# Powerful batch conformal prediction for classification

**Ulysse Gazin**
Université Paris Cité

**Ruth Heller**
Tel Aviv University

**Étienne Roquain**
Sorbonne Université

**Aldo Solari**
Ca' Foscari University of Venice

## Abstract

In a split conformal framework with $K$ classes, a calibration sample of $n$ labeled examples is observed for inference on the label of a new unlabeled example. We explore the setting where a 'batch' of $m$ independent such unlabeled examples is given, and the goal is to construct a batch prediction set with 1-$\alpha$ coverage. Unlike individual prediction sets, the batch prediction set is a collection of label vectors of size $m$, while the calibration sample consists of univariate labels. A natural approach is to apply the Bonferroni correction, which concatenates individual prediction sets at level $1-\alpha/m$. We propose a uniformly more powerful solution, based on specific combinations of conformal $p$-values that exploit the Simes inequality. We provide a general recipe for valid inference with any combinations of conformal $p$-values, and compare the performance of several useful choices. Intuitively, the pooled evidence of relatively 'easy' examples within the batch can help provide narrower batch prediction sets. Additionally, we introduce a more computationally intensive method that aggregates batch scores and can be even more powerful. The theoretical guarantees are established when all examples are independent and identically distributed (iid), as well as more generally when iid is assumed only conditionally within each class. Notably, our results remain valid under label distribution shift, since the distribution of the labels need not be the same in the calibration sample and in the new batch. The effectiveness of the methods is highlighted through illustrative synthetic and real data examples.

Keywords: conformal inference, multiple testing, label distribution shift, Simes inequality.

## 1 Introduction

Conformal prediction is a popular tool for providing prediction sets with valid coverage (Vovk et al., 2005). The strength of the approach is that the guarantee holds for any underlying data-distribution, and can be combined with any machine learning algorithm. In this paper, we follow the split/inductive conformal prediction in a classification setting for which a machine has been pre-trained on an independent training sample (Papadopoulos et al., 2002; Vovk et al., 2005; Lei et al., 2014) and an independent calibration sample with *individual* labeled examples is available. We would like to use the calibration sample efficiently, to derive the prediction set for the label vector of a *batch* of new examples, without making any distributional assumption.

Formally, let $X_i \in \mathcal{X}$ (the space $\mathcal{X}$ is without restrictions) be the covariate and $Y_i \in [K]$[1] be the class label for example $i$. We observe a calibration sample $\{(X_i, Y_i), i \in [n]\}$, and only the covariates from the batch $\{(X_{n+i}, Y_{n+i}), i \in [m]\}$. We assume that a machine has been pre-trained (with an independent training sample) and is able to produce non-conformity scores $S_k(x)$ for any label $k \in [K]$ and any *individual* covariate $x \in \mathcal{X}$. The considered task is to produce a collection $\mathcal{C}^m_\alpha$ (called a batch prediction set) of batch label vectors $y := (y_i)_{i\in[m]} \in [K]^m$ such that one of the two following guarantees holds:

$$\mathbb{P}((Y_{n+i})_{i\in[m]} \in \mathcal{C}^m_\alpha) \geq 1-\alpha\,; \quad (1)$$

$$\mathbb{P}((Y_{n+i})_{i\in[m]} \in \mathcal{C}^m_\alpha \mid (Y_{n+i})_{i\in[m]} = y) \geq 1-\alpha\,, \quad (2)$$

where the guarantee in (2) is meant to hold for any possible batch $y \in [K]^m$. The unconditional guarantee in (1) is considered for the *iid model*, for which the probability is taken with respect to (wrt) the sample $\{(X_i, Y_i), i \in [n+m]\}$ which is assumed to have iid components. By contrast, the stronger conditional

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

[1] Throughout the paper, we denote by $[\ell]$ the set $\{1, \ldots, \ell\}$, for any integer $\ell \geq 1$.

guarantee in (2) is considered for the *conditional model* where the batch label vector $(Y_{n+i})_{i\in[m]}$ is fixed and the probability is taken wrt the distribution of the calibration sample $\{(X_i, Y_i), i \in [n]\}$ and the conditional distribution of $(X_{n+i})_{i\in[m]}$ given $(Y_{n+i})_{i\in[m]}$. Note that by independence, the conditional distribution of $(X_{n+i})_{i\in[m]}$ given $(Y_{n+i})_{i\in[m]}$ is simply equal to the product of the marginal distributions of $X_{n+i}$ given $Y_{n+i}$ for $i \in [m]$.

