y$  is the ground truth label set and  $\mathcal{T}$  is the prediction set function. That is, we want to predict sets that contain no more than  $\alpha$  proportion of false labels on average. In this case, we take our prediction to be the following set of classes:

$$\mathcal{T}_\lambda(x) = \{z \in \{1, \dots, K\} : \hat{f}_z(x) > \lambda\}, \quad (3)$$

where  $\hat{f} : \mathcal{X} \rightarrow [0, 1]^K$  is a base classifier that outputs estimated probabilities that each class is present in the image. The intuition behind this set-valued prediction is simple: all sufficiently probable classes (as judged by our base model) are included in  $\mathcal{T}_\lambda(X)$ . The Learn then Test framework will then allow us to set  $\lambda$  in such a way that guarantees FDR control.

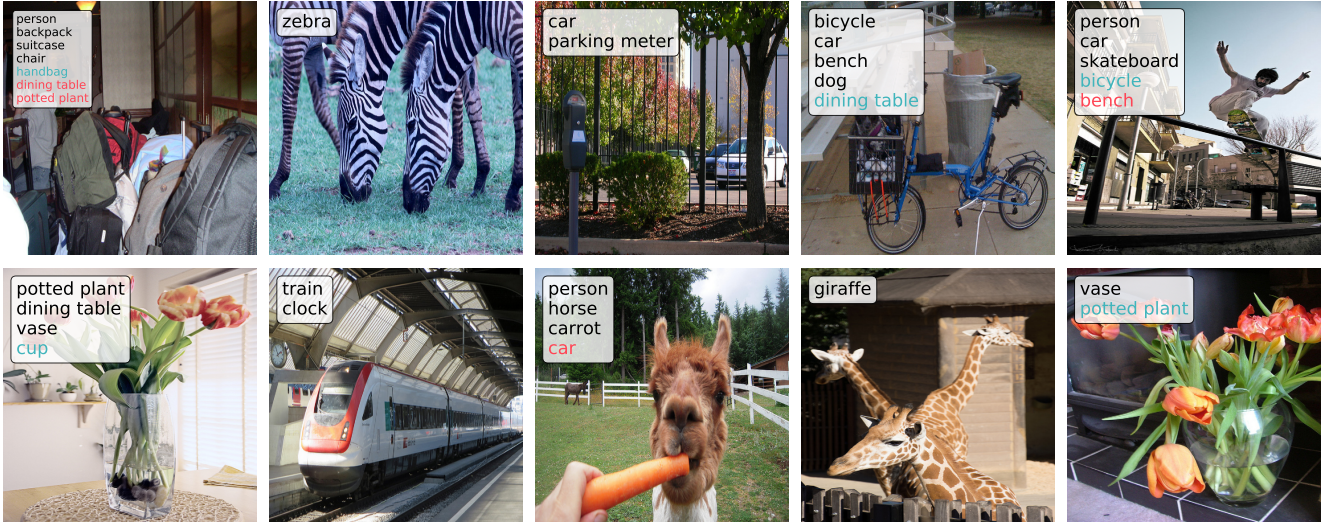


Figure 2: **Multi-label prediction set examples on MS COCO.** Black classes are correctly identified (true positives), blue ones are spurious (false positives), and red ones are missed (false negatives). The sets are produced with parameters  $\alpha = 0.2$  and  $\delta = 0.1$  using the fixed sequence testing procedure.

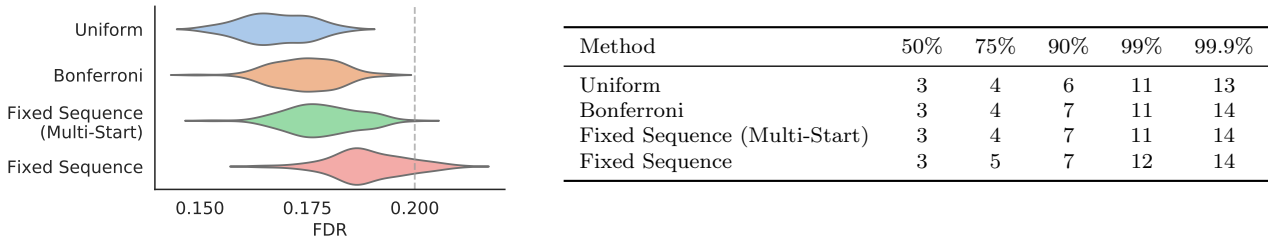


Figure 3: **Numerical results of our multi-label classification procedure.** The risk is plotted as a violin plot over 100 random splits of MS COCO, with parameters  $\alpha = 0.2$ , shown as the grey dotted line, and  $\delta = 0.1$ . The set sizes are shown as a table on the right-hand side, which reports quantiles of set sizes for each method (larger is better). For details see Section 3.

We use the Microsoft Common Objects in Context (MS COCO) computer vision dataset [51] to evaluate the proposed procedure, as well as a baseline based on uniform concentration (see Section 2.5). MS COCO is an 80-class dataset where each image may have several labels because it contains several objects. Following the development of [2], we use TResNet as the base model [52], and threshold the vector of softmax probabilities as in (3). To set the threshold, we choose  $\hat{\lambda}$  as in Algorithm 1, using 4,000 calibration points, and then we evaluate the FDR on an additional test set of 1,000 points. Seeking to control the FDR at level  $\alpha = 0.2$  with tolerance parameter  $\delta = 0.1$ , we plot the empirically-observed FDR and set size in Figure 3. In particular, we report on three versions of Learn then Test that each use a different multiple testing subroutines as well as a uniform concentration baseline. We also present the results on ten random images in Figure 2. See Figure 12 in Appendix A for the results with  $\alpha = 0.5$ . All methods control the risk, with fixed sequence testing being essentially tight. Note the poor performance of the uniform concentration approach in this example.

## 4 Example: pFDR Control for Selective Classification

We now consider the selective classification problem, sometimes called classification with a reject option or classification with abstention. In detail, we consider a single-class classification setting where  $\mathcal{Y} = \{1, \dots, K\}$  for some  $K$ , and  $\hat{f} : \mathcal{X} \rightarrow \Delta^K$  (the simplex on  $K$  entries, since there can only be one true class). For each test point, the classifier can either return a prediction set or abstain from making a prediction, which we encode as returning the

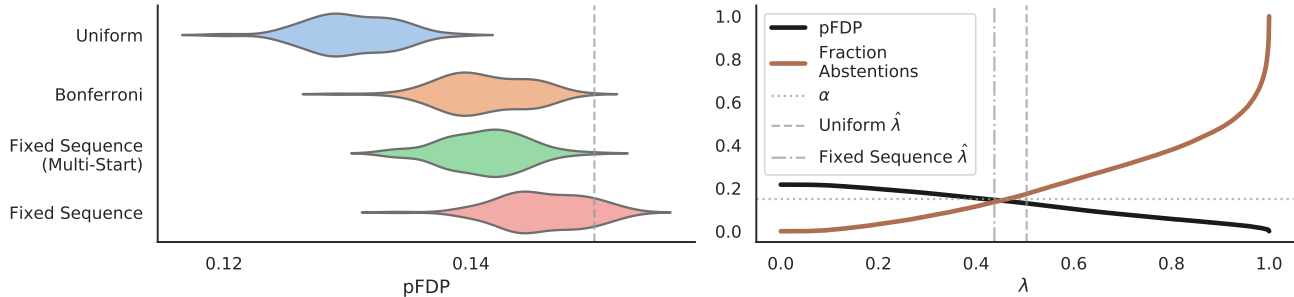


Figure 4: **Numerical results of our pFDR control procedure on Imagenet.** The left-hand violins show the density of the pFDP over 100 independent random splits of the calibration and validation set, at levels  $\alpha = 0.15$  and  $\delta = 0.1$ . The right hand side shows the empirical pFDP and fraction of abstentions using the ResNet model when sweeping across values of  $\lambda$ . We also plot the median values of  $\hat{\lambda}$  chosen by the uniform bound and fixed sequence procedure as vertical gray lines; farther left means fewer abstentions.

empty set  $\emptyset$ . As a notion of error in this setting, we consider the positive FDR (pFDR) [53]:

$$\text{pFDR}(\mathcal{T}) := \mathbb{E} \left[ \frac{|\mathcal{T}(X) \setminus Y|}{|\mathcal{T}(X)|} \mid |\mathcal{T}(X)| > 0 \right]. \quad (4)$$

In order to control the pFDR, we show that it can be massaged to take the form of an unconditional average of bounded random variables, so that we can apply the techniques from Section 2. To this end, notice that  $\text{pFDR}(\mathcal{T}_\lambda)$  can be written as  $v(\lambda)/r(\lambda)$ , where  $v(\lambda) = \mathbb{E} [|\mathcal{T}_\lambda(X) \setminus Y| / |\mathcal{T}_\lambda(X)| \cdot \mathbb{I}\{|\mathcal{T}_\lambda(X)| > 0\}]$  and  $r(\lambda) = \mathbb{E} [\mathbb{I}\{|\mathcal{T}_\lambda(X)| > 0\}]$ . We then have

$$\frac{v(\lambda)}{r(\lambda)} \leq \alpha \iff v(\lambda) - \alpha r(\lambda) + \alpha \leq \alpha \iff \mathbb{E} \left[ \frac{|\mathcal{T}_\lambda(X) \setminus Y|}{|\mathcal{T}_\lambda(X)|} \cdot \mathbb{I}\{|\mathcal{T}_\lambda(X)| > 0\} + \alpha \mathbb{I}\{|\mathcal{T}_\lambda(X)| = 0\} \right] \leq \alpha.$$

Thus, we have reformulated our problem as a risk control problem without the conditioning originally present in (4). Moreover, we arranged terms so that the integrand falls in  $[0, 1]$ , so we can apply our techniques from Section 2.3 to obtain p-values and thus valid hypothesis tests.

In the special case where  $|\mathcal{T}(X)| \in \{0, 1\}$ —that is, we either make a point prediction or abstain—the pFDR in (4) is simply the classification error conditional on making a prediction. In our empirical example below, we will work in this setting. In particular, we instantiate the following family of set-valued predictions:

$$\mathcal{T}_\lambda(x) = \begin{cases} \{y_{\max}\} & \hat{f}_{y_{\max}}(x) > \lambda \\ \emptyset & \text{else,} \end{cases}$$

where  $y_{\max} = \arg \max_y \hat{f}_y(x)$ . These predictions are either the singleton set containing the most likely class or the null set, depending on the confidence of the model.

We now demonstrate this pFDP control on the Imagenet [54] computer vision dataset using a ResNet-152 [55] model as the base predictive model. In particular,  $\hat{f}$  maps inputs to the probability simplex on Imagenet’s 1000 classes. We report numerical results in Figure 4, using 30K calibration points and 20K validation points with  $\alpha = 0.15$  and  $\delta = 0.1$ . As before, we find that the finite-sample valid multiple testing approaches are all correct, and that the best testing method is essentially tight. By contrast, the approach based on uniform concentration is very conservative.

## 5 Example: Selective Regression with MSE Control

In the next example, we perform a regression task, but abstain from predicting when the model is least confident; this task is sometimes called *selective regression*. Learn then Test allows us to choose the abstention cutoff to control the mean-squared error (MSE). We will demonstrate results on the Medical Expenditure Panel Survey (MEPS)

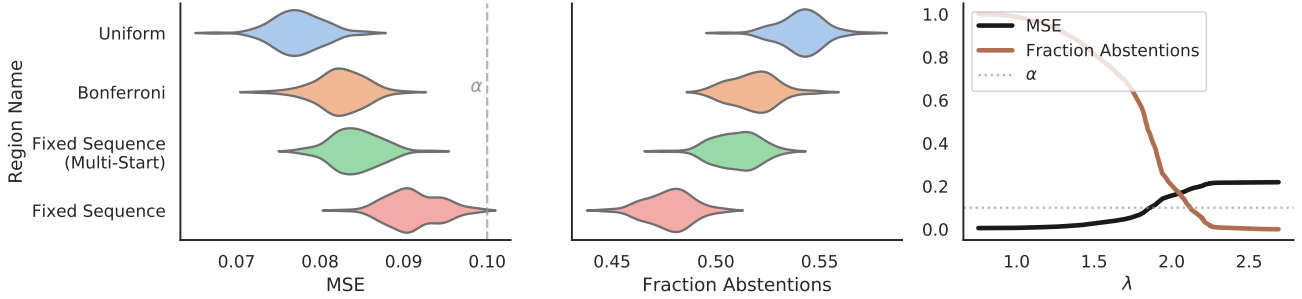


Figure 5: **Numerical results of selective regression on the MEPS dataset.** On the left, the MSE is plotted as a violin plot over 100 random splits of the MEPS data, with parameters  $\alpha = 0.1$ , shown as the grey dotted line, and  $\delta = 0.1$ . In the middle, the fraction of abstentions is plotted similarly. On the right-hand side, the set sizes are shown as a table reporting quantiles of set sizes for each method (larger is better). For details see Section 5.

dataset [56], wherein we try to predict medical expenditures of American households from a table of thousands of covariates, such as whether there is a diabetes or cancer diagnosis, the income level of the household, and so on.

Formally, each input  $X$  is a vector in  $\mathcal{X} = \mathbb{R}^d$ , where each of the  $d$  dimensions signifies a covariate (a column index in the tabular dataset). The response  $Y \in \mathbb{R}_+$  is in this case a medical expenditure. The goal is to predict when we can be assured a low MSE and to abstain otherwise; that is, we control the conditional risk

$$R(\mathcal{T}) = \mathbb{P}\left(\left(\mathcal{T}(X) - Y\right)^2 \mid \mathcal{T}(X) \neq \emptyset\right),$$

where  $\mathcal{T}$  takes values in  $\mathbb{R}$  or the value  $\emptyset$  signifying abstention. While this is a conditional risk, risk control can be expressed as marginal average, as done previously in Section 4. Thus, we can obtain finite-sample valid p-values for this setting.

To create our final selective regression procedure, we will use two base models, in our case gradient boosting trees [57]; the first model  $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}_+$  estimates the conditional mean, and the second model  $\hat{r} : \mathcal{X} \rightarrow \mathbb{R}_+$  estimates the magnitude of the residual of the first model. These models are trained in sequence—first  $\hat{f}$ , then  $\hat{r}$ . This is similar to the approach of [58]. We combine the two models, abstaining when the estimated residual is high,

$$\mathcal{T}_\lambda(x) = \begin{cases} \hat{f}(x) & \hat{r}(x) \leq \lambda \\ \emptyset & \text{else} \end{cases}.$$

As  $\lambda$  shrinks, we get more abstentions, and if  $\hat{r}$  is any good, we also get a lower MSE because the abstentions happen on points with large errors.

We demonstrate this MSE control technique on the Medical Expenditure Prediction Survey (MEPS) data from 2019. The MEPS dataset consists of covariates about households paired with their yearly medical expenditures; we seek to predict the latter from the former. We followed the procedure of [18] to preprocess and clean the data; we also process the expenditures to take values between 0 and 1 by dividing by and clipping to the maximum value in the training dataset in order to produce p-values with (2). For  $\hat{f}$  and  $\hat{r}$  we use gradient boosting trees [57, 59] implemented with SciPy [60]. We set  $\alpha = \delta = 0.1$ , then run the LTT algorithm using three FWER-controlling algorithms along with a uniform concentration baseline (see Section 2.5). The numerical results in Figure 5 show control of the conditional MSE at the correct level, and also that the tighter control offered by fixed sequence testing leads to significantly fewer abstentions (a little more than 5% on the absolute scale).

## 6 Example: Prediction Sets with OOD Detection

Next, we seek to make predictions that either declare an input to be out-of-distribution (OOD) or return a prediction set with a coverage guarantee. In this setting, we want to detect as many out-of-distribution inputs as possible while only falsely rejecting some pre-specified fraction (e.g., 1%) of in-distribution inputs. Furthermore, within the subset of inputs deemed in-distribution, we ask for prediction sets with coverage  $1 - \alpha_2$  (e.g., 99%). We formalize

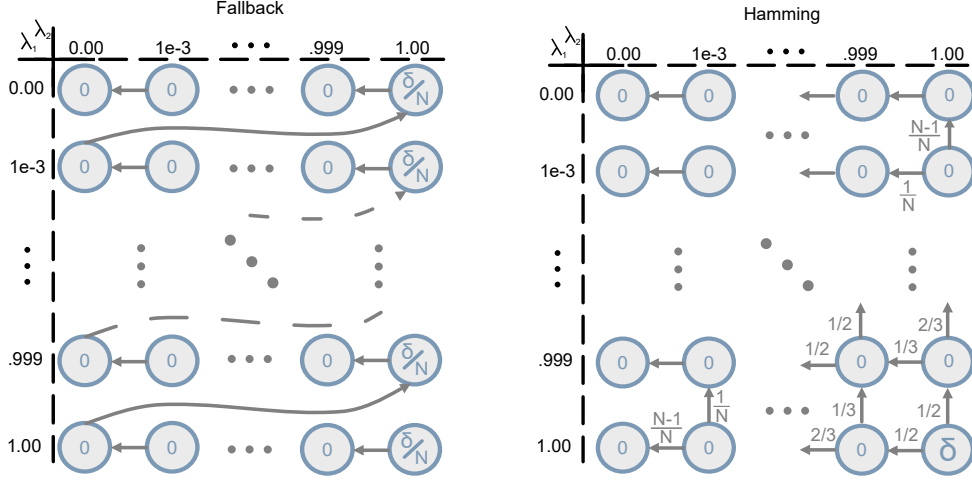


Figure 6: **The SGT graphs** for the two multiple testing procedures used in Section 6. Each node corresponds to the null hypothesis  $H_{i,j}$  as described in (5). The initial allocations of the error budget are shown within the nodes, and the edges define the transition matrix of the graph. Edges without a label have weight 1. The left graph describes a form of SGT called a *fallback procedure*, wherein the hypotheses are chained together, with the initial error budget distributed throughout the chain. The right graph is the *Hamming* graph, which allocates the entire error budget to the bottom right node and then propagates the error budget outwards; see Appendix D.

this goal by using a vector-valued risk function with two coordinates:  $R(\mathcal{T}) = (R_1(\mathcal{T}), R_2(\mathcal{T})) \in [0, 1]^2$ . We take

$$R_1(\mathcal{T}) = \mathbb{P}(|\mathcal{T}(X)| = 0) \quad \text{and} \quad R_2(\mathcal{T}) = \mathbb{P}(Y \notin \mathcal{T}(X) \mid |\mathcal{T}(X)| > 0).$$

The probability in the above is over the distribution of in-distribution data: the distribution of the calibration points  $(X_1, Y_1), \dots, (X_n, Y_n)$ . To find a function  $\mathcal{T}$  such that  $R_1(\mathcal{T}) \leq \alpha_1$  and  $R_2(\mathcal{T}) \leq \alpha_2$  holds with probability at least  $1 - \delta$ , we will apply Learn then Test, using the extension described in Section 2.4 to handle vector-valued risk functions.

To deal with these two risk functions simultaneously, we will consider a family of models  $\mathcal{T}_\lambda$  parameterized by a two-dimensional vector  $\lambda = (\lambda_1, \lambda_2) \in [0, 1]^2$ . The first parameter  $\lambda_1$  is used to control the OOD detection. In particular, we use an OOD score,  $\text{OOD}(x)$ , designed to be large when the input is predicted as out-of-distribution, and we use  $\lambda_1$  as the threshold for declaring a point to be out-of-distribution. Next, we use a function  $s(x, y)$ , that controls which classes are in the prediction set, and we will threshold this function at level  $\lambda_2$  to create prediction sets. Concretely, the set construction procedure is:

$$\mathcal{T}_\lambda(X) = \begin{cases} \emptyset, & \text{OOD}(X) \geq \lambda_1, \\ \{y : s(X, y) \leq \lambda_2\} \cup \arg \min_y s(X, y), & \text{OOD}(X) < \lambda_1. \end{cases}$$

These sets have a simple interpretation: when  $\mathcal{T}_\lambda(X) = \emptyset$ , the example is deemed OOD. Otherwise, we return a set of labels that are guaranteed to include the true label with probability  $1 - \alpha_2$ . We will soon instantiate specific choices for the functions  $\text{OOD}(x)$  and  $s(x, y)$  based on standard pre-trained deep learning models. With this notation in place, the output of our calibration procedure will be all pairs of  $(\lambda_1, \lambda_2)$  that can control both the type-1 error of OOD detection, and the coverage conditional on prediction simultaneously.

Next, we specialize a multiple-testing method that takes advantage of the structure of our risk function. Notice that the risks are coordinate-wise monotonic, meaning

$$\text{if } \lambda_2 \text{ is fixed, } \lambda_1^{(1)} \leq \lambda_1^{(2)} \implies R_1((\lambda_1^{(1)}, \lambda_2)) \leq R_1((\lambda_1^{(2)}, \lambda_2)),$$

and vice-versa when  $\lambda_2$  varies with  $\lambda_1$  fixed. Furthermore, for a fixed  $\lambda_1$ , we are most interested in the smallest  $\lambda_2$ , and vice versa; this essentially defines a Pareto frontier between  $R_1$  and  $R_2$ . To efficiently identify values of  $(\lambda_1, \lambda_2)$  that control the risk, we will use SGT, as described in 2.3, implemented with two different choices of graph (described below) to try to maximize power. We will also compare to a version of fixed sequence testing adapted

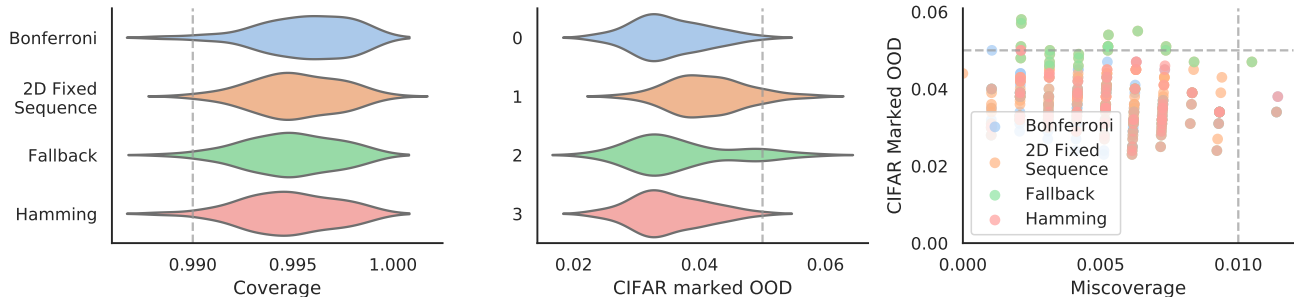


Figure 7: **Numerical performance of methods for simultaneous OOD type-1 error and coverage control on CIFAR-10.** We show violin plots describing the numerical performance of our testing methods. The violins quantify coverage, OOD type-1 error, and the power of the OOD procedure against Imagenet images, which are from a different dataset. The randomness is over 1000 different splits of the calibration and validation data. In these plots, we set the risk levels as  $\alpha_1 = 0.05$ ,  $\alpha_2 = 0.01$ , and  $\delta = 0.1$ . The grey dotted lines show the coordinates of  $\alpha$  on their respective plots. See Section 6 for an explanation of these results.

for 2-D problems and the Bonferroni correction. In all cases, we discretize  $\Lambda = \{0, \frac{1}{1000}, \frac{2}{1000}, \dots, 1\}^2$ , yielding the null hypotheses

$$H_{(i,j)} : R_1(\Lambda_{(i,j)}) > \alpha_1 \text{ or } R_2(\Lambda_{(i,j)}) > \alpha_2, \text{ for each } 0 \leq i, j \leq 1000. \quad (5)$$

We now turn to the details of the three multiple testing methods. First, we consider the SGT with graph shown in the left panel of Figure 6, which is known as the *fallback procedure* [61, 62]. Here, we start with error budget spread equally across levels of  $\lambda_1$ , and then propagate the error budget towards the aforementioned Pareto frontier, starting from the safe points where  $\lambda_2$  is large. This is nearly the same as the fixed sequence test that splits the error budget between each  $\lambda_1$  and then running a fixed-sequence test on each to find the smallest  $\lambda_2$  for each  $\lambda_1$ , but it is more powerful due to the edges connecting adjacent rows.

Next, we consider SGT with the *Hamming* graph in the right panel of Figure 6. Here, the initial error budget is all allocated to the bottom-right node, and then it cascades outward. The weights are chosen to yield a balanced procedure, in the sense that the maximum possible error budget a node can be allocated (which happens if there are rejections at all ancestor nodes) is  $\delta$  divided by the Hamming distance to the bottom right. See Appendix D for details.

As a final attempt on the multiple testing design, we designed a cascaded 2D fixed-sequence test. First, a budget of  $\delta/2$  gets allocated to picking  $\lambda_1$  via fixed-sequence testing. After finding the smallest  $\hat{\lambda}_1$  that controls  $R_1$ , we do a second fixed-sequence test to find the smallest  $\lambda_2$  such that the pair  $(\hat{\lambda}_1, \lambda_2)$  controls both risks.

We evaluate these methods on the CIFAR-10 dataset [63], which has 10 classes:  $\mathcal{Y} = \{1, 2, \dots, 10\}$ . We take a DenseNet classifier  $\hat{f}$  as our base model. As our OOD score, we use a method called ODIN [64]; see Appendix A.3 for details. As our classification score function  $s(x, y)$ , we use the conformal score from the Adaptive Prediction Sets procedure introduced in [65] and [20]—see [11] for a description of the version we use here. Formally, the score function is

$$s(x, y) = \sum_{j=1}^k \hat{f}_{\pi_j}(x), \text{ where } y = \pi_k, \quad (6)$$

and  $\pi$  is the permutation of  $\{1, \dots, K\}$  that sorts  $\hat{f}(X)$  from most to least likely.

The numerical performance of our method is reported in Figure 7. Here, we used  $\Lambda = \{0, \frac{1}{1000}, \frac{2}{1000}, \dots, 1\}^2$  with 8000 calibration points and 2000 validation points. As expected, we see that we control both risks at the desired level. To evaluate the power of the OOD detection substep, we also tested it on 10,000 downsampled images from Imagenet; it correctly identifies  $> 99\%$  of Imagenet images as OOD. Thus, our strategy leads to reasonable results which are not conservative in coverage or OOD Type-1 error and easily distinguish against Imagenet. Indeed, the SGT procedure violates the desired risks 10% of the time when we set  $\delta = 10\%$ ; the procedure hits exactly the target level.

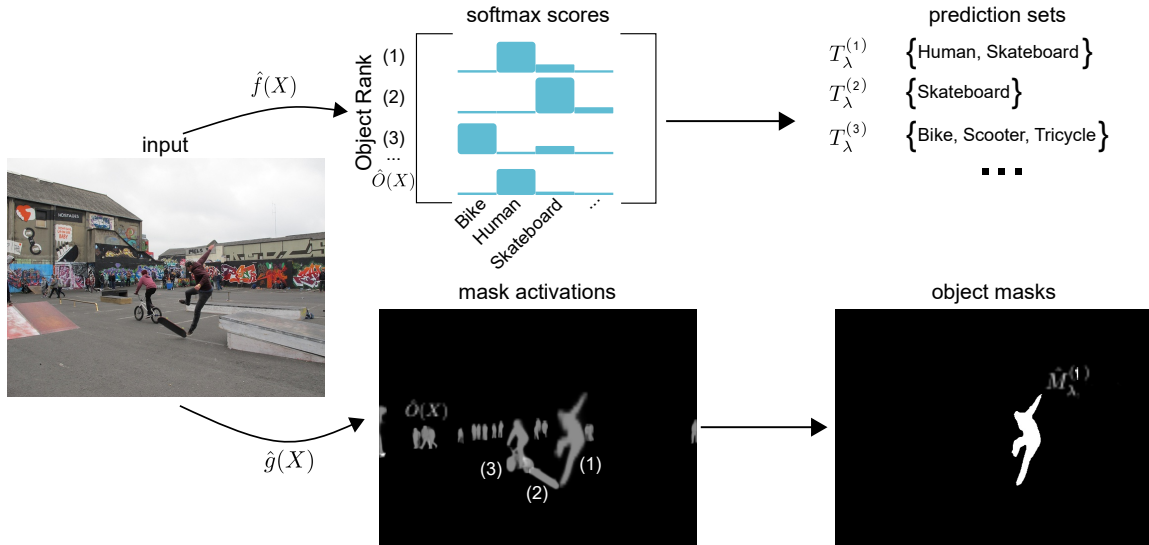


Figure 8: **Our detection pipeline** annotated with the formal mathematical notation to aid the reader.

## 7 Example: Instance Segmentation with mIOU, Coverage, and Recall Guarantees

We finish with our flagship example: object detection. We will focus on a variant of object detection called *instance segmentation*, in which we are given an image and asked to (1) identify all distinct objects within the image, (2) segment them from their background, and (3) classify them. See Figure 9 for several examples of our procedure. These three goals can be formally encoded by evaluating a detector’s recall, the *Intersection-over-Union* (IOU) of the segmentation masks with the ground truth mask, and the misclassification rate, respectively. Traditionally, these measures are combined into a single metric called *average precision* to heuristically evaluate the performance of a detector [66]. Although the average precision can also be handled by our methods, here we will take the stronger stance of *controlling* all three error rates at the same time.

In more detail, we will use **recall**, **IOU**, and **coverage** as the error rates corresponding to the three subtasks of detection. (These error rates will be formally defined soon.) We set up our experiments such that the detector has three final tuning parameters,  $\lambda_1, \lambda_2$  and  $\lambda_3$ , that each controls the detector’s performance on one of the three tasks. In particular,  $\lambda_1$  tunes the number of objects that are selected,  $\lambda_2$  tunes the size of the bounding regions, and  $\lambda_3$  tunes the certainty level for classification. The setup can be summarized as follows:

Goal	Quantity to Control	Parameter
(1) Locate distinct objects	<b>recall</b>	$\lambda_1$
(2) Find the precise set of pixels for each object	<b>IOU</b>	$\lambda_2$
(3) Assign the right class to each object	<b>coverage</b>	$\lambda_3$

In truth, this is a simplification, in that the parameters  $\lambda_1, \lambda_2$  and  $\lambda_3$  are not entirely disentangled: changing any one parameter can change all three error rates. Nonetheless,  $\lambda_1$  primarily corresponds to the **recall** error rate, and so on. This will be made precise in the next section. Lastly, before launching into the formal details, we note that state-of-the-art object detectors like **detectron2** [67] have many substeps, and for the sake of brevity we will only explain the relevant ones.

### 7.1 Formal specification of object detection

Now we begin a formal treatment of the instance segmentation problem. Our inputs are images  $X_i \in \mathbb{R}^{H \times W \times D}$ ,  $i = 1, \dots, n$  where  $H$ ,  $W$ , and  $D$  are the height and width and channel depth respectively (in practice each image is a different size, but we ignore this for notational convenience). Along with each input, we receive a set of tuples containing a mask and a class for each of the  $O(X_i)$  number of objects in image  $X_i$ . The response



$M_i^{(j)} \in \{0, 1\}^{H \times W}$ ,  $j = 1, \dots, O(X_i)$  represents a binary segmentation mask for the  $j$ th object in image  $i$  and the response  $C_i^{(j)} \in \{1, \dots, K\}$  represents the class of that object. Thus, for a given  $X_i$ , the response  $Y_i$  is the sequence of masks and classes

$$Y_i = \left\{ \left( M_i^{(j)}, C_i^{(j)} \right) \right\}_{j=1}^{O(X_i)}.$$

For our purposes, an object detector comprises three functions:  $\hat{O}(X_i)$ , which gives the number of predicted objects,  $\hat{f}(X_i) \in [0, 1]^{\hat{O}(X_i) \times K}$  (with elements  $\hat{f}_k^{(j)}(X_i)$  where  $j \in \{1, \dots, \hat{O}(X_i)\}$  and  $k \in \{1, \dots, K\}$ ), a set of estimated class probabilities for each of the predicted objects, and  $\hat{g}(X_i) \in [0, 1]^{\hat{O}(X_i) \times H \times W}$  (with elements  $\hat{g}_{h,w}^{(j)}$  where  $j \in \{1, \dots, \hat{O}(X_i)\}$ ,  $h \in \{1, \dots, H\}$ , and  $w \in \{1, \dots, W\}$ ), representing the probability that a pixel comes from each predicted object. In our eventual pipeline, the job of  $\hat{f}$  will be to classify each predicted object, and the job of  $\hat{g}$  will be to form binary segmentation masks for each. It will soon become relevant that the number of predicted objects is not equal to the number of ground truth objects, i.e.,  $\hat{O}(X_i) \neq O(X_i)$ . Now, we must tackle the task of turning the raw outputs  $\hat{f}(X)$  and  $\hat{g}(X)$  into binary masks and classes, while filtering out those predictions deemed unreliable.