While unconditional guarantees of the type (1) are the most used targets for inference in the conformal literature (Angelopoulos and Bates, 2021; Angelopoulos et al., 2024), we emphasize that (2) is a much stronger guarantee (Vovk et al., 2005; Sadinle et al., 2019; Romano et al., 2020), which is a particular case of Mondrian conformal prediction. In our framework, since the true labels are fixed, the batch prediction set can be seen as a *batch confidence set*, that is, it is valid for all possible values of the true labels, and covers the case of a label distribution shift between the calibration sample and the batch: while methods built for the iid case implicitly use exchangeability of the labels and thus fail to cover the true batch in that case (see § F.4 for an illustration), methods with conditional coverage (2) cover the true batch even if the classes are arbitrarily unbalanced. This is of practical importance given that this situation is commonly met in real data sets.

The typical inference on a 'batch' only reports a prediction set for each example (Lee et al., 2024). By providing powerful methods that guarantee (1),(2), the inference is far more flexible. First, we can extract a prediction set for each example with a $1-\alpha$ coverage guarantee: for instance, (2) entails for all $y \in [K]^m$,

$$\mathbb{P}(\forall i \in [m],\, Y_{n+i} \in \mathcal{C}^m_{i,\alpha} \mid (Y_{n+i})_{i\in[m]} = y) \geq 1-\alpha,$$

where $\mathcal{C}^m_{i,\alpha}$ is the set of the $i$-th coordinates of all the vectors in $\mathcal{C}^m_\alpha$, that is, $\mathcal{C}^m_{i,\alpha} = \{y_i \in [K] : \exists (y_j)_{j\in[m]\setminus\{i\}} \in [K]^{m-1} : (y_j)_{j\in[m]} \in \mathcal{C}^m_\alpha\}$. In addition to this, we can also extract from the resulting batch prediction set bounds on the number of examples from each class. For any possible batch vector $y \in [K]^m$, let

$$m_k(y) := \sum_{i=1}^m \mathbf{1}\{y_i = k\},\;\; k \in [K], \tag{3}$$

be the number of examples from class $k$ in the batch $y$. The guarantees (1),(2) ensure that with (conditional) probability at least $1-\alpha$, all unknown numbers $m_k((Y_{n+i})_{i\in[m]})$ are included in a range

$$[\ell^{(k)}_\alpha, u^{(k)}_\alpha] := [\min \mathcal{N}_k(\mathcal{C}^m_\alpha), \max \mathcal{N}_k(\mathcal{C}^m_\alpha)], \tag{4}$$

where $\mathcal{N}_k(\mathcal{C}^m_\alpha) := \{m_k(y) : y \in \mathcal{C}^m_\alpha\}$, for all $k \in [K]$.

We mention two applications of our work, where the covariate corresponds to an image and we should produce a prediction set for the label vector of a *batch* of such images:

(i) Reading zip code (Vovk, 2013): given a machine trained to classify hand-written digits, we observe a written zip code, that is a batch of $m = 5$ images, and we should produce a list of plausible zip codes (a subset of $[K]^m$) for this batch; building $\mathcal{C}^m_\alpha$ ensuring (1) or (2) provides a solution, see also Figure 1a below.

(ii) Survey animal populations: given a machine trained to classify animal images, we observe a set of $m$ animal images and we should provide a prediction sets for the counts of each animal; building $[\ell^{(k)}_\alpha, u^{(k)}_\alpha]$ as in (4) provides a solution, see illustrations in § F.3.

In a very recent paper, Lee et al. (2024) suggest constructing prediction sets for functions of the batch points (e.g., for the mean or median outcome of the batch), assuming exchangeability of the calibration and test data, for both regression and classification. Their motivation is thus the same as ours, of providing model-free joint inference on multiple test points. They did not develop methodology targeting the inferential guarantees (1),(2). For their aims, they use a similar approach to the permutation approach we suggest in § 3.

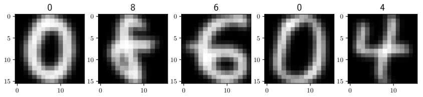
The guarantee (1) has been considered in Vovk (2013). To achieve the $1-\alpha$ guarantee, the problem of a batch prediction set is seen as the problem of testing at level $\alpha$ each of the $y \in [K]^m$ possible sets of labels. Vovk (2013) suggested in the full/transductive conformal setting using Bonferroni for each partitioning hypothesis. The advantage is that only $m \cdot K$ conformal $p$-values, i.e., $K$ for each example, need to be computed. So there is no need to go over all $K^m$ possible vectors of labels since $m \cdot K$ computations are enough. However, the computational simplicity comes at a severe cost: the batch prediction set using Bonferroni may be unnecessarily large, and thus less informative, than using more computationally intensive methods.