First, we select only the most confident detections for eventual display to the user. In words, we will only output objects with a sufficiently high softmax score, since lower scores suggest that the object is more likely to be spurious. We will call the indexes of those objects

$$\tilde{J}_\lambda(X) = \{j : \max_k \hat{f}_k^{(j)}(X) \geq 1 - \lambda_1\} \subseteq \{1, \dots, \hat{O}(X)\}.$$

We will use  $\tilde{J}_\lambda(X)$  to ignore objects with top scores lower than the threshold  $1 - \lambda_1$  in later risk computations.

Second, we explain the construction of binary masks. We set the  $(h, w)$  coordinate of the estimated masks for the  $j$ th object as

$$\hat{M}_{\lambda, (h, w)}^{(j)}(X) = \mathbb{1} \left\{ \hat{g}_{h, w}^{(j)}(X) \geq 1 - \lambda_2, j \in \tilde{J}_\lambda(X) \right\}.$$

Let us reflect on the construction of the mask. If the pixel truly comes from the object,  $\hat{g}_{h, w}^{(j)}(X)$  should be large, which motivates us to threshold these values at  $1 - \lambda_2$ . Also note that if the model is not sufficiently confident about the object's class, then  $j \notin \tilde{J}_\lambda(X)$ , causing the procedure to give up and output a mask of all zeros.

Third, we discuss the formation of the prediction sets for the class of the object. We set

$$T_\lambda^{(j)}(X) = \begin{cases} \emptyset & j \notin \tilde{J}_\lambda(X) \\ \{k : s^{(j)}(X, k) \leq \lambda_3\} \cup \left\{ \arg \max_k \hat{f}_k^{(j)}(X) \right\} & j \in \tilde{J}_\lambda(X), \end{cases}$$

where  $s^{(j)}(X, k)$  is the score from (6) of class  $k$  on predicted object  $j$  from image  $X$ , which is lower for more likely classes (see (6)). Holding  $\lambda_1$  fixed, the prediction set grows when  $\lambda_3$  grows. Also, as in the example of the mask construction, if the detector is not sufficiently confident about the top class, the procedure gives up and outputs the null set.

Having described both ingredients of our final output, we can now bring them together. Given an input image  $X$ , our final prediction is a sequence of  $\hat{O}(X)$  masks and prediction sets for the object class,

$$\hat{Y}_\lambda = \left\{ \left( \hat{M}_\lambda^{(j)}(X), \hat{T}_\lambda^{(j)}(X) \right) \right\}_{j=1}^{\hat{O}(X)}.$$

Now we will formally define the risk functions. Before doing so, we begin by defining the IOU for two binary masks:

$$\text{IOU}(M, \hat{M}) = \frac{\# \left\{ (h, w) : M_{(h, w)} = 1 \text{ and } \hat{M}_{(h, w)} = 1 \right\}}{\# \left\{ (h, w) : M_{(h, w)} = 1 \text{ or } \hat{M}_{(h, w)} = 1 \right\}}.$$

The IOU is an FDR-like quantity for segmentation masks. Importantly, the IOU function gets used to determine the correspondence between the predicted objects and the ground truth objects—recall that we do not know which



Figure 9: **Instance segmentation examples on MS COCO.** The input images are on the left, and the output instance segmentations are on the right. The segmentation masks are shown in random colors over their respective objects, and the prediction sets are included as color-coded lists of classes adjacent to the bounding box for each object. We produced these images by running the fixed-testing procedure with  $\alpha_1 = 0.5$ ,  $\alpha_2 = 0.5$ ,  $\alpha_3 = 0.25$ , and  $\delta = 0.1$ .

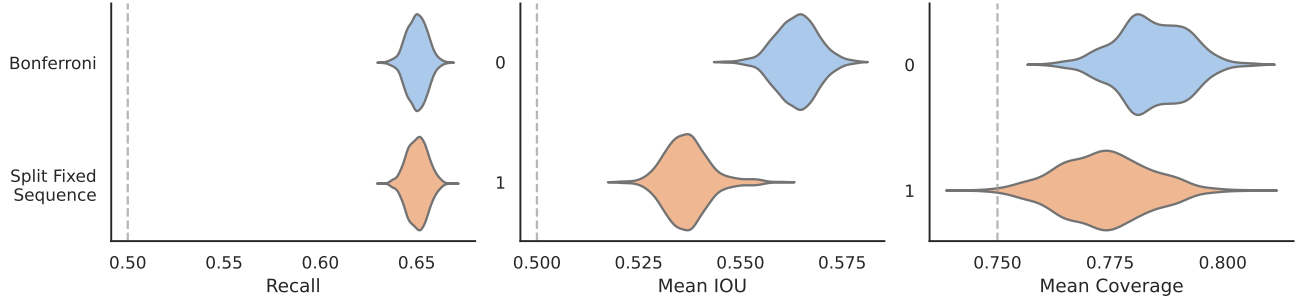


Figure 10: **Numerical results for our instance segmentation algorithm.** In order from left to right, we plot the mean **coverage**, mean **IOU**, and **recall** over 1000 random splits of the MS-COCO dataset. The grey dotted lines represent the coordinates of  $\alpha$ . We chose  $\alpha_1 = 0.5$ ,  $\alpha_2 = 0.5$ ,  $\alpha_3 = 0.25$ , and  $\delta = 0.1$ . We define the split fixed sequence testing procedure in Appendix C. Here, we have chosen  $\hat{\lambda}$  in such a way as to make coverage as tight as possible for both procedures. This leads to the conservativeness of **recall** and **IOU**; see the discussion at the end of Section 7.

predicted objects, if any, correspond to each ground truth object. To match the predictions with the ground truth, we set  $\tilde{\pi}$  to be the permutation maximizing

$$\min(O(X), \hat{O}(X)) \sum_{j=1}^{\min(O(X), \hat{O}(X))} \text{IOU}(M^{(j)}, \hat{M}_{\lambda}^{\tilde{\pi}(j)}(X)) \mathbb{1}\{j \in \tilde{J}_{\lambda}(X)\}.$$

From now on, assume without loss of generality that  $\tilde{\pi}(j) = j$ . Finally, we define our three risks, beginning with the negative of the **recall**,

$$R_1 = \mathbb{E} \left[ 1 - \frac{1}{O(X)} \sum_{j=1}^{\min(O(X), \hat{O}(X))} \mathbb{1}\left\{C^{(j)} = \arg \max_k \hat{f}_k^{(j)}(X), j \in \tilde{J}_{\lambda}(X)\right\} \right],$$

continuing with the negative of the average **IOU**,

$$R_2 = \mathbb{E} \left[ 1 - \frac{1}{|\tilde{J}_{\lambda}(X)|} \sum_{j=1}^{\min(O(X), \hat{O}(X))} \text{IOU}(M^{(j)}, \hat{M}_{\lambda}^{(j)}(X)) \mathbb{1}\{j \in \tilde{J}_{\lambda}(X)\} \right],$$

and ending with the negative of the image-wise average **coverage**,

$$R_3 = \mathbb{E} \left[ 1 - \frac{1}{|\tilde{J}_{\lambda}(X)|} \sum_{j=1}^{\min(O(X), \hat{O}(X))} \mathbb{1}\left\{C^{(j)} \in \hat{T}_{\lambda}^{(j)}(X), j \in \tilde{J}_{\lambda}(X)\right\} \right].$$

The reader should notice that in  $R_2$  and  $R_3$ , the term  $|\tilde{J}_{\lambda}(X)|$  penalizes spurious detections.

As with our FWER-controlling procedure, we developed a new method called split fixed sequence testing, described in detail in Appendix C. The basic inspiration for split fixed sequence testing is that when it is unclear how to construct the underlying graph for SGT, we can learn it from an extra split of data. As a final remark on the theory, once we select a set  $\hat{\Lambda}$  with a FWER-controlling procedure as in Proposition 7, we still have to select the specific element  $\hat{\lambda} \in \hat{\Lambda}$  to report our results. Because we use FWER control to pick  $\hat{\Lambda}$ , we can optimize over the parameters however we wish. Accordingly, we follow the following procedure (note that all coordinates of  $\lambda$

depend on each other, and must be chosen jointly):

$$\begin{aligned}\hat{\lambda}_3 &= \min_{\lambda \in \hat{\Lambda}} \left\{ \lambda_3 : \exists \lambda_1 \in \hat{\Lambda}, 1 - \lambda_1 < \lambda_3 \right\} \\ \hat{\lambda}_1 &= \max_{\lambda \in \hat{\Lambda}} \left\{ \lambda_1 : \lambda_3 = \hat{\lambda}_3 \right\} \\ \hat{\lambda}_2 &= \max_{\lambda \in \hat{\Lambda}} \left\{ \lambda_2 : [\hat{\lambda}_1, \lambda_2, \hat{\lambda}_3] \in \arg \min_{\lambda \in \hat{\Lambda}} \hat{R}_2(\lambda) \right\},\end{aligned}$$

which optimizes over  $\hat{\Lambda}$  to get meaningful prediction sets, many detections, and a high IOU.

We again used the MS-COCO dataset for experiments, this time using Facebook AI Research’s `detectron2` library to provide a pretrained detector. We modified the detector to give raw softmax outputs for each object’s class and raw sigmoid outputs for the binary masks. These became the functions  $\hat{f}$  and  $\hat{g}$  we described earlier. We discretized the space, picking

$$\Lambda = \underbrace{\{0.2, \dots, 0.5\}}_{\text{linspace}(0.2, 0.5, 50)} \times \underbrace{\{0.3, \dots, 0.7\}}_{\text{linspace}(0.3, 0.7, 5)} \times \underbrace{\{0.99, \dots, 1\}}_{\text{logspace}(-0.00436, 0, 25)} .$$

Over 1000 random splits of the 5000 point validation set into a calibration set of 3000 points and a validation set of 2000 points, we produced violin plots of the three risks using the Bonferroni version of our method, with the target levels  $\alpha = [0.5, 0.5, 0.75]$  and  $\delta = 0.1$ . Although the levels  $\alpha$  are nominally higher than they were in previous examples, they are quite stringent in this context. For example, note that  $R_3$  almost looks like coverage, but in practice is much more difficult to achieve because of the average over all predicted objects; if an object is predicted that does not match with a ground truth object, it counts as a miscoverage event. So, the level 0.75 is challenging.

Lastly, the reader may notice that the `recall` and `IOU` violin plots in Figure 10 are conservative. This is due to the algorithm for choosing  $\hat{\lambda}$ ; we choose  $\lambda_3$  first,  $\lambda_1$  second, and  $\lambda_2$  last. By choosing  $\lambda_3$  first, we told the algorithm to choose small prediction sets, then to optimize `recall` and `IOU` to be both controlled and as far from the nominal level as possible, provided the constraint on `coverage` holds. This is why the `coverage` is controlled tightly, while the other risks are farther away from the constraint (as shown in Figure 10). Importantly, all risks are always controlled, so the calibrated detector can be used with confidence.

## 8 Discussion

Our examples show that many reliability guarantees now become possible because of the ability of our new framework to tightly control (multiple) non-monotonic risks in practical settings. Furthermore, the SGT framework, combined with the reframing of risk control as hypothesis testing, allows for very tight control of these risks. This connection is somewhat surprising; SGT was developed within the medical statistics community for very small hypothesis spaces, such as two major and two minor endpoints of a clinical trial. However, in our work the graphical testing framework is most useful—in fact, nearly required—for large hypothesis spaces.

On a similar note, the reader may wonder why uniform concentration performs so poorly on our examples, given its long history of use for the analysis of machine learning algorithms. This fact is even more surprising given our use of various advanced techniques in uniform concentration that yield sharp constants [e.g. 68, 69]. Roughly, the reason for this conservativeness is that the existing techniques to derive the concentration inequalities apply the union bound (i.e., the Bonferroni correction) over a large grid of values. By contrast, a carefully designed multiple testing approach judiciously prioritizes a smaller set of promising hypotheses, avoiding the severe multiplicity correction.

Turning to open questions, we first mention what has become a standard surgeon-general’s warning in the field of distribution-free statistics: all of the guarantees in this paper are marginal. The reader should internalize what this means: we cannot guarantee that the errors are balanced over different strata of  $X$ - and  $Y$ -space, even meaningful ones such as object class, race, sex, illumination, et cetera. All of the errors may occur in one pathological bin—although it is possible to guard against such behaviors by designing a good score and evaluating the algorithms over relevant strata [11]. Extending the proposed techniques to have errors exactly or approximately balanced across strata is an open direction of great importance.

The work presented herein opens many other questions for future work. We have not explored the question of optimal testing procedures—neither when they exist, nor what they would be, nor when they are SGTs. Additionally, the SGTs used above are relatively simple and somewhat hand-designed heuristic procedures. One could hope to do better by learning the graph from another data split. We are not aware of work addressing this topic—such methods would not have been relevant in the medical statistics community that originated SGTs. On a different note, the capability to tightly control non-monotonic risks should enable computer scientists and statisticians in many applied areas to more carefully audit and certify their algorithmic predictions. We hope our examples demonstrate to this community that, even in complex setups with sophisticated algorithms, pragmatic and explicit statistical guarantees are possible.

## References

- [1] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. New York, NY, USA: Springer, 2005.
- [2] S. Bates, A. Angelopoulos, L. Lei, J. Malik, and M. I. Jordan, “Distribution-free, risk-controlling prediction sets,” *Journal of the ACM*, vol. 68, no. 6, Sep. 2021.
- [3] S. S. Wilks, “Determination of sample sizes for setting tolerance limits,” *Annals of Mathematical Statistics*, vol. 12, no. 1, pp. 91–96, 1941.
- [4] —, “Statistical prediction with special reference to the problem of tolerance limits,” *Annals of Mathematical Statistics*, vol. 13, no. 4, pp. 400–409, 1942.
- [5] A. Wald, “An extension of Wilks’ method for setting tolerance limits,” *Annals of Mathematical Statistics*, vol. 14, no. 1, pp. 45–55, 1943.
- [6] J. W. Tukey, “Non-parametric estimation II. Statistically equivalent blocks and tolerance regions—the continuous case,” *Annals of Mathematical Statistics*, vol. 18, no. 4, pp. 529–539, 1947.
- [7] K. Krishnamoorthy and T. Mathew, *Statistical Tolerance Regions: Theory, Applications, and Computation*. Wiley, 2009.
- [8] S. Park, O. Bastani, N. Matni, and I. Lee, “PAC confidence sets for deep neural networks via calibrated prediction,” in *International Conference on Learning Representations (ICLR)*, 2020.
- [9] S. Park, E. Dobriban, I. Lee, and O. Bastani, “PAC prediction sets under covariate shift,” *arXiv preprint arXiv:2106.09848*, 2021.
- [10] V. Vovk, A. Gammerman, and C. Saunders, “Machine-learning applications of algorithmic randomness,” in *International Conference on Machine Learning*, 1999, pp. 444–453.
- [11] A. N. Angelopoulos and S. Bates, “A gentle introduction to conformal prediction and distribution-free uncertainty quantification,” *arXiv preprint arXiv:2107.07511*, 2021.
- [12] H. Papadopoulos, K. Proedrou, V. Vovk, and A. Gammerman, “Inductive confidence machines for regression,” in *Machine Learning: European Conference on Machine Learning*, 2002, pp. 345–356.
- [13] J. Lei, A. Rinaldo, and L. Wasserman, “A conformal prediction approach to explore functional data,” *Annals of Mathematics and Artificial Intelligence*, vol. 74, pp. 29–43, 2015.
- [14] V. Vovk, “Cross-conformal predictors,” *Annals of Mathematics and Artificial Intelligence*, vol. 74, no. 1-2, pp. 9–28, 2015.
- [15] R. F. Barber, E. J. Candès, A. Ramdas, and R. J. Tibshirani, “Predictive inference with the jackknife+,” *The Annals of Statistics*, vol. 49, no. 1, pp. 486–507, 2021.
- [16] M. Sadinle, J. Lei, and L. Wasserman, “Least ambiguous set-valued classifiers with bounded error levels,” *Journal of the American Statistical Association*, vol. 114, pp. 223–234, 2019.
- [17] R. Barber, E. Candès, A. Ramdas, and R. Tibshirani, “The limits of distribution-free conditional predictive inference,” *Information and Inference*, vol. 10, no. 2, pp. 455–482, Aug. 2021.
- [18] Y. Romano, E. Patterson, and E. Candès, “Conformalized quantile regression,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 3543–3553.
- [19] R. Izbicki, G. Shimizu, and R. Stern, “Flexible distribution-free conditional predictive bands using density estimators,” in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020, pp. 3068–3077.

- [20] Y. Romano, M. Sesia, and E. Candès, “Classification with valid and adaptive coverage,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 3581–3591.
- [21] M. Cauchois, S. Gupta, and J. C. Duchi, “Knowing what you know: Valid and validated confidence sets in multiclass and multilabel prediction,” *Journal of Machine Learning Research*, vol. 22, no. 81, pp. 1–42, 2021.
- [22] L. Guan, “Conformal prediction with localization,” *arXiv preprint arXiv:1908.08558*, 2019.
- [23] A. N. Angelopoulos, S. Bates, J. Malik, and M. I. Jordan, “Uncertainty sets for image classifiers using conformal prediction,” in *International Conference on Learning Representations (ICLR)*, 2021.
- [24] J. Lei, “Classification with confidence,” *Biometrika*, vol. 101, no. 4, pp. 755–769, Oct. 2014.
- [25] Y. Hechtlinger, B. Póczos, and L. Wasserman, “Cautious deep learning,” *arXiv preprint arXiv:1805.09460*, 2018.
- [26] L. Guan and R. Tibshirani, “Prediction and outlier detection in classification problems,” *arXiv preprint arXiv:1905.04396*, 2019.
- [27] V. Vovk, I. Petej, P. Toccaceli, A. Gammerman, E. Ahlberg, and L. Carlsson, “Conformal calibrators,” in *Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications*, vol. 128, 2020, pp. 84–99.
- [28] L. Lei and E. J. Candès, “Conformal inference of counterfactuals and individual treatment effects,” *arXiv preprint arXiv:2006.06138*, 2020.
- [29] E. J. Candès, L. Lei, and Z. Ren, “Conformalized survival analysis,” *arXiv preprint arXiv:2103.09763*, 2021.
- [30] A. N. Angelopoulos, S. Bates, T. Zrnic, and M. I. Jordan, “Private prediction sets,” *arXiv preprint arXiv:2102.06202*, 2021.
- [31] S. Bates, E. Candès, L. Lei, Y. Romano, and M. Sesia, “Testing for outliers with conformal p-values,” *arXiv preprint arXiv:2104.08279*, 2021.
- [32] A. Fisch, T. Schuster, T. Jaakkola, and R. Barzilay, “Efficient conformal prediction via cascaded inference with expanded admission,” *arXiv preprint arXiv:2007.03114*, 2020.
- [33] T. Schuster, A. Fisch, T. Jaakkola, and R. Barzilay, “Consistent accelerated inference via confident adaptive transformers,” *arXiv preprint arXiv:2104.08803*, 2021.
- [34] A. Fisch, T. Schuster, T. Jaakkola, and R. Barzilay, “Few-shot conformal prediction with auxiliary tasks,” *arXiv preprint arXiv:2102.08898*, 2021.
- [35] V. Chernozhukov, K. Wüthrich, and Z. Yinchu, “Exact and robust conformal inference methods for predictive machine learning with dependent data,” in *Proceedings of the 31st Conference On Learning Theory*, ser. PMLR, vol. 75, Jun. 2018, pp. 732–749.
- [36] R. Dunn, L. Wasserman, and A. Ramdas, “Distribution-free prediction sets with random effects,” *arXiv preprint arXiv:1809.07441*, 2020.
- [37] R. J. Tibshirani, R. Foygel Barber, E. Candès, and A. Ramdas, “Conformal prediction under covariate shift,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 2530–2540.
- [38] M. Cauchois, S. Gupta, A. Ali, and J. C. Duchi, “Robust validation: Confident predictions even when distributions shift,” *arXiv preprint arXiv:2008.04267*, 2020.
- [39] X. Hu and J. Lei, “A distribution-free test of covariate shift using conformal prediction,” *arXiv preprint arXiv:2010.07147*, 2020.
- [40] I. Gibbs and E. Candès, “Adaptive conformal inference under distribution shift,” *arXiv preprint arXiv:2106.00170*, 2021.
- [41] V. Vovk, “Testing Randomness Online,” *Statistical Science*, vol. 36, no. 4, pp. 595–611, 2021.
- [42] V. Vovk and C. Bendtsen, “Conformal predictive decision making,” in *Proceedings of the Seventh Workshop on Conformal and Probabilistic Prediction and Applications*, ser. Proceedings of Machine Learning Research, vol. 91, 2018, pp. 52–62.
- [43] V. Vovk, “Universally consistent conformal predictive distributions,” in *Proceedings of the Eighth Symposium on Conformal and Probabilistic Prediction and Applications*, ser. PMLR, vol. 105, Sep. 2019, pp. 105–122.
- [44] P. Bauer, “Multiple testing in clinical trials,” *Statistics in Medicine*, vol. 10, no. 6, pp. 871–890, 1991.

- [45] S. Holm, “A simple sequentially rejective multiple test procedure,” *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.
- [46] W. Hoeffding, “Probability inequalities for sums of bounded random variables,” *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.
- [47] V. Bentkus, “On Hoeffding’s inequalities,” *The Annals of Probability*, vol. 32, no. 2, pp. 1650–1673, 2004.
- [48] C. Bonferroni, “Teoria statistica delle classi e calcolo delle probabilita,” *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, vol. 8, pp. 3–62, 1936.
- [49] E. Sonnemann, H. Finner, and T. J. Kunert, “Analyse von verlaufskurven,” *Biometriekurs*, 1986.
- [50] F. Bretz, W. Maurer, W. Brannath, and M. Posch, “A graphical approach to sequentially rejective multiple test procedures,” *Statistics in Medicine*, vol. 28, no. 4, pp. 586–604, 2009.
- [51] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in *European Conference on Computer Vision*, Springer, 2014, pp. 740–755.
- [52] T. Ridnik, H. Lawen, A. Noy, and I. Friedman, “TResNet: High performance GPU-dedicated architecture,” *arXiv:2003.13630*, 2020.
- [53] J. D. Storey, “The positive false discovery rate: a Bayesian interpretation and the q-value,” *The Annals of Statistics*, vol. 31, no. 6, pp. 2013–2035, 2003.
- [54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.
- [55] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [56] J. W. Cohen, S. B. Cohen, and J. S. Bantlin, “The medical expenditure panel survey: A national information resource to support healthcare cost research and inform policy and practice,” *Medical Care*, S44–S50, 2009.
- [57] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Annals of Statistics*, pp. 1189–1232, 2001.
- [58] A. N. Angelopoulos, A. P. Kohli, S. Bates, M. I. Jordan, J. Malik, T. Alshaabi, S. Upadhyayula, and Y. Romano, “Image-to-image regression with distribution-free uncertainty quantification and applications in imaging,” *arXiv preprint arXiv:2202.05265*, 2022.
- [59] L. Breiman, “Arcing the edge,” Technical Report 486, Statistics Department, University of California at Berkeley, Tech. Rep., 1997.
- [60] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, *et al.*, “Scipy 1.0: Fundamental algorithms for scientific computing in python,” *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.
- [61] B. L. Wiens, “A fixed sequence bonferroni procedure for testing multiple endpoints,” *Pharmaceutical Statistics: The Journal of Applied Statistics in the Pharmaceutical Industry*, vol. 2, no. 3, pp. 211–215, 2003.
- [62] B. L. Wiens and A. Dmitrienko, “The fallback procedure for evaluating a single family of hypotheses,” *Journal of Biopharmaceutical Statistics*, vol. 15, no. 6, pp. 929–942, 2005.
- [63] A. Krizhevsky, G. Hinton, *et al.*, *Learning multiple layers of features from tiny images*, <https://www.cs.toronto.edu/kriz/learning-features-2009-TR.pdf>, 2009.
- [64] S. Liang, Y. Li, and R. Srikant, “Enhancing the reliability of out-of-distribution image detection in neural networks,” *arXiv preprint arXiv:1706.02690*, 2017.
- [65] R. Izbicki, G. T. Shimizu, and R. B. Stern, “Flexible distribution-free conditional predictive bands using density estimators,” *arXiv preprint arXiv:1910.05575*, 2019.
- [66] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [67] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, *Detectron2*, <https://github.com/facebookresearch/detectron2>, 2019.

- [68] M. Anthony and J. Shawe-Taylor, “A result of Vapnik with applications,” *Discrete Applied Mathematics*, vol. 47, no. 3, pp. 207–217, 1993.
- [69] J. Parrondo and C. Van den Broeck, “Vapnik-Chervonenkis bounds for generalization,” *Journal of Physics A: Mathematical and General*, vol. 26, no. 9, p. 2211, 1993.
- [70] J. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” *Advances in Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.
- [71] R. Durrett, *Probability: Theory and Examples*, Second. Belmont, CA: Duxbury Press, 1996, pp. xiii+503.
- [72] A. Hole, “Some variants on a theorem of Vapnik,” *Preprint series: Pure mathematics <http://urn.nb.no/URN:NBN:no-8076>*, 1995.
- [73] P. Bartlett and G. Lugosi, “An inequality for uniform deviations of sample averages from their means,” *Statistics & Probability Letters*, vol. 44, no. 1, pp. 55–62, 1999.
- [74] B. Bercu, E. Gassiat, and E. Rio, “Concentration inequalities, large and moderate deviations for self-normalized empirical processes,” *The Annals of Probability*, vol. 30, no. 4, pp. 1576–1604, 2002.
- [75] E. Giné, V. Koltchinskii, and J. A. Wellner, “Ratio limit theorems for empirical processes,” in *Stochastic inequalities and applications*, Springer, 2003, pp. 249–278.
- [76] V. Vapnik and A. Y. Chervonenkis, “On the uniform convergence of relative frequencies of events to their probabilities,” *Theory of Probability & Its Applications*, vol. 16, no. 2, pp. 264–280, 1971.
- [77] R. Bardenet, O.-A. Maillard, *et al.*, “Concentration inequalities for sampling without replacement,” *Bernoulli*, vol. 21, no. 3, pp. 1361–1385, 2015.
- [78] I. Pinelis, “An asymptotically Gaussian bound on the Rademacher tails,” *Electronic Journal of Probability*, vol. 17, 2012.
- [79] V. K. Bentkus and D. Dzindzalieta, “A tight Gaussian bound for weighted sums of Rademacher random variables,” *Bernoulli*, vol. 21, no. 2, pp. 1231–1237, 2015.
- [80] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 1995.
- [81] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities*. Oxford University Press, 2013.



# A Experimental Details and Additional Results

## A.1 Numerical comparisons

We directly compare the numerical performance of the methods that we have presented in a synthetic first-order autoregressive (AR) process. An AR process is a sequence of random variables where the elements in the sequence depend on the previous value plus a noise term. Although the elements in the sequence are correlated, the mean of the process can be designed to have any shape. Loosely, we model the loss of example  $i \in \{1, \dots, n\}$  as an independently drawn AR process indexed by a one dimensional ordered set of  $\Lambda$ , with a “V”-shaped mean. In other words, the risk is “V”-shaped, with the tip falling below  $\alpha$ . We will make this description mathematically concrete in a moment.

In addition to the multiple testing strategies presented in the main text, we consider an alternative approach outlined in Appendix E, using uniform concentration results to guarantee risk control. Unfortunately, this approach was seen to be more conservative than the multiple testing approach in the main-text examples.

Now, we turn to the concrete definition of our AR process. We consider a case where  $\Lambda = \{0.001, 0.002, \dots, 1\}$  and the true risk  $R(\lambda)$  is “V”-shaped, falling below the desired level  $\alpha$  in the center of the interval; see Figure 11. For each observation  $i = 1, \dots, n$ , we simulate losses for  $\mathcal{T}_{\lambda_j}$  from the following first-order autoregressive model:

$$L_{i,j} = \Phi(u_j + \mu_j)$$

$$u_j = \text{corr} \cdot u_{j-1} + \sqrt{1 - \text{corr}^2} \cdot \mathcal{N}(0, 1).$$

Here, the  $\mu_j$  are chosen in such a way as to create the “V”-shaped risk curve; note that there is no dependence on  $i$  anywhere, because an independent random process gets generated for each sample  $i \in \{1, \dots, n\}$ . From this simulated data, we form p-values for each hypothesis in (1) using the Hoeffding-Bentkus bound from Section 2.3. Then, we apply the Bonferroni test and the fixed sequence test to these p-values.

Before turning to the results, we pause to think about the meaning of our setting. Here, we wish to find a value of  $\lambda$  such that the risk is below  $\alpha$ , but do not know in advance where this may happen. Therefore, we use the data to test whether the risk is below  $\alpha$  for each value of  $\lambda$  using the calibration data. Note that because in practice all tests are based on the same calibration data, the results will be dependent, which is why our simulated setting includes the autoregressive correlation term.

The results are recorded in Figure 11, which we parse next. Firstly, the uniform concentration bound, perhaps predictably, performs badly in practice; see our discussion of this topic in Section 8. Secondly, the fixed sequence method is less conservative than the Bonferroni method, as expected; fixed sequence testing is designed to make the rejection set as wide as possible on the right side.

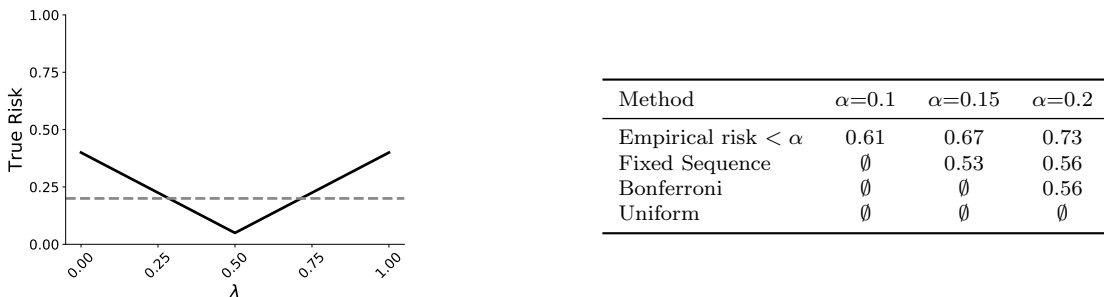


Figure 11: **Numerical comparison of FWER-controlling algorithms on a synthetic AR process.** On the left-hand side, we plot a representative of the “V”-shaped risk used to generate the AR process, along with the desired risk level  $\alpha = 0.1$  as a gray dotted line. In the right table, we report the rightmost endpoint of the set  $\hat{\Lambda}$  for different FWER controlling procedures (higher is less conservative). The subplots show different levels  $\alpha$  and correlation parameters for the AR process as in Section A.1. An entry of  $\emptyset$  means that the procedure failed to reject anywhere:  $\hat{\Lambda} = \emptyset$ . We used  $n = 5000$  data points,  $N = 1000$  evenly spaced grid points of  $\lambda$  between 0 and 1,  $\text{corr} = 0.9$ , and  $\delta = 0.1$ . The entry titled ‘Empirical risk  $< \alpha$ ’ in the table is a baseline procedure that selects the largest  $\lambda$  such that the empirical risk is below  $\alpha$ . This baseline does not control the risk, and represents an upper bound on what we can hope to achieve with a risk-controlling procedure.