Our main contributions are as follows. We start by casting the problem of finding the batch prediction set as the problem of finding all the vectors that are not rejected when testing each of the $y \in [K]^m$ possible sets of labels in the conditional and iid models. By using the well-known Simes test, we show that there is a uniformly better (i.e., narrower) batch prediction set than Bonferroni's, that we refer to as the *Simes batch prediction set*. We further introduce an adaptive variant (its theoretical guarantee are established for conformal

$p$-values with a possible label shift, enriching the available literature (Storey, 2003; Bates et al., 2023)). We show how to construct batch prediction sets with any $p$-value combining function in Algorithm 1. We also provide a computational shortcut algorithm to compute the bounds (4) that maintains the $1-\alpha$ coverage guarantee. Finally, we provide an alternative method that combines batch scores rather than conformal $p$-values in Algorithm 2. We suggest the estimated likelihood ratio statistic, and show it has excellent power but a large computational cost compared to the methods that combine conformal $p$-values. We demonstrate the usefulness of our recommendations for image classification and USPS digits problems.[2]

The novel methods are available in two versions, each being valid for the iid or conditional model. The theoretical proofs are deferred to the supplementary file. The latter also contains additional illustrations, numerical experiments and mathematical materials.

To illustrate our method, Table 1 provides an example of batch prediction set for the particular zip code displayed in Figure 1a. For each combining function, Bonferroni or Simes, the proposed batch prediction set can be expressed as the batches with a $p$-value larger than $\alpha$ (see (6), (8) below). At 5%, we see that the Bonferroni batch prediction set is of size 8, whereas the Simes batch prediction set is of size 6 and is able to exclude the batches $(0, 6, 5, 5, 4)$ and $(0, 6, 6, 5, 4)$ from the prediction set. This is because all digits of the batch are acceptable according to Bonferroni's method, but are not acceptable *together* according to Simes' method. To show that this phenomenon is not due to the particular data generation, a violin plot for 500 replications is provided in Figure 1. Below the violin plot, the scatter plot of the number of rejections by each method clearly shows that the batch prediction set using Simes can be much narrower than using Bonferroni (and is never larger than using Bonferroni).



(a)

| 0 | 8 | 6 | 0 | 4 | Bonferroni | Simes |
|---|---|---|---|---|---|---|
| 0 | 6 | 5 | 5 | 4 | **0.065** | 0.038 |
| 0 | 6 | 6 | 5 | 4 | **0.065** | 0.038 |
| 0 | 6 | 5 | 0 | 4 | **0.065** | **0.065** |
| 0 | 6 | 6 | 0 | 4 | **0.065** | **0.065** |
| 0 | 8 | 5 | 5 | 4 | **0.077** | **0.077** |
| 0 | 8 | 6 | 5 | 4 | **0.077** | **0.077** |
| 0 | 8 | 5 | 0 | 4 | **0.277** | **0.277** |
| 0 | 8 | 6 | 0 | 4 | **0.605** | **0.345** |

(b)

Table 1: Batch prediction sets at level 0.05 for Bonferroni's and Simes' methods computed on the particular batch displayed above the table (from the USPS dataset provided by the US Postal Service for the paper LeCun et al. (1989)) previously studied by Vovk (2013)). Columns 6 and 7 provide the combination $p$-values using combining functions (7) and (9), respectively. The batch prediction set corresponds to batch $p$-values displayed in bold.

Finally, let us describe some related works. Our methodology is tightly related to the multiple testing literature, in particular Benjamini and Yekutieli (2001); Benjamini et al. (2006); Barber and Ramdas (2017); Bogomolov (2023); Heller and Solari (2023), where Simes and adaptive Simes variants are shown to be useful for inference on a family of null hypotheses. Existing work for the task of building prediction sets concentrated thus far primarily on providing a false coverage rate (FCR) guarantee (Bates et al., 2023; Gazin et al., 2024a,b; Jin and Ren, 2024). To derive our theoretical results, we rely on the literature on conformal novelty detection (Bates et al., 2023; Marandon et al., 2024) under the 'full null' configuration, that is, when the test sample is not contaminated by novelties. While we show that these works yield *de facto* the unconditional guarantee (1), we extend the theory to also cover the more challenging conditional guarantee (2). We emphasize that our work consider the setting where we observe a calibration sample of *examples* (not batches), as in Lee et al. (2024). If a calibration sample of *batches* is at hand, the usual conformal inference pipeline can (and should) be used by defining batch scores that take into account the interaction between batch elements (Messoudi et al., 2020, 2021; Johnstone and Cox, 2021; Johnstone and Ndiaye, 2022). In our work, the batch examples are assumed independent and the calibration sample only contains scores for individual examples, so our setting is markedly different.