## A.2 Additional experimental results for FDR control

In Figure 12, we report additional results from the experiment from Section 3, this time with desired FDR level  $\alpha = 0.5$ .

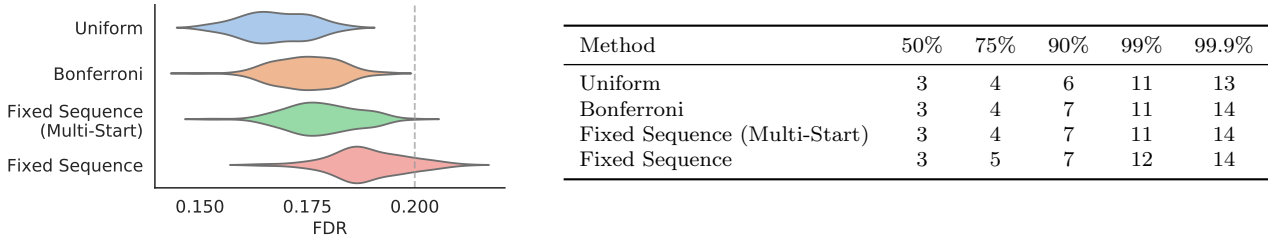


Figure 12: **Numerical results of our multi-label classification procedure on MS COCO.** This plot is the same figure as Figure 3 in the main text, but with  $\alpha = 0.5$ .

## A.3 Details about the ODIN score

Next, we precisely define the ODIN function, which we use for as an OOD score. Let  $\hat{f}_y(x)$  be the output of the DenseNet’s final, temperature-scaled [70] softmax layer, meant to estimate  $\mathbb{P}(Y | X)$ . Additionally, let  $\hat{f}_{(i)}(x)$  be the softmax output of the  $i$ -th most likely class. The key step in ODIN is to add a small perturbation to the input image designed to help distinguish between in-distribution and out-of-distribution images. In particular, the perturbation

$$w(X) = X - \epsilon \text{sign} \left( -\nabla_X \log \hat{f}_{(1)}(X) \right)$$

encourages the top softmax output to be more confident on the perturbed input  $w(X)$ . The tuning parameter  $\epsilon$  is the magnitude of the perturbation—we took it to be 0.0014, the default value from the ODIN GitHub repository. Then, the OOD function is taken to be

$$\text{OOD}(X) = 1 - \hat{f}(w(X))_{(1)}.$$

The reader can view ODIN as an improvement upon the obvious OOD function,  $1 - \hat{f}(X)_{(1)}$ ; the improvement relies on the empirical observation that the top softmax output grows more for in-distribution images when perturbed than it does for out-of-distribution images.

We make one final note here about the scaling of the ODIN function. With this particular model on the CIFAR-10 dataset, all values of  $\text{OOD}(X)$  were between the numbers 0.890 and 0.899. In other words, this score is extremely poorly scaled; when choosing 1000 equally spaced points between 0 and 1000, only 9 of them will land in this interval. We re-scaled ODIN manually to fix this problem, although it could be fixed using the training dataset.

## B Proofs

*Proof of Theorem 1.* The result is immediate. □

*Proof of Proposition 1.* This is a combination of the results from Hoeffding [46] and Bentkus [47]. See [2] for details on how to hybridize the two inequalities. □

*Proof of Proposition 2.* This is a straightforward consequence of the central limit theorem. A statement can be found in any graduate probability textbook, such as Durrett [71]. □

*Proof of Proposition 3.* This is a classical result in multiple testing. For example, see [45] for discussion. □

*Proof of Proposition 4.* Fixed sequence testing has existed in the literature for some time; see, e.g., Sonnemann, Finner, and Kunert [49] and Bauer [44]. We include a proof of the version we use for completeness.

Consider first the case where  $|\mathcal{J}| = 1$ . Then there will be a first index at which you encounter a null. The probability of making a false discovery at that index is bounded by  $\delta$ . Thus, the probability of making any false discoveries is bounded by  $\delta$ .

Turning to the  $|\mathcal{J}| > 1$  case, the procedure is equivalent to running many instances of the  $|\mathcal{J}| = 1$  procedure in parallel, at level  $\delta/|\mathcal{J}|$ . By the union bound, the probability of any false rejections is then bounded by  $\delta$ .  $\square$

*Proof of Proposition 5.* Since the procedure is sequential, it rejects a null iff it rejects  $H_{j^*}$ , in which case  $p_{j^*} \leq \delta$ .  $\square$

*Proof of Proposition 7.* The result is immediate.  $\square$

*Proof of Proposition 8.* By the definition of a uniform bound,  $R^+$  fails with probability at most  $\alpha$  at  $\hat{\lambda}$ .  $\square$

## C Split Fixed Sequence Testing

In Section 2.3.4, we showed how any pre-specified SGT can control the FWER. The technique is flexible and powerful, especially for situations where it is straightforward to hand-design a graph, such as Figure 6 in the OOD detection example. However, for the large-scale machine learning systems we use, designing the graph is not always scaleable. For example, consider the object detection example in Section 7. Because all three coordinates of  $\lambda$  affect all three risks, and we do not know *a-priori* which values of  $\lambda$  control the risks, there is no graph that is obviously best. In such situations, we might desire an automated procedure that defines the graph by looking at promising hypotheses in our data; in other words, we seek to *learn the graph*.

The procedure we define here, which we call *split fixed sequence testing*, is a very simple way of learning the graph from an extra data split. The idea is inspired by the following fact: in many settings, we hope to identify points that violate several risk functions equally often. Therefore, we pick a sequence of  $\lambda$  that have nearly the same p-value using the first split of data. Then, we simply use fixed sequence testing on this selected sequence using the fresh data. This allows us to decide our path through  $\Lambda$  using data while still providing rigorous FWER control.

Concretely, the algorithm begins by partitioning the calibration data points  $\mathcal{I}_{\text{cal}}$  into a graph selection set  $\mathcal{I}_{\text{graph}}$  and a multiple testing set  $\mathcal{I}_{\text{testing}}$ . We will learn our path through  $\Lambda$  only by looking at  $\mathcal{I}_{\text{graph}}$ ; we will reserve the other split of data simply for running SGT. Following the notation from Section 2.4, and calculating the p-values only on the graph selection set  $\mathcal{I}_{\text{graph}}$ , we define the function

$$\tilde{\lambda}(\beta) = \lambda_j, \text{ where } j = \arg \min_{j'} \left\| [p_{j,1}, \dots, p_{j,m}] - [\beta, \dots, \beta] \right\|_{\infty},$$

where  $\beta$  ranges from 0 to 1. The function  $\tilde{\lambda}(\beta)$  picks a point where the p-values for all risks are nearly equal to  $\beta$ . We parameterize our path by discretizing this function. For some positive integer  $D$ , we set

$$\tilde{\lambda}_d = \tilde{\lambda} \left( \frac{d}{D} \right), \text{ for } d = 0, 1, \dots, D.$$

Note that in practice, this sequence can have repeated values, usually adjacent to one another; we remove them in practice, and ignore them for notational convenience.

Once we have the sequence  $\{\tilde{\lambda}_d\}_{d=0}^D$ , we do fixed sequence testing directly on the sequence as it is naturally ordered using the multiple testing set  $\mathcal{I}_{\text{testing}}$ . Figure 10 shows the result of this procedure in our object detection example.

## D Hamming-equalized SGT

Recall that the SGT procedure from Section 2.3 allows us to design any graph to propagate our error budget while still controlling type-I error. We considered two hand-designed graphs in the main text: the fallback procedure

and the so-called Hamming graph presented in Figure 6. Now, we return to the Hamming graph, which improves upon the fallback procedure under certain conditions, and discuss its properties here. In the Hamming graph, all of the error budget starts in the bottom-right corner, and propagates through two subparts of the graph. In the lower-right triangle, the graph perfuses the error budget such that nodes with equal Hamming distance from the bottom-right node are designed to receive the same error budget if all previous nodes reject. We will call this property *balanced inflow*; see Definition 3 below. In the upper-left triangle, the graph propagates the error leftwards.

Concretely, we associate the index  $(i, j)$  to the node in the  $i$ th row from the bottom and the  $j$ th row from the right of the graph. Then we set the weights of our graph to

$$W_{(i,j) \rightarrow (i,j+1)} = \begin{cases} \frac{j}{i+j} & i+j \leq N \\ 1 & i+j > N \text{ and } i = N \\ 0 & \text{else,} \end{cases}$$

and

$$W_{(i,j) \rightarrow (i+1,j)} = \begin{cases} \frac{i}{i+j} & i+j \leq N \\ 1 & i+j > N \text{ and } i < N \\ 0 & \text{else,} \end{cases}$$

with all other edge weights equaling zero. Setting these edge weights creates a graph where all nodes in the bottom-right triangle have *balanced inflow*, i.e.,

**Definition 3** (Balanced Inflow). *Let  $i + j = L$  and consider the set of incoming paths to node  $(i, j)$  as*

$$\mathcal{P}(i, j) = \left\{ (i_1, j_1), (i_2, j_2), \dots, (i_{L-1}, j_{L-1}), (i_L, j_L), \text{ where } (i_1, j_1) = (1, 1) \text{ and } (i_L, j_L) = (i, j) \right\}.$$

*A graph has balanced inflow if*

$$\sum_{\substack{\{(i_{i'}, j_{i'})\}_{i'=1}^L \\ \in \mathcal{P}(i, j)}}} \prod_{l=1}^{L-1} W_{(i_l, j_l) \rightarrow (i_{l+1}, j_{l+1})} = \frac{1}{L-1}.$$

In English, this property encodes the idea of balanced inflow we discussed earlier—nodes that are equally far away from the root can all potentially receive the same error budget (although in practice, rejections in the lower-right triangle may thwart this effort). Proving that the Hamming SGT has balanced inflow is simple by induction on  $L$ . This graph structure plays well with coordinatewise monotonicity and other such settings where the goal is to push from one corner to another in the space of hypotheses.

At a high level, the Hamming SGT improves on the fallback procedure when there are many rejections in the lower-right triangle. Indeed, the Hamming SGT is much more powerful in this region, since the error budget is more concentrated. In the best-case scenario, where all of the lower-right triangle is rejected, each node along the main diagonal gets a budget of  $\frac{\delta}{N-1}$ , and a fallback procedure is performed on the upper-left triangle. However, the graph's structure can also hurt us; if a large fraction of nulls fail to be rejected in the lower-right triangle, the procedure could be less effective in rejecting an extreme setting of  $\lambda$ . Thus, we suggest the use of the Hamming SGT if there is reason to believe that the lower-right triangle will contain many rejections. We repeat that balanced inflow is not a necessary condition for a SGT graph to perform well, and in Section 6 we find that other approaches are more powerful.

## E Alternative Approach: Uniform Concentration

Rather than use multiple testing, one can consider creating risk-controlling predictions using uniform concentration results. We outline this strategy below. Unfortunately, this approach is typically much more conservative than the multiple testing approach, even when using state-of-the-art concentration results. This is evidenced in our many simulations. Nonetheless, since this is a seemingly natural alternative, we record this approach in detail here.

## E.1 Framework

We stated how uniform bounds lead to risk control in Proposition 8.

The main challenge is to come up with a valid uniform confidence bound  $R^+$ . We have developed such bounds, stating the technical details in Appendix E.2. In the experiments presented in the main text, we find that this is less powerful than the multiple-testing approach.

## E.2 A new class of concentration inequalities for self-normalized empirical processes

We next develop a state-of-the-art set of uniform concentration results for our setting. Recalling that

$$\hat{R}(\mathcal{T}_\lambda) = \frac{1}{n} \sum_{i=1}^n L(\mathcal{T}_\lambda(X_i), Y_i),$$

we can view  $\{\hat{R}(\mathcal{T}_\lambda) : \lambda \in \Lambda\}$  as an empirical process indexed by  $\lambda$ . The simplest empirical process is the empirical CDF. The celebrated Dvoretzky-Kiefer-Wolfowitz-Massart (DKWM) inequality bounds the absolute difference  $|\hat{F}_n(w) - F(w)|$ . Although the constant is tight, the sup-difference metric ignores the non-constant variance of  $\hat{F}_n(w)$ . Specifically,  $n\hat{F}_n(w) \sim \text{Binom}(n, F(w))$  and thus  $\text{Var}[\hat{F}_n(w)] = F(w)(1 - F(w))/n$ . As a result, when  $F(w)$  is close to 0 or 1,  $\hat{F}_n(w)$  is more concentrated around  $F(w)$ .

To improve the DKWM bounds for tail events, a line of work has established concentration inequalities for self-normalized empirical processes [e.g. 68, 72–75]. In particular, they derived upper tail bounds for

$$\sup_{w \in \mathbb{R}} \frac{\hat{F}_n(w) - F(w)}{\sqrt{\hat{F}_n(w)}} \quad \text{and} \quad \sup_{w \in \mathbb{R}} \frac{F(w) - \hat{F}_n(w)}{\sqrt{F(w)}},$$

where  $\hat{F}_n(w)$  denotes the empirical CDF of  $n$  i.i.d. samples  $W_1, \dots, W_n$  drawn from the distribution  $F$ . For instance, [68] prove that, with probability  $1 - \delta$ ,

$$\sup_{w \in \mathbb{R}} \frac{F(w) - \hat{F}_n(w)}{\sqrt{F(w)}} \leq \sqrt{\frac{4}{n} \log \left( \frac{8(n+1)}{\alpha} \right)} = O \left( \sqrt{\frac{\log n}{n}} \right).$$

When  $\hat{F}_n(w) = O(\log n/n)$ , the Anthony–Shaw–Taylor inequality yields an upper confidence bound of  $F(w)$  of order  $O(\log n/n)$  as well, while the DKWM inequality implies an upper confidence bound of  $F(w)$  of order  $O(1/\sqrt{n})$ . Thus the former yields a tighter results for rare events for large  $n$ .

In our context,  $\hat{R}(\mathcal{T}_\lambda)$  is more complicated than the empirical CDF because the summands are no longer binary. We adapt the proofs of [68], [69], and [76] to the general case and further improve their constants using the recently developed tools for sampling without replacement [77] and weighted sums of Rademacher variables [78, 79]. To state the general results, we consider a generic empirical process

$$\hat{s}_n(\lambda) = \frac{1}{n} \sum_{i=1}^n S(\lambda; Z_i), \quad s(\lambda) = \mathbb{E}[S(\lambda; Z_i)], \quad S(\lambda; Z_i) \in [0, 1] \text{ almost surely for every } \lambda \in \Lambda. \quad (7)$$

Note that  $S(\lambda; Z_i) = L(\mathcal{T}_\lambda(X_i), Y_i)$  in our context. Furthermore, we define  $\Delta(n)$  as the

$$\Delta(n) = \sup_{z_1, \dots, z_n} \left| \{ \{ S(\lambda; z_1), \dots, S(\lambda; z_n) \} : \lambda \in \Lambda \} \right|.$$

In the literature,  $\log \Delta(n)$  is often referred to as the growth function [80, Section 2].