## 2 Methods using combinations of conformal $p$-values

Henceforth, we make the classical assumption that the scores $S_{Y_i}(X_i)$, $i \in [n+m]$, have no ties almost surely.

[2] The code used in all our experiments is made publicly available at `https://github.com/ulyssegazin/BatchCP_Classification`

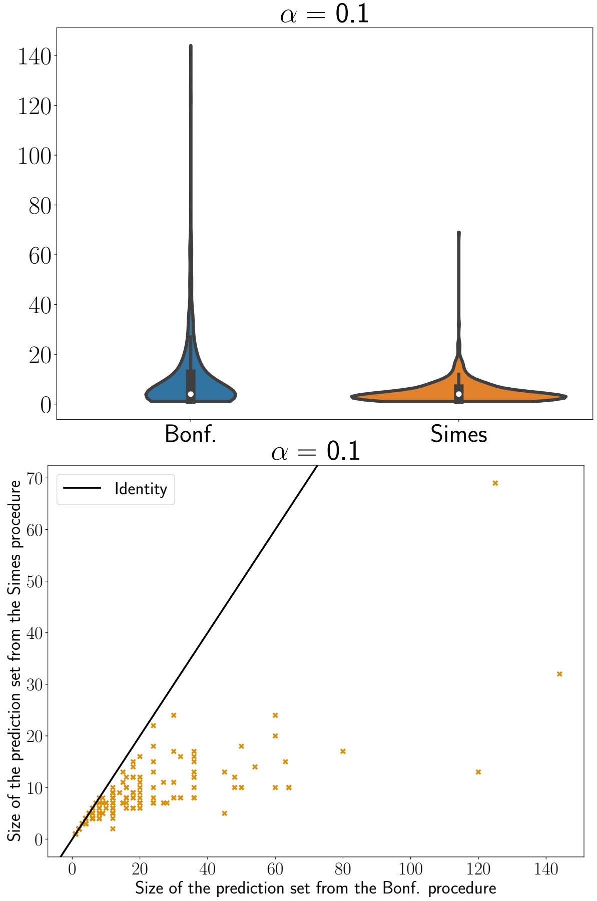


Figure 1: Violin plots (top row) and scatter plot (bottom row) for the size of the batch prediction sets of Bonferroni's and Simes' methods ($m = 3$, $K = 10$, 500 replications) for $\alpha = 0.1$. Generated from the USPS dataset, as described in § 4.2.

### 2.1 Conformal $p$-values

For $k \in [K]$, we consider the conformal $p$-value (Vovk et al., 2005) for testing the null "$Y_{n+i} = k$" versus "$Y_{n+i} \neq k$" in the test sample. Formally, the $p$-value family $(p_i^{(k)}, k \in [K], i \in [m])$ is given as follows:

$$p_i^{(k)} = \frac{1}{|\mathcal{D}_{\text{cal}}^{(k)}| + 1}\Big(1 + \sum_{j \in \mathcal{D}_{\text{cal}}^{(k)}} \mathbf{1}\{S_{Y_j}(X_j) \geq S_k(X_{n+i})\}\Big), \tag{5}$$

with $\mathcal{D}_{\text{cal}}^{(k)}$ being either $[n]$, of size $n$, in the iid setting or $\{j \in [n] : Y_j = k\}$, of size $n_k$, in the conditional setting. The $p$-values in (5) are referred to as *full-calibrated* $p$-values in the iid setting and *class-calibrated* $p$-values in the conditional setting.

Since scores $\{S_{Y_j}(X_j), j \in \mathcal{D}_{\text{cal}}^{(Y_{n+i})}\} \cup \{S_{Y_{n+i}}(X_{n+i})\}$ are exchangeable both in the iid and class-conditional setting, the following, well known property, holds.