**Theorem E.1.** *Under the setting (7), for any  $\eta \geq 0$ ,*

$$\mathbb{P} \left( \sup_{\lambda \in \Lambda} \frac{\hat{s}_n(\lambda) - s(\lambda)}{\sqrt{\hat{s}_n(\lambda) + \eta}} \geq t \right) \leq \inf_{\gamma \in (0, 1), n' \in \mathbb{Z}^+} \frac{\Delta(n + n') \exp\{-g_2(t; n, n', \gamma, \eta)\}}{1 - \exp\{-g_1(t; n', \gamma, \kappa^+)\}}, \quad (8)$$

and

$$\mathbb{P}\left(\sup_{\lambda \in \Lambda} \frac{s(\lambda) - \hat{s}_n(\lambda)}{\sqrt{s(\lambda) + \eta}} \geq t\right) \leq \inf_{\gamma \in (0,1), n' \in \mathbb{Z}^+} \frac{\Delta(n+n') \exp\{-g_2(t; n, n', \gamma, \kappa^-)\}}{1 - \exp\{-g_1(t; n', \gamma, \eta)\}}, \quad (9)$$

where

$$g_1(t; n', \gamma, \kappa) = \max\left\{\frac{n't^2}{2} \frac{\gamma^2}{1 + \gamma^2 t^2 / 36\kappa}, \log\left(\frac{n't^2 \gamma^2}{(\sqrt{1 + \kappa} - \sqrt{\kappa})^2}\right)\right\},$$

$$g_2(t; n, n', \gamma, \kappa) = \frac{nt^2}{2} \left(\frac{n'}{n+n'}\right)^2 \frac{(1-\gamma)^2}{1 + (1-\gamma)^2 t^2 / 36\kappa},$$

and

$$\kappa^+ = \eta + \frac{t^2}{2} + t\sqrt{\frac{t^2}{4} + \eta}, \quad \kappa^- = \eta + \frac{n + \gamma n'}{n + n'} \sqrt{\kappa^+}$$

**Theorem E.2.** Under the setting (7), for any  $\eta \geq 0$ ,

$$\mathbb{P}\left(\sup_{\lambda \in \Lambda} \frac{\hat{s}_n(\lambda) - s(\lambda)}{\sqrt{\hat{s}_n(\lambda) + \eta}} \geq t\right) \leq \inf_{\gamma \in (0,1)} \frac{\Delta(2n) \tilde{g}\left(\sqrt{\frac{n(1+\eta)}{2}}(1-\gamma)t\right)}{1 - \exp\{-g_1(t; n, \gamma, \kappa^+)\}},$$

and

$$\mathbb{P}\left(\sup_{\lambda \in \Lambda} \frac{s(\lambda) - \hat{s}_n(\lambda)}{\sqrt{s(\lambda) + \eta}} \geq t\right) \leq \inf_{\gamma \in (0,1)} \frac{\Delta(2n) \tilde{g}\left(\sqrt{\frac{n(1+\eta)}{2}}(1-\gamma)t\right)}{1 - \exp\{-g_1(t; n, \gamma, \eta)\}}, \quad (10)$$

where  $g_1$  is defined in Theorem E.1,  $\tilde{g}(x) = \min\{\tilde{g}_1(x), \tilde{g}_2(x), \tilde{g}_3(x)\}$ ,

$$\tilde{g}_1(x) = c_1(1 - \Phi(x)), \quad c_1 = 1/4(1 - \Phi(\sqrt{2})) \approx 3.178,$$

$$\tilde{g}_2(x) = 1 - \Phi(x) + \frac{c_2}{9 + x^2} \exp\left\{-\frac{x^2}{2}\right\}, \quad c_2 = 5\sqrt{e}(2\Phi(1) - 1) \approx 5.628,$$

$$\tilde{g}_3(x) = \exp\left\{-\frac{x^2}{2}\right\}.$$

The proofs of both theorems are lengthy and relegated to Section E.5.

### E.3 Uniform upper confidence bound for RCP

From these general results, we obtain the following corollary, which will result in a practical algorithm momentarily.

**Corollary E.1.** Let

$$R^+(\mathcal{T}_\lambda) := \hat{R}(\mathcal{T}_\lambda) + t(\eta; \delta) \sqrt{\hat{R}(\mathcal{T}_\lambda) + \eta + \frac{t(\eta; \delta)^2}{4} + \frac{t(\eta; \delta)^2}{2}}.$$

where  $t(\eta; \delta)$  is defined as the  $t$  that solves

$$\delta = \inf_{\gamma \in (0,1), n' \in \mathbb{Z}^+} \min\left\{\frac{\Delta(n+n') \exp\{-g_2(t; n, n', \gamma, \kappa^-)\}}{1 - \exp\{-g_1(t; n', \gamma, \eta)\}}, \frac{\Delta(2n) \tilde{g}\left(\sqrt{\frac{n(1+\eta)}{2}}(1-\gamma)t\right)}{1 - \exp\{-g_1(t; n, \gamma, \eta)\}}\right\},$$

where the functions and quantities are defined in Theorem E.1 and E.2. Then

$$\mathbb{P}(R(\mathcal{T}_\lambda) \leq R^+(\mathcal{T}_\lambda) \text{ for all } \lambda \in \Lambda) \geq 1 - \delta.$$

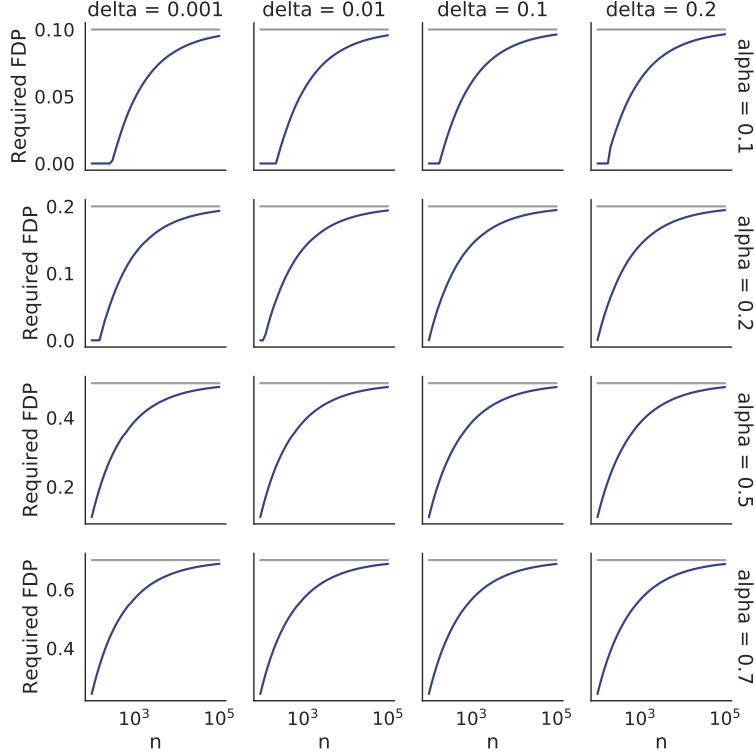


Figure 13: **Empirical FDP needed to achieve FDR control.** We plot several desired FDR levels and high-probability choices  $\delta$ . The gray horizontal line indicates the target FDR level  $\alpha$ .

### An optimal choice of $\eta$

The parameter  $\eta$  needs to be chosen in advance. For a given level  $(\alpha, \delta)$ , however, there is an optimal choice. Let

$$x(\eta; \delta) := \alpha - t(\eta; \delta)\sqrt{\alpha + \eta},$$

which is the largest solution  $x$  of

$$\alpha = x + t(\eta; \delta)\sqrt{x + \eta + \frac{t(\eta; \delta)^2}{4}} + \frac{t(\eta; \delta)^2}{2}.$$

This is the largest value for which  $\text{FDP}(\mathcal{T}) \leq x$  implies that  $\text{FDR}^+(\mathcal{T}) \leq \alpha$ . In other words, with  $\eta$  fixed, our procedure will find the largest  $\lambda$  such that  $\text{FDP}(\mathcal{T}_\lambda) \leq x(\eta; \delta)$ , and select this value as  $\hat{\lambda}$ . Thus, we should select the value of  $\eta$  that makes  $x(\eta; \delta)$  as small as possible;

$$\eta^*(\delta) := \arg \min_{\eta \geq 0} x(\eta; \delta)$$

is the best value of  $\eta$ .

## E.4 Concrete instantiation for FDR control

To apply Corollary E.1 for FDR control, we need to derive the growth function. Let  $Z_i = (X_i, Y_i)$ ,  $\{1, \dots, K\}$  be the set of labels, and

$$\Lambda_{ij} := \inf\{\lambda \in \mathbb{R} : j \notin \mathcal{T}_\lambda(X_i)\}$$

In our example,  $\mathcal{T}_\lambda$  is decreasing in  $\lambda$ . Thus,

$$\text{FDR} = \mathbb{E}[S(\lambda; Z_i)], \quad \text{where } S(\lambda; Z_i) = \frac{\sum_{j=1}^m I(\Lambda_{ij} \leq \lambda, j \notin Y_i)}{\sum_{j=1}^m I(\Lambda_{ij} \leq \lambda)}.$$

Here,  $0/0$  is defined as 0. Note that  $(S(\lambda; Z_1), \dots, S(\lambda; Z_n))$  changes value only when  $\lambda = \Lambda_{ij}$  for some  $i$  and  $j$ , and thus

$$\Delta(n) \leq nm + 1.$$

In Algorithm 3 below, we present the full algorithm corresponding to Corollary 8, specialized to the FDR control setting from Section 3 for concreteness. In addition, in Figure 13, we report numerical information about the size of the bound. In particular, we show the empirical FDR that must be achieved in order to conclude that the true FDR is below  $\alpha$  with probability at least  $1 - \delta$ ; that is,  $x(\eta^*; \delta)$  in our notation above. When this curve is farther below the nominal rate  $\alpha$ , it means that the procedure is quite conservative. We find that the procedure is heavily conservative until  $n$  is of order  $10^5$ . This provides further evidence that the uniform concentration approach is looser than the multiple testing approach, so it requires a much larger amount of calibration data.

---

**Algorithm 3** FDR Calibration via Uniform Concentration

---

**Input:** Nested-set-valued function  $\mathcal{T}_\lambda$ , desired FDR  $\alpha$ , calibration set  $(X_1, Y_1), \dots, (X_n, Y_n)$ , step size  $\zeta$ .

- 1: **procedure** LARGESTSET( $\mathcal{T}_\lambda, \alpha, (X_1, Y_1), \dots, (X_n, Y_n)$ )
- 2:      $\lambda \leftarrow 1$
- 3:     **while**  $\lambda > 0$  &  $\alpha \geq \widehat{\text{FDR}}(\mathcal{T}_\lambda) + t(\eta^*; \delta) \sqrt{\widehat{\text{FDR}}(\mathcal{T}_\lambda) + \eta^* + t(\eta^*; \delta)^2/4} + t(\eta^*; \delta)/2$  **do**
- 4:          $\lambda \leftarrow \lambda - \zeta$
- 5:     **return**  $\lambda$

**Output:** A parameter  $\hat{\lambda}$  that controls the FDR at level  $\alpha$  with probability  $\delta$ .

---

## E.5 Technical proofs

*Proof of Corollary E.1.* Let  $S(\lambda; Z_i) = L(\mathcal{T}_\lambda(X_i), Y_i)$ . By the second inequalities in Theorem E.1 and E.2,

$$\mathbb{P} \left( \sup_{\lambda \in \Lambda} \frac{R(\mathcal{T}_\lambda) - \hat{R}(\mathcal{T}_\lambda)}{\sqrt{R(\mathcal{T}_\lambda) + \eta}} \leq t(\eta; \delta) \right) \geq 1 - \delta.$$

On this event, for any  $\lambda \in \Lambda$ ,

$$\begin{aligned} R(\mathcal{T}_\lambda) - t(\eta; \delta) \sqrt{R(\mathcal{T}_\lambda) + \eta} &\leq \hat{R}(\mathcal{T}_\lambda) \\ \iff \left( \sqrt{R(\mathcal{T}_\lambda) + \eta} - \frac{t(\eta; \delta)}{2} \right)^2 &\leq \hat{R}(\mathcal{T}_\lambda) + \eta + \frac{t(\eta; \delta)^2}{4} \\ \iff \sqrt{R(\mathcal{T}_\lambda) + \eta} &\leq \sqrt{\hat{R}(\mathcal{T}_\lambda) + \eta + \frac{t(\eta; \delta)^2}{4}} + \frac{t(\eta; \delta)}{2} \\ \implies R(\mathcal{T}_\lambda) &\leq \hat{R}(\mathcal{T}_\lambda) + t(\eta; \delta) \sqrt{\hat{R}(\mathcal{T}_\lambda) + \eta + \frac{t(\eta; \delta)^2}{4}} + \frac{t(\eta; \delta)^2}{2}. \end{aligned}$$

□

**Proposition E.1.** [Section 2.7 of [81]] Let  $Z_1, \dots, Z_n \in [0, 1]$  be i.i.d. random variables with  $\mathbb{E}[Z_1] = \mu$  and  $\text{Var}[Z_1] = \sigma^2$ . Further let  $\hat{\mu} = (1/n) \sum_{i=1}^n Z_i$ . Then for any  $x > 0$ ,

$$\mathbb{P}(\hat{\mu} - \mu \geq x) \leq \exp \left\{ -\frac{nx^2/2}{\sigma^2 + x/3} \right\}.$$

**Proposition E.2.** [[77], Theorem 2.4] Let  $x_1, \dots, x_N$  be a fixed finite population of  $N > 1$  real points with  $x_i \in [0, 1]$  and  $\bar{x} = (1/N) \sum_{i=1}^N x_i$ . Further let  $\Pi$  be a random permutation of  $\{1, \dots, N\}$ . Then for any  $\epsilon > 0$ ,

$$\mathbb{P} \left( \frac{1}{n} \sum_{k=1}^n x_{\pi(k)} - \bar{x} \geq \epsilon \right) \leq \exp \left\{ -\frac{2n\epsilon^2}{(1 - n/N)(1 + 1/n)} \right\}.$$



**Proposition E.3.** [[77], Proposition 1.4] With the same setting as Proposition E.2, for any  $\epsilon > 0$ ,

$$\mathbb{P}\left(\frac{1}{n}\sum_{k=1}^n x_{\pi(k)} - \bar{x} \geq \epsilon\right) \leq \exp\left\{-\frac{n\epsilon^2/2}{\sigma^2 + \epsilon/3}\right\},$$

where

$$\sigma^2 = \frac{1}{n}\sum_{k=1}^n (x_k - \bar{x})^2.$$

**Proposition E.4.** [[79], Theorem 1.1] Let  $\epsilon_1, \dots, \epsilon_n$  be i.i.d. Rademacher random variables and  $a = (a_1, \dots, a_n)$  be a vector with  $\|a\|_2 \leq 1$ . Then

$$\mathbb{P}\left(\sum_{i=1}^n a_i \epsilon_i \geq x\right) \leq c_1(1 - \Phi(x))$$

where  $\Phi(x)$  is the CDF of the standard normal distribution and  $c_1 = 1/4(1 - \Phi(\sqrt{2})) \approx 3.178$ .

**Proposition E.5.** [[78], Theorem 1.1] With the same assumptions and notation as in Proposition E.4,

$$\mathbb{P}\left(\sum_{i=1}^n a_i \epsilon_i \geq x\right) \leq 1 - \Phi(x) + \frac{c_2}{9 + x^2} \exp\left\{-\frac{x^2}{2}\right\},$$

where  $c_2 = 5\sqrt{e}(2\Phi(1) - 1) \approx 5.628$ .

**Proof of Theorem E.1.** Let  $Z_{n+1}, \dots, Z_{n+n'}$  be i.i.d. fresh samples drawn from the same distribution as  $Z_1$  and

$$\hat{s}_{n'}(\lambda) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} S(\lambda; Z_i).$$

Since the proof is quite involved, we decompose it into four steps.