**Proposition 2.1.** *The conformal $p$-values are marginally super-uniform, that is, for all $i \in [m]$, for all $u \in [0,1]$, $\mathbb{P}(p_i^{(Y_{n+i})} \leq u) \leq u$ for full-calibrated $p$-values and $\mathbb{P}(p_i^{(Y_{n+i})} \leq u \mid (Y_j)_{j \in [n+m]}) \leq u$ for class-calibrated $p$-values.*

Proposition 2.1 ensures that each individual label set $\mathcal{C}_{i,\alpha} := \{y_i \in [K] : p_i^{(y_i)} > \alpha\}$ is a prediction set for $Y_{n+i}$ of (conditional) coverage at least $1 - \alpha$.

### 2.2 Bonferroni batch prediction set

The Bonferroni batch prediction set is given as follows:

$$\begin{aligned} \mathcal{C}_{\alpha,\text{Bonf}}^m := \quad & \{y = (y_i)_{i \in [m]} \in [K]^m : \\ & F_{\text{Bonf}}((p_i^{(y_i)})_{i \in [m]}) > \alpha\}, \end{aligned} \tag{6}$$

where the $p$-value for the batch $y$ and for the Bonferroni method is given by

$$F_{\text{Bonf}}((p_i^{(y_i)})_{i \in [m]}) := m \min_{i \in [m]} \{p_i^{(y_i)}\}. \tag{7}$$

Hence, this prediction set is rectangular: $\mathcal{C}_{\alpha,\text{Bonf}}^m = \times_{i=1}^m \{k \in [K] : p_i^{(k)} > \alpha/m\}$; it is simply the product of standard individual conformal prediction sets, taken at level $1 - \alpha/m$. By Proposition 2.1 and a simple union bound, it is clear that (1) and (2) hold by using the full-calibrated and class-calibrated $p$-values, respectively.

### 2.3 Simes batch prediction set

Let us denote by $p_{(\ell)}((y_i)_{i \in [m]})$ the $\ell$-th largest element among the vector $(p_i^{(y_i)}, i \in [m])$. The Simes batch prediction set is given as follows:

$$\begin{aligned} \mathcal{C}_{\alpha,\text{Simes}}^m := \quad & \{y = (y_i)_{i \in [m]} \in [K]^m : \\ & F_{\text{Simes}}((p_i^{(y_i)})_{i \in [m]}) > \alpha\}, \end{aligned} \tag{8}$$

where the $p$-value for the batch $y$ and for the Simes method is given by

$$F_{\text{Simes}}((p_i^{(y_i)})_{i \in [m]}) := \min_{\ell \in [m]} \{m p_{(\ell)}(y)/\ell\}. \tag{9}$$

The latter always improves the Bonferroni batch prediction set, that is, $\mathcal{C}_{\alpha,\text{Simes}}^m \subset \mathcal{C}_{\alpha,\text{Bonf}}^m$ pointwise. Note that the Simes batch prediction set is not a hyper-rectangle, and cannot be obtained from the individual prediction sets of each element of the batch. In addition, the next result shows that it provides the correct (conditional) coverage.

**Theorem 2.2.** *The prediction set $\mathcal{C}_{\alpha,\mathit{Simes}}^m$ satisfies* (1) *and* (2) *by using the full-calibrated and class-calibrated $p$-values, respectively.*

To prove Theorem 2.2, we establish that the Simes inequality (Simes, 1986) holds for the class/full-calibrated $p$-values in § B.1. This comes from the fact that the conformal $p$-value family is positively dependent in a specific sense.

The conformal $p$-values are discrete, and therefore the guarantee (1) or (2) is typically a strict inequality. To resolve the conservativeness of the coverage that follows from the discreteness of the conformal $p$-values, a standard solution is to use randomized conformal $p$-values (Vovk, 2013). This solution is (arguably) unattractive since decisions are randomized. Interestingly, exact coverage is possible without need for randomization for specific values of $\alpha$ detailed in the following theorem.

**Theorem 2.3.** *The coverage for $\mathcal{C}^m_{\alpha,Simes}$ is exactly $1-\alpha$ in the two following cases:*

- *in the iid model, for full-calibrated p-values, if $\alpha(n+1)/m$ is an integer;*
- *in the conditional model, for class-calibrated p-values if $\alpha(n_k+1)/m$ is an integer for all $k \in [K]$.*

The proof is given in § B.2.

### 2.4 Adaptive Simes batch prediction set

For any possible label vector $y = (y_i)_{i\in[m]} \in [K]^m$, let

$$m_0(y) := \sum_{i\in[m]} \mathbf{1}\{y_i = Y_{n+i}\}, \tag{10}$$

the number of coordinates of $y$ that are equal to the true label vector $Y = (Y_{n+i})_{i\in[m]}$. Since $m_0(Y) = m$, the Simes batch prediction set $\mathcal{C}^m_{\alpha,\text{Simes}}$ has exactly the same coverage when replacing $m$ by $m_0(y)$ in the threshold. Meanwhile, using $m_0(y)$ may narrow the batch prediction set, because $m_0(y) < m$ for any vector $y \neq Y$. Unfortunately, $m_0(y)$ is unknown so that this improved prediction region is only an 'oracle' one that cannot be used. Our approach consists first in estimating $m_0(y)$ by

$$\hat{m}_0(y) := (1-\lambda)^{-1}\Big(1 + \sum_{i=1}^m \mathbf{1}\{p_i^{(y_i)} \geq \lambda\}\Big), \tag{11}$$

which is an analogue of the so-called Storey estimator in the multiple testing literature (Storey, 2002). Here, $\lambda \in (0,1)$ is a parameter that is free but should be such that $(n+1)\lambda$ is an integer in the iid setting, or such that $(n_k+1)\lambda$ is an integer for all $k \in [K]$ in the conditional setting. If these conditions are too strict, we can accommodate any value of $\lambda \in (0,1)$ by adjusting the formula (11) to account for discreteness: the modification is minor, see § A.

The adaptive Simes batch prediction set is

$$\begin{aligned}\mathcal{C}^m_{\alpha,\text{A-Simes}} := \quad & \{y = (y_i)_{i\in[m]} \in [K]^m \; : \\ & F_{\text{A-Simes}}((p_i^{(y_i)})_{i\in[m]}) > \alpha\},\end{aligned} \tag{12}$$

where the $p$-value for batch $y$ and for the adaptive Simes method is given by

$$F_{\text{A-Simes}}((p_i^{(y_i)})_{i\in[m]}) := \min_{\ell\in[m]}\{\hat{m}_0(y)\cdot p_{(\ell)}(y)/\ell\}, \tag{13}$$

and $\hat{m}_0(y)$ is an estimator of $m_0(y)$ (10), typically as in (11).

**Theorem 2.4.** *The coverage for $\mathcal{C}^m_{\alpha,A\text{-}Simes}$ with the Storey estimator* (11) *is at least $1-\alpha$ both in the iid model (using full-calibrated p-values) and in the conditional model (using class-calibrated p-values).*

The proof is given in § B.3. Note that the adaptive Simes method with estimator (11) (referred to as *Storey Simes* in what follows) does not provide a uniform improvement over Simes (or Bonferroni), because $\hat{m}_0(y) > m$ is possible for some batches $y$. However, $\hat{m}_0(y)$ is typically (much) smaller than $m$ for batches $y$ which are far from the true batch. Hence, the adaptive version leads to a substantial improvement in a situation where the batch prediction set is large ('weak' signal), see examples in § 4.

The tuning parameter $\lambda$ is by default chosen equal to $1/2$ but other choices are possible, see § A in the supplement. Therein, we also provide another type of estimator, corresponding to the so-called 'quantile' estimator (Benjamini et al., 2006; Marandon et al., 2024) and for which a choice of parameter is the 'median' estimator (and the corresponding method is referred to as *median Simes*). While we have no theoretical guarantee for median Simes, the performance of median Simes tends to be better than Storey Simes for strong signal and worse when the signal is weak, see § F.

### 2.5 General $p$-value combining prediction set

We present a general method for guaranteeing (1) and (2) using any combining function, denoted by $F((p_i^{(y_i)})_{i\in[m]})$, for the conformal $p$-values that test that the batch labels are $y \in [K]^m$. Consider a batch prediction set of the form

$$\mathcal{C}^m_{t,F} := \{(y_i)_{i\in[m]} \in [K]^m \, : \, F((p_i^{(y_i)})_{i\in[m]}) \geq t\}, \tag{14}$$

where $t$ is some threshold, possibly depending on the $p$-value vector. From Theorems 2.2 and 2.4, a valid choice is $t = \alpha$ and $F = F_{\text{A-Simes}}$ as in (13) with either $\hat{m}_0(y) = m$ or $\hat{m}_0(y)$ as in (11). Algorithm 1 shows how to find a valid empirical choice of $t$ for any $F$ (see also the simplified version given in § E, Algorithm 5, for the particular case of the iid model). Importantly for computation, the empirical threshold (line 10 in Algorithm 1) does not depend on the actual scores from the calibration and test examples. However, in the conditional model, the threshold depends on $(m_k(y))_{k\in[K]}$ so $B$ permutations of $[n+m]$ should be generated for every configuration of $(m_k)_{k\in[K]}$ such that $\sum_{k=1}^{K} m_k = m$ (where $m_k \in [0, m]$). Hence, the computational cost is more severe than for the iid model, which only requires $B$ permutations of $[n+m]$. However, these computations can be done once for all, before observing the data for the batch.