**Step 1 for (8):** we shall prove that

$$\mathbb{P}\left(\sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_n(\lambda) - \hat{s}_{n'}(\lambda)}{\sqrt{\hat{s}_{n+n'}(\lambda) + \eta}} \geq (1 - \gamma)t\right) \geq \mathbb{P}\left(\sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_n(\lambda) - s(\lambda)}{\sqrt{\hat{s}_n(\lambda) + \eta}} \geq t\right) \inf_{\lambda \in \mathbb{R}} \mathbb{P}\left(\frac{\hat{s}_{n'}(\lambda) - s(\lambda)}{\sqrt{s(\lambda) + \eta}} \leq \gamma t\right). \quad (11)$$

Consider the event that there exists  $\lambda^* \in \mathbb{R}$  such that

$$\frac{\hat{s}_n(\lambda^*) - s(\lambda^*)}{\sqrt{\hat{s}_n(\lambda^*) + \eta}} \geq t, \quad \frac{\hat{s}_{n'}(\lambda^*) - s(\lambda^*)}{\sqrt{s(\lambda^*) + \kappa^+}} \leq \gamma t. \quad (12)$$

The first inequality of (12) implies that

$$(\hat{s}_n(\lambda^*) + \eta) - t\sqrt{\hat{s}_n(\lambda^*) + \eta} \geq \eta + s(\lambda^*) \implies \left(\sqrt{\hat{s}_n(\lambda^*) + \eta} - \frac{t}{2}\right)^2 \geq \frac{t^2}{4} + \eta + s(\lambda^*).$$

Since  $\hat{s}_n(\lambda) + \eta > 0$  and  $\sqrt{t^2/4 + \eta} > t/2$ , we have

$$\begin{aligned} \hat{s}_n(\lambda^*) + \eta &\geq \left(\frac{t}{2} + \sqrt{\frac{t^2}{4} + \eta + s(\lambda^*)}\right)^2 = \frac{t^2}{2} + \eta + s(\lambda^*) + t\sqrt{\frac{t^2}{4} + \eta + s(\lambda^*)} \\ &\geq s(\lambda^*) + \eta + \frac{t^2}{2} + t\sqrt{\frac{t^2}{4} + \eta} = s(\lambda^*) + \kappa^+. \end{aligned} \quad (13)$$

The second inequality of (12) and (13) imply that

$$\frac{\hat{s}_{n'}(\lambda^*) - s(\lambda^*)}{\sqrt{s(\lambda^*) + \kappa^+}} \leq \gamma t \leq t \leq \frac{\hat{s}_n(\lambda^*) - s(\lambda^*)}{\sqrt{\hat{s}_n(\lambda^*) + \eta}} \leq \frac{\hat{s}_n(\lambda^*) - s(\lambda^*)}{\sqrt{s(\lambda^*) + \kappa^+}} \implies \hat{s}_n(\lambda^*) \geq \hat{s}_{n'}(\lambda^*). \quad (14)$$

As a result,

$$\hat{s}_{n+n'}(\lambda) = \frac{n'}{n+n'}\hat{s}_n(\lambda) + \frac{n}{n+n'}\hat{s}_{n'}(\lambda) \leq \hat{s}_n(\lambda) \quad (15)$$

By (12) - (15),

$$\begin{aligned} \frac{\hat{s}_n(\lambda^*) - \hat{s}_{n'}(\lambda^*)}{\sqrt{\hat{s}_{n+n'}(\lambda^*) + \eta}} &\stackrel{(14) \text{ and } (15)}{\geq} \frac{\hat{s}_n(\lambda^*) - \hat{s}_{n'}(\lambda^*)}{\sqrt{\hat{s}_n(\lambda^*) + \eta}} \\ &\stackrel{(12)}{\geq} \frac{t\sqrt{\hat{s}_n(\lambda^*) + \eta} - (\hat{s}_{n'}(\lambda^*) - s(\lambda^*))}{\sqrt{\hat{s}_n(\lambda^*) + \eta}} \\ &\stackrel{(12)}{\geq} \frac{t\sqrt{\hat{s}_n(\lambda^*) + \eta} - \gamma t\sqrt{s(\lambda^*) + \kappa^+}}{\sqrt{\hat{s}_n(\lambda^*) + \eta}} \\ &\stackrel{(13)}{\geq} \frac{t\sqrt{\hat{s}_n(\lambda^*) + \eta} - \gamma t\sqrt{\hat{s}_n(\lambda^*) + \eta}}{\sqrt{\hat{s}_n(\lambda^*) + \eta}} \\ &= (1 - \gamma)t. \end{aligned}$$

Given  $(Z_1, \dots, Z_n)$ , if  $\lambda \mapsto \frac{\hat{s}_n(\lambda) - s(\lambda)}{\sqrt{\hat{s}_n(\lambda) + \eta}}$  achieves the supremum, we take  $\lambda^*$  as the maximizer. Then  $\lambda^*$  is measurable with respect to  $\{Z_1, \dots, Z_n\}$  and independent of  $\hat{s}_{n'}$ . As a result,

$$\begin{aligned} &\mathbb{P}\left(\sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_n(\lambda) - \hat{s}_{n'}(\lambda)}{\sqrt{\hat{s}_{n+n'}(\lambda) + \eta}} \geq (1 - \gamma)t\right) \\ &\geq \mathbb{P}\left(\frac{\hat{s}_n(\lambda^*) - \hat{s}_{n'}(\lambda^*)}{\sqrt{\hat{s}_{n+n'}(\lambda^*) + \eta}} \geq (1 - \gamma)t\right) \\ &\geq \mathbb{P}\left(\frac{\hat{s}_n(\lambda^*) - s(\lambda^*)}{\sqrt{\hat{s}_n(\lambda^*) + \eta}} \geq t, \frac{\hat{s}_{n'}(\lambda^*) - s(\lambda^*)}{\sqrt{s(\lambda^*) + \kappa^+}} \leq \gamma t\right) \\ &= \mathbb{E}\left[\mathbb{P}\left(\frac{\hat{s}_n(\lambda^*) - s(\lambda^*)}{\sqrt{\hat{s}_n(\lambda^*) + \eta}} \geq t, \frac{\hat{s}_{n'}(\lambda^*) - s(\lambda^*)}{\sqrt{s(\lambda^*) + \kappa^+}} \leq \gamma t \mid Z_1, \dots, Z_n\right)\right] \\ &= \mathbb{E}\left[\mathbb{1}\left\{\frac{\hat{s}_n(\lambda^*) - s(\lambda^*)}{\sqrt{\hat{s}_n(\lambda^*) + \eta}} \geq t\right\} \mathbb{P}\left(\frac{\hat{s}_{n'}(\lambda^*) - s(\lambda^*)}{\sqrt{s(\lambda^*) + \kappa^+}} \leq \gamma t \mid Z_1, \dots, Z_n\right)\right] \\ &\geq \mathbb{P}\left(\frac{\hat{s}_n(\lambda^*) - s(\lambda^*)}{\sqrt{\hat{s}_n(\lambda^*) + \eta}} \geq t\right) \inf_{\lambda \in \mathbb{R}} \mathbb{P}\left(\frac{\hat{s}_{n'}(\lambda) - s(\lambda)}{\sqrt{s(\lambda) + \kappa^+}} \leq \gamma t\right) \\ &= \mathbb{P}\left(\sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_n(\lambda) - s(\lambda)}{\sqrt{\hat{s}_n(\lambda) + \eta}} \geq t\right) \inf_{\lambda \in \mathbb{R}} \mathbb{P}\left(\frac{\hat{s}_{n'}(\lambda) - s(\lambda)}{\sqrt{s(\lambda) + \kappa^+}} \leq \gamma t\right). \end{aligned}$$

This concludes (11). If the supremum of  $\lambda \mapsto \frac{\hat{s}_n(\lambda) - s(\lambda)}{\sqrt{\hat{s}_n(\lambda) + \eta}}$  cannot be achieved, we can find a sequence of events  $\{\lambda_\ell : \ell = 1, 2, \dots\}$  at which the values converge to the supremum. For each  $\lambda_\ell$ , we can prove the above inequality and (11) can be proved by taking  $\ell \rightarrow \infty$ .

**Step 2 for (8):** we shall prove that

$$\inf_{\lambda \in \mathbb{R}} \mathbb{P}\left(\frac{\hat{s}_{n'}(\lambda) - s(\lambda)}{\sqrt{s(\lambda) + \kappa^+}} \leq \gamma t\right) \geq 1 - \exp\{-g_1(t; n', \gamma, \kappa^+)\}. \quad (16)$$

Given any subset  $\lambda \in \mathbb{R}$ ,

$$\hat{s}_{n'}(\lambda) - s(\lambda) = \frac{1}{n'} \sum_{i=n+1}^{n+n'} (S(\lambda; Z_i) - s(\lambda)),$$

and

$$\mathbb{E}[S(\lambda; Z_i) - s(\lambda)] = 0, \quad \mathbb{E}[(S(\lambda; Z_i) - s(\lambda))^2] \leq \mathbb{E}[S(\lambda; Z_i)^2] \leq \mathbb{E}[S(\lambda; Z_i)] = s(\lambda).$$

By the Bernstein inequality (Proposition E.1),

$$\mathbb{P}\left(\hat{s}_{n'}(\lambda) - s(\lambda) \geq \gamma t \sqrt{s(\lambda) + \kappa^+}\right) \leq \exp\left\{-\frac{n't^2}{2} \gamma^2 \frac{s(\lambda) + \kappa^+}{s(\lambda) + \gamma t \sqrt{s(\lambda) + \kappa^+}/3}\right\}.$$

It remains to prove that

$$\frac{s(\lambda) + \kappa^+}{s(\lambda) + \gamma t \sqrt{s(\lambda) + \kappa^+}/3} \geq \left(1 + \frac{\gamma^2 t^2}{36\kappa^+}\right)^{-1}. \quad (17)$$

Using the fact that  $\sqrt{a} \leq (ba + 1/b)/2$ , we have

$$\sqrt{s(\lambda) + \kappa^+} \leq \frac{\gamma t}{12\kappa^+} (s(\lambda) + \kappa^+) + \frac{3\kappa^+}{\gamma t}.$$

This entails that

$$\inf_{\lambda \in \mathbb{R}} \mathbb{P}\left(\frac{\hat{s}_{n'}(\lambda) - s(\lambda)}{\sqrt{s(\lambda) + \kappa^+}} \leq \gamma t\right) \geq 1 - \exp\{-g_{11}(t; n', \gamma, \kappa^+)\},$$

where

$$g_{11}(t; n', \gamma, \kappa^+) = \frac{n't^2}{2} \frac{\gamma^2}{1 + \gamma^2 t^2 / 36\kappa^+}.$$

On the other hand, since  $\mathbb{E}[\hat{s}_{n'}(\lambda)] = s(\lambda)$ , by Chebyshev's inequality,

$$\mathbb{P}\left(\frac{\hat{s}_{n'}(\lambda) - s(\lambda)}{\sqrt{s(\lambda) + \kappa^+}} \geq \gamma t\right) \leq \frac{1}{\gamma^2 t^2} \text{Var}\left(\frac{\hat{s}_{n'}(\lambda) - s(\lambda)}{\sqrt{s(\lambda) + \kappa^+}}\right) = \frac{1}{n'\gamma^2 t^2} \frac{s(\lambda)(1 - s(\lambda))}{s(\lambda) + \kappa^+}.$$

Let  $m(x) = x(1 - x)/(x + \kappa^+)$ . Then

$$\frac{d}{dx} \log[m(x)] = \frac{1}{x} - \frac{1}{1 - x} - \frac{1}{x + \kappa^+} = \frac{\kappa^+ - 2\kappa^+ x - x^2}{x(x + \kappa^+)(1 - x)}.$$

Since  $\eta \geq 0$ ,  $\kappa^+ \geq 0$ . Via some tedious algebra, we can show that  $m(x)$  achieves its maximum at  $x^* = \sqrt{\kappa^+ + \kappa^{+2}} - \kappa^+$  at which

$$m(x^*) = (\sqrt{1 + \kappa^+} - \sqrt{\kappa^+})^2.$$

Therefore,

$$\mathbb{P}\left(\frac{\hat{s}_{n'}(\lambda) - s(\lambda)}{\sqrt{s(\lambda) + \kappa^+}} \leq \gamma t\right) \geq 1 - \exp\{-g_{12}(t; n', \gamma, \kappa^+)\},$$

where

$$g_{12}(t; n', \gamma, \kappa^+) = \log\left(n'\gamma^2 t^2 / (\sqrt{1 + \kappa^+} - \sqrt{\kappa^+})^2\right).$$

Putting two pieces together, (16) is proved by noting that  $g_1 = g_{11} \wedge g_{12}$ .

**Step 3 for (8):** we shall prove that

$$\mathbb{P}\left(\sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_n(\lambda) - \hat{s}_{n'}(\lambda)}{\sqrt{\hat{s}_{n+n'}(\lambda) + \eta}} \geq (1 - \gamma)t\right) \leq \Delta(n + n') \exp\{-g_2(t; n, n', \gamma, \eta)\}. \quad (18)$$

Let  $\Pi$  be any given permutation over  $\{1, \dots, n + n'\}$ . Since  $Z_1, \dots, Z_{n+n'}$  are i.i.d.,

$$(Z_1, \dots, Z_{n+n'}) \stackrel{d}{=} (Z_{\Pi(1)}, \dots, Z_{\Pi(n+n')}).$$

As a result, for any  $\Pi$ ,

$$\mathbb{P}\left(\sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_n(\lambda) - \hat{s}_{n'}(\lambda)}{\sqrt{\hat{s}_{n+n'}(\lambda) + \eta}} \geq (1 - \gamma)t\right) = \mathbb{P}\left(\sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_{n,\Pi}(\lambda) - \hat{s}_{n',\Pi}(\lambda)}{\sqrt{\hat{s}_{n+n'}(\lambda) + \eta}} \geq (1 - \gamma)t\right)$$

where  $\hat{s}_{n,\Pi}(\lambda)$  is the empirical CDF of  $Z_{\Pi(1)}, \dots, Z_{\Pi(n)}$  and  $\hat{s}_{n',\Pi}(\lambda)$  is the empirical CDF of  $Z_{\Pi(n+1)}, \dots, Z_{\Pi(n+n')}$ . Note that  $\hat{s}_{n+n'}(\lambda)$  is invariant with respect to  $\Pi$ . With a slight abuse of notation, we take  $\Pi$  as a uniform permutation over  $\{1, \dots, n+n'\}$ . Then

$$\mathbb{P} \left( \sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_n(\lambda) - \hat{s}_{n'}(\lambda)}{\sqrt{\hat{s}_{n+n'}(\lambda) + \eta}} \geq (1-\gamma)t \right) = \mathbb{E}_{\hat{s}_{n+n'}} \left[ \mathbb{P}_{\Pi} \left( \sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_{n,\Pi}(\lambda) - \hat{s}_{n',\Pi}(\lambda)}{\sqrt{\hat{s}_{n+n'}(\lambda) + \eta}} \geq (1-\gamma)t \mid \hat{s}_{n+n'} \right) \right].$$

Let  $\Gamma(n+n')$  be the collection of distinct sets in the form of

$$\{S(\lambda; Z_1), \dots, S(\lambda; Z_{n+n'}) : \lambda \in \mathbb{R}\}$$

It is easy to see that  $|\Gamma(n+n')| \leq \Delta(n+n')$ . Then

$$\begin{aligned} & \mathbb{P}_{\Pi} \left( \sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_{n,\Pi}(\lambda) - \hat{s}_{n',\Pi}(\lambda)}{\sqrt{\hat{s}_{n+n'}(\lambda) + \eta}} \geq (1-\gamma)t \mid \hat{s}_{n+n'} \right) \\ & \leq \sum_{\lambda \in \Gamma(n+n')} \mathbb{P}_{\Pi} \left( \frac{\hat{s}_{n,\Pi}(\lambda) - \hat{s}_{n',\Pi}(\lambda)}{\sqrt{\hat{s}_{n+n'}(\lambda) + \eta}} \geq (1-\gamma)t \mid \hat{s}_{n+n'} \right). \end{aligned}$$

It remains to prove that

$$\mathbb{P}_{\Pi} \left( \frac{\hat{s}_{n,\Pi}(\lambda) - \hat{s}_{n',\Pi}(\lambda)}{\sqrt{\hat{s}_{n+n'}(\lambda) + \eta}} \geq (1-\gamma)t \mid \hat{s}_{n+n'} \right) \leq \exp\{-g_2(t; n, n', \gamma, \eta)\} \quad \text{a.s..} \quad (19)$$

By definition,

$$\hat{s}_{n,\Pi}(\lambda) - \hat{s}_{n',\Pi}(\lambda) = \hat{s}_{n,\Pi}(\lambda) - \frac{1}{n'} ((n+n')\hat{s}_{n+n'}(\lambda) - n\hat{s}_{n,\Pi}(\lambda)) = \frac{n+n'}{n'} (\hat{s}_{n,\Pi}(\lambda) - \hat{s}_{n+n'}(\lambda)).$$

Note that  $n\hat{s}_n(\lambda)$  is the sum of  $n$  elements from the unordered set  $\{S(\lambda; Z_1), \dots, S(\lambda; Z_{n+n'})\}$  sampled without replacement. By Proposition E.3,

$$\begin{aligned} & \mathbb{P} \left( \hat{s}_n(\lambda) - \hat{s}_{n+n'}(\lambda) \geq \frac{n'}{n+n'} (1-\gamma)t \sqrt{\hat{s}_{n+n'}(\lambda) + \eta} \mid \hat{s}_{n+n'} \right) \\ & \leq \exp \left\{ -\frac{nt^2}{2} \left( \frac{n'}{n+n'} \right)^2 (1-\gamma)^2 \frac{\hat{s}_{n+n'}(\lambda) + \eta}{\hat{s}_{n+n'}(\lambda) + (1-\gamma)t \frac{n'}{n+n'} \sqrt{\hat{s}_{n+n'}(\lambda) + \eta/3}} \right\} \\ & \leq \exp \left\{ -\frac{nt^2}{2} \left( \frac{n'}{n+n'} \right)^2 (1-\gamma)^2 \frac{\hat{s}_{n+n'}(\lambda) + \eta}{\hat{s}_{n+n'}(\lambda) + (1-\gamma)t \sqrt{\hat{s}_{n+n'}(\lambda) + \eta/3}} \right\} \end{aligned}$$

Using the same argument as (17), we can prove that

$$\frac{\hat{s}_{n+n'}(\lambda) + \eta}{\hat{s}_{n+n'}(\lambda) + (1-\gamma)t \sqrt{\hat{s}_{n+n'}(\lambda) + \eta/3}} \geq \left( 1 + \frac{(1-\gamma)^2 t^2}{36\eta} \right)^{-1}.$$

Therefore,

$$\mathbb{P} \left( \hat{s}_n(\lambda) - \hat{s}_{n+n'}(\lambda) \geq \frac{n'}{n+n'} t \sqrt{\hat{s}_{n+n'}(\lambda) + \eta} \mid \hat{s}_{n+n'} \right) \leq \exp\{-g_2(t; n, n', \gamma, \eta)\}.$$

Since the bound is independent of  $\hat{s}_{n+n'}$ , (19) is proved and thus step 3.