Finally, the attentive reader may have noticed that the inequality in (14) is not strict, which is in contrast with the previous sections. This is necessary to obtain a general valid coverage as stated in Theorem 2.5 below and is consistent with standard randomized test theory for the batch statistic $F((p_i^{(y_i)})_{i\in[m]})$ (Romano and Wolf, 2005), which is intended to be small when a rejection should be made. The prediction set (14) can be equivalently expressed as $\mathcal{C}_{t,F}^m = \{y \in [K]^m : \hat{q}(y) > \alpha\}$, where $\hat{q} = \hat{q}(y)$ is some batch $p$-value (see (23) in § B.4), possibly depending on $(m_k(y))_{k\in[K]}$ in the class-conditional case. This representation is similar to the representation in previous sections. Both representations have the same computational complexity, but the representation in the algorithm has the advantage of making it clear that the thresholds $t^{(m_k(y))_{k\in[K]}}$ do not depend on the data, i.e., the construction of the prediction set is *distribution free.*

**Theorem 2.5.** *The coverage of the batch prediction set $\mathcal{C}_{t,F}^m$ provided in Algorithm 1. is at least $1-\alpha$ both in the iid model (using full-calibrated p-values) and in the conditional model (using class-calibrated p-values). For the iid model, the outer loop (lines 1, 2, 12 in Algorithm 1) is not needed, see Algorithm 5 in § E.*

The proof is provided in § B.4. The method is very flexible: combined with adaptive Simes combination $F_{\mathrm{Simes}}$, any estimator $\hat{m}_0(\cdot)$ can be used, see detailed suggestions in § A. Since there is not one uniformly best estimator, and which estimator to use depends on the unknown properties of the data at hand, it is possible to take as $\hat{m}_0(\cdot)$ the smallest of several estimators of $\hat{m}_0(\cdot)$. More generally, any $p$-value combination can be used, for instance the Fisher combination

$$F_{\mathrm{Fisher}}((p_i^{(y_i)})_{i\in[m]}) = T\Big(-2\sum_{i\in[m]} \log(p_i^{(y_i)})\Big), \quad (15)$$

where $T$ is the survival function of a $\chi^2(2m)$ distribution. The corresponding method is referred to as *Fisher* batch prediction set in what follows. We refer to Heller and Solari (2023), and references within, for more examples of such combining functions.

**Algorithm 1:** Constructing a batch prediction set using conformal $p$-values combination

```
Input: Number of examples from class $k$ in the calibration set $n_k$, $k \in [K]$; combining function $F$; level $\alpha \in (0,1)$; number of permutations $B$; conformal $p$-values $(p_i^{(y_i)})_{i\in[m]}$.
1  for each possible allocation $h =: (h_k)_{k\in[K]}$ such that $0 \le h_k \le m$ and $\sum_{k=1}^{K} h_k = m$ do
2      define $z = z(h) = (z_i)_{i\in[m]} \in [K]^m$ as any element such that $m_k(z) = h_k$ for all $k \in [K]$;
3      for each $b \in [B]$ do
4          Generate a random permutation $\pi_b$ of $[n+m]$;
5          Compute null conformal $p$-values:
               $\hat{p}_{i,b}^{(z_i)} \leftarrow \frac{1 + \sum_{j\in\mathcal{D}_{\mathrm{cal}}^{(z_i)}} \mathbf{1}\{\pi_b(j) \ge \pi_b(n+i)\}}{|\mathcal{D}_{\mathrm{cal}}^{(z_i)}| + 1}$
             for $i \in [m]$;
6          Compute combined statistic:
7          $\xi_b^h \leftarrow F((\hat{p}_{i,b}^{(z_i)}, i \in [m]))$;
8      end
9      Compute threshold:
10     $t^h \leftarrow \xi^h_{(\lfloor (B+1)\alpha \rfloor)}$,
11     where $\xi^h_{(1)} \le \ldots \le \xi^h_{(B)}$ are the sorted values of $\xi_1^h, \ldots, \xi_B^h$ and $\xi^h_{(0)} := -\infty$;
12 end
13 Construct batch prediction set:
14 $\mathcal{C}_{t,F}^m \leftarrow \{y \in [K]^m : F((p_i^{(y_i)})_{i\in[m]}) \ge t^{(m_k(y))_{k\in[K]}}\}$;
Output: Batch prediction set $\mathcal{C}_{t,F}^m$.
```