**Step 4 for (8):** putting (11), (16) and (18) together, we prove that for any  $\gamma \in (0, 1)$  and  $n' \in \mathbb{Z}^+$ ,

$$\mathbb{P} \left( \sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_n(\lambda) - s(\lambda)}{\sqrt{\hat{s}_n(\lambda) + \eta}} \geq (1-\gamma)t \right) \leq \frac{\Delta(n+n') \exp\{-g_2(t; n, n', \gamma, \eta)\}}{1 - \exp\{-g_1(t; n', \gamma, \kappa^+)\}}.$$

Since the right-handed side is deterministic, we can take infimum over  $\gamma$  and  $n'$ , which yields (8).

To prove (9), we follow the same steps as above.

**Step 1 for (9):** we shall prove that

$$\mathbb{P} \left( \sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_{n'}(\lambda) - \hat{s}_n(\lambda)}{\sqrt{\hat{s}_{n+n'}(\lambda) + \kappa^-}} \geq (1 - \gamma)t \right) \geq \mathbb{P} \left( \sup_{\lambda \in \mathbb{R}} \frac{s(\lambda) - \hat{s}_n(\lambda)}{\sqrt{s(\lambda) + \eta}} \geq t \right) \inf_{\lambda \in \mathbb{R}} \mathbb{P} \left( \frac{s(\lambda) - \hat{s}_{n'}(\lambda)}{\sqrt{s(\lambda) + \eta}} \leq \gamma t \right). \quad (20)$$

Consider the event that there exists  $\lambda^* \in \mathbb{R}$  such that

$$\frac{s(\lambda^*) - \hat{s}_n(\lambda^*)}{\sqrt{s(\lambda^*) + \eta}} \geq t, \quad \frac{s(\lambda^*) - \hat{s}_{n'}(\lambda^*)}{\sqrt{s(\lambda^*) + \eta}} \leq \gamma t$$

As with (21), we can show that

$$s(\lambda^*) + \eta \geq \hat{s}_n(\lambda^*) + \kappa^+. \quad (21)$$

On the other hand, by (20),

$$\hat{s}_{n'}(\lambda^*) \geq s(\lambda^*) - \gamma t \sqrt{s(\lambda^*) + \eta} \geq s(\lambda^*) - t \sqrt{s(\lambda^*) + \eta} \geq \hat{s}_n(\lambda^*). \quad (22)$$

Let  $a, b, c$  be arbitrary positive numbers and  $d(x) = (x - a)/\sqrt{bx + c}$ . Then

$$\frac{d}{dx} \log d(x) = \frac{1}{x - a} - \frac{b}{2(bx + c)} = \frac{bx + 2c + ab}{2(x - a)(bx + c)}.$$

Thus,  $d(x)$  is increasing on  $[a, \infty)$ . Take  $a = \hat{s}_n(\lambda)$ ,  $b = n'/(n + n')$ , and  $c = na/(n + n') + \kappa^-$ . By (22),

$$\begin{aligned} \frac{\hat{s}_{n'}(\lambda^*) - \hat{s}_n(\lambda^*)}{\sqrt{\hat{s}_{n+n'}(\lambda^*) + \kappa^-}} &= d(\hat{s}_{n'}(\lambda^*)) \geq d\left(s(\lambda^*) - \gamma t \sqrt{s(\lambda^*) + \eta}\right) \\ &= \frac{s(\lambda^*) - \gamma t \sqrt{s(\lambda^*) + \eta} - \hat{s}_n(\lambda^*)}{\sqrt{\frac{n'}{n+n'}(s(\lambda^*) - \gamma t \sqrt{s(\lambda^*) + \eta}) + \frac{n}{n+n'}\hat{s}_n(\lambda^*) + \kappa^-}} \\ &\stackrel{(21)}{\geq} \frac{s(\lambda^*) - \gamma t \sqrt{s(\lambda^*) + \eta} - \hat{s}_n(\lambda^*)}{\sqrt{\frac{n'}{n+n'}(s(\lambda^*) - \gamma t \sqrt{s(\lambda^*) + \eta}) + \frac{n}{n+n'}(s(\lambda^*) - t \sqrt{s(\lambda^*) + \eta}) + \kappa^-}} \\ &= \frac{s(\lambda^*) - \gamma t \sqrt{s(\lambda^*) + \eta} - \hat{s}_n(\lambda^*)}{\sqrt{s(\lambda^*) - \frac{n+\gamma n'}{n+n'}\sqrt{s(\lambda^*) + \eta} + \kappa^-}} \\ &\stackrel{(21)}{\geq} \frac{s(\lambda^*) - \gamma t \sqrt{s(\lambda^*) + \eta} - \hat{s}_n(\lambda^*)}{\sqrt{s(\lambda^*) - \frac{n+\gamma n'}{n+n'}\sqrt{\kappa^+} + \kappa^-}} \\ &= \frac{s(\lambda^*) - \gamma t \sqrt{s(\lambda^*) + \eta} - \hat{s}_n(\lambda^*)}{\sqrt{s(\lambda^*) + \eta}} \\ &\stackrel{(20)}{\geq} (1 - \gamma)t. \end{aligned}$$

Similar to step 1 for (9), we complete the proof of (20).

**Step 2 for (9):** using exactly the same proof of (23), we can prove that

$$\inf_{\lambda \in \mathbb{R}} \mathbb{P} \left( \frac{s(\lambda) - \hat{s}_{n'}(\lambda)}{\sqrt{s(\lambda) + \eta}} \leq \gamma t \right) \geq 1 - \exp\{-g_1(t; n', \gamma, \eta)\}. \quad (23)$$

**Step 3 for (9):** using exactly the same proof of (24), we can prove that

$$\mathbb{P} \left( \sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_{n'}(\lambda) - \hat{s}_n(\lambda)}{\sqrt{\hat{s}_{n+n'}(\lambda) + \kappa^-}} \geq (1 - \gamma)t \right) \leq \Delta(n + n') \exp\{g_2(t; n, n', \gamma, \kappa^-)\}. \quad (24)$$

**Step 4 for (9):** putting (20), (23) and (24) together, we prove that for any  $\gamma \in (0, 1)$  and  $n' \in \mathbb{Z}^+$ ,

$$\mathbb{P} \left( \sup_{\lambda \in \mathbb{R}} \frac{s(\lambda) - \hat{s}_n(\lambda)}{\sqrt{s(\lambda) + \eta}} \geq t \right) \leq \frac{\Delta(n + n') \exp\{-g_2(t; n, n', \gamma, \kappa^-)\}}{1 - \exp\{-g_1(t; n', \gamma, \eta)\}}.$$

Since the right-handed side is deterministic, we can take infimum over  $\gamma$  and  $n'$ , which yields (9).  $\square$

**Proof of Theorem E.2.** We prove (10) first. We will use the same notation hereafter as in the proof of Theorem E.1. By (20) and (23) with  $n' = n$ , it remains to prove that

$$\mathbb{P} \left( \sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_n(\lambda) - \hat{s}'_n(\lambda)}{\sqrt{\hat{s}_{2n}(\lambda) + \eta}} \geq (1 - \gamma)t \right) \leq \Delta(2n)\tilde{g} \left( \sqrt{\frac{n(1 + \eta)}{2}}(1 - \gamma)t \right), \quad (25)$$

where  $\hat{s}'_n = \hat{s}_{n'}$  is the empirical CDF of  $Z_{n+1}, \dots, Z_{2n}$ . Instead of conditioning on the unordered set of  $\{Z_1, \dots, Z_{2n}\}$ , we condition on a more refined statistic

$$\mathcal{Z}_{\text{swap}} \triangleq (\{Z_1, Z_{n+1}\}, \{Z_2, Z_{n+2}\}, \dots, \{Z_n, Z_{2n}\}).$$

Note that  $\mathcal{Z}_{\text{swap}}$  is a function of the unordered set of  $\{Z_1, \dots, Z_{2n}\}$ , the number of distinct  $\mathcal{Z}_{\text{swap}}$  over  $\lambda \in \mathbb{R}$  is at most  $\Delta(2n)$ . Thus, by a union bound,

$$\mathbb{P} \left( \sup_{\lambda \in \mathbb{R}} \frac{\hat{s}_n(\lambda) - \hat{s}'_n(\lambda)}{\sqrt{\hat{s}_{2n}(\lambda) + \eta}} \geq (1 - \gamma)t \right) \leq \Delta(2n) \sup_{\lambda \in \mathbb{R}} \mathbb{P} \left( \frac{\hat{s}_n(\lambda) - \hat{s}'_n(\lambda)}{\sqrt{\hat{s}_{2n}(\lambda) + \eta}} \geq (1 - \gamma)t \right).$$

Conditional on  $\mathcal{Z}_{\text{swap}}$ ,  $\hat{s}_{2n}$  is deterministic and for any event  $A$ ,

$$(S(\lambda; Z_i) - S(\lambda; Z_{n+i}))_{i=1}^n \stackrel{d}{=} (S(\lambda; Z_{n+i}) - S(\lambda; Z_i))_{i=1}^n$$

where  $\epsilon_1, \dots, \epsilon_n$  are i.i.d. Rademacher random variables, i.e.  $\mathbb{P}(\epsilon_i = \pm 1) = 1/2$ , and are independent of  $\mathcal{Z}_{\text{swap}}$ . As a result,

$$\begin{aligned} & \mathbb{P} \left( \frac{\hat{s}_n(\lambda) - \hat{s}'_n(\lambda)}{\sqrt{\hat{s}_{2n}(\lambda) + \eta}} \geq (1 - \gamma)t \mid \mathcal{Z}_{\text{swap}} \right) \\ &= \mathbb{P} \left( \frac{\sum_{i=1}^n (S(\lambda; Z_i) - S(\lambda; Z_{n+i}))\epsilon_i}{\sqrt{\sum_{i=1}^n (S(\lambda; Z_i) + S(\lambda; Z_{n+i})) + 2n\eta}} \geq \sqrt{\frac{n}{2}}(1 - \gamma)t \mid \mathcal{Z}_{\text{swap}} \right) \\ &= \mathbb{P} \left( \frac{\sum_{i=1}^n \sqrt{1 + \eta}(S(\lambda; Z_i) - S(\lambda; Z_{n+i}))\epsilon_i}{\sqrt{\sum_{i=1}^n (S(\lambda; Z_i) + S(\lambda; Z_{n+i})) + 2n\eta}} \geq \sqrt{\frac{n(1 + \eta)}{2}}(1 - \gamma)t \mid \mathcal{Z}_{\text{swap}} \right). \end{aligned} \quad (26)$$

Let

$$a_i = \frac{\sqrt{1 + \eta}(S(\lambda; Z_i) - S(\lambda; Z_{n+i}))}{\sqrt{\sum_{i=1}^n (S(\lambda; Z_i) + S(\lambda; Z_{n+i})) + 2n\eta}}.$$

Then  $a_i$  is deterministic conditional on  $\mathcal{Z}_{\text{swap}}$  and

$$\begin{aligned} \sum_{i=1}^n a_i^2 &= \frac{(1 + \eta) \sum_{i=1}^n (S(\lambda; Z_i) - S(\lambda; Z_{n+i}))^2}{\sum_{i=1}^n (S(\lambda; Z_i) + S(\lambda; Z_{n+i})) + 2n\eta} \\ &\stackrel{(i)}{\leq} \frac{(1 + \eta) \sum_{i=1}^n (S(\lambda; Z_i) + S(\lambda; Z_{n+i}))}{\sum_{i=1}^n (S(\lambda; Z_i) + S(\lambda; Z_{n+i})) + 2n\eta} \\ &\stackrel{(ii)}{\leq} 1, \end{aligned}$$

where (i) uses the fact that  $(S(\lambda; Z_i) - S(\lambda; Z_{n+i}))^2 \leq S(\lambda; Z_i)^2 + S(\lambda; Z_{n+i})^2 = S(\lambda; Z_i) + S(\lambda; Z_{n+i})$ , and (ii) uses the fact that  $\sum_{i=1}^n (S(\lambda; Z_i) + S(\lambda; Z_{n+i})) \leq 2n$ . Then (26) implies that

$$\mathbb{P} \left( \frac{\hat{s}_n(\lambda) - \hat{s}'_n(\lambda)}{\sqrt{\hat{s}_{2n}(\lambda) + \eta}} \geq (1 - \gamma)t \mid \mathcal{Z}_{\text{swap}} \right) \leq \sup_{\|a\|_2^2 \leq 1} \mathbb{P} \left( \sum_{i=1}^n a_i \epsilon_i \geq \sqrt{\frac{n(1 + \eta)}{2}}(1 - \gamma)t \right).$$

By the Bentkus-Dzindzalieta inequality (Proposition E.4),

$$\mathbb{P}\left(\sum_{i=1}^n a_i \epsilon_i \geq \sqrt{\frac{n(1+\eta)}{2}}(1-\gamma)t\right) \leq \tilde{g}_1\left(\sqrt{\frac{n(1+\eta)}{2}}(1-\gamma)t\right).$$

By the Pinelis inequality (Proposition E.5),

$$\mathbb{P}\left(\sum_{i=1}^n a_i \epsilon_i \geq \sqrt{\frac{n(1+\eta)}{2}}(1-\gamma)t\right) \leq \tilde{g}_2\left(\sqrt{\frac{n(1+\eta)}{2}}(1-\gamma)t\right).$$

Finally, since  $\epsilon_i$  is subgaussian with parameter 1 and  $\|a\|_2^2 = 1$ ,  $\sum_{i=1}^n a_i \epsilon_i$  is also subgaussian with parameter 1. By Hoeffding's inequality, we have

$$\mathbb{P}\left(\sum_{i=1}^n a_i \epsilon_i \geq \sqrt{\frac{n(1+\eta)}{2}}(1-\gamma)t\right) \leq \tilde{g}_3\left(\sqrt{\frac{n(1+\eta)}{2}}(1-\gamma)t\right).$$

Putting the pieces together, (10) is proved.

Similarly, to prove (10), it remains to prove

$$\mathbb{P}\left(\sup_{\lambda \in \mathbb{R}} \frac{\hat{s}'_n(\lambda) - \hat{s}_n(\lambda)}{\sqrt{\hat{s}_{2n}(\lambda) + \eta}} \geq (1-\gamma)t\right) \leq \Delta(2n)\tilde{g}\left(\sqrt{\frac{n(1+\eta)}{2}}(1-\gamma)t\right).$$

Since  $\hat{s}_n$  and  $\hat{s}'_n$  are symmetric, it is implied by (25). Thus, the proof of (10) is also completed.  $\square$