## 2.6 Shortcut for computing bounds

Computing naively the bounds $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$ in (4) incurs exponential complexity and thus is difficult when both $K$ and $m$ increase. A pseudoalgorithm for a computational shortcut, which reduces the time complexity for calculating the bounds from $O(K^m)$ to $O(K \times m^2)$, is given in § D. This shortcut is exact when $K = 2$ and the scores produced by the machine learning model are probabilities, i.e. they satisfy the relationship $S_k(x_{n+i}) = 1 - S_{3-k}(x_{n+i})$ for $k \in \{1,2\}$ and $i \in [m]$. However, when $K > 2$ or when arbitrary scores are used, the shortcut may become conservative, resulting in wider bounds but never narrower ones. This ensures that the coverage guarantee of at least $1-\alpha$ probability is maintained. In Appendix F.1 we examine the performance of the shortcut in our numerical experiments. Interestingly, the bounds using the shortcut are almost identical to the bounds derived from the batch prediction set for Simes (see § F.1).

From the bounds produced by the shortcut, it is straightforward to produce a conservative batch prediction set. The size of the set is the sum of all valid assignments of $(m_1, \ldots, m_K)$ occurrences, where $\ell_\alpha^{(k)} \le m_k \le u_\alpha^{(k)}$ for each $k \in \{1, \ldots, K\}$, and $m_1 + \cdots + m_K = m$, with each valid assignment counted by the multinomial coefficient $\binom{m}{m_1, m_2, \ldots, m_K}$, see § D for more details.

Finally, we note that since for any $y \in [K]^m$, the rejection by Bonferroni necessarily entails rejection using Simes, then we can first apply the Bonferroni procedure, and then apply the suggested shortcut for Simes on the $(K-R_1)\times\cdots\times(K-R_m)$ remaining partitions, where $R_i$ are the number of conformal $p$ values at most $\alpha/m$ for the $i$-th example of the batch.

## 3 Method using batch scores

Thus far, we have considered methods that combine conformal $p$-values. Next, we present a general method for guaranteeing (1)-(2) using any function that combines the non-conformity scores of the batch. We suggest a specific function, the estimated likelihood ratio (LRT) statistic, which has been successfully used in hypothesis testing and has asymptotic optimality properties (Lehmann and Romano, 2005). We show in § 4 that the batch-score algorithm using the estimated LRT statistic has excellent power, but also non-negligible increased computational complexity, compared with the suggested methods that are based on combining conformal $p$-values, as shows in § F.5. The added computational complexity is due to the fact that the null distribution of the estimated LRT statistic depends on the actual scores. In contrast, the null distribution of the combination of conformal $p$-values does not depend on the actual scores (it does, however, depend on the number of examples from each class in the calibration set for the class conditional model).

Let $G((x_i)_{i\in[m]}, (y_i)_{i\in[m]})$ denote the batch-score function. We suggest $G((x_i)_{i\in[m]}, (y_i)_{i\in[m]}) = \prod_{i=1}^m \frac{\max_{k\in[K]}(1-S^{(k)}(x_i))}{1-S^{(y_i)}(x_i)}$. So our test statistic, called the *estimated LRT statistic*, for testing the null hypothesis that $(Y_{n+i})_{i\in[m]} = (y_i)_{i\in[m]}$, is $G((X_{n+i})_{i\in[m]}, (y_i)_{i\in[m]})$. This test statistic is expected to have excellent power when $1-S^{(k)}(X_{n+i})$ is close to the probability of observing $Y_{n+i}=k$ given $X_{n+i}$. To see this, suppose the true (unknown) batch label vector is $\tilde{y}$. Then the approximate likelihood of observing $(X_{n+i})_{i\in[m]}$ together with the true $\tilde{y}$ or together with the null $y$ is, respectively, $\Pi_{i=1}^m \left(1-S^{(\tilde{y}_i)}(X_{n+i})\right)\mathbb{P}(X_{n+i})$ or $\Pi_{i=1}^m \left(1-S^{(y_i)}(X_{n+i})\right)\mathbb{P}(X_{n+i})$, where $\mathbb{P}(X_{n+i})$ denotes the density of $X_{n+i}$ taken at point $X_{n+i}$ (when it exists and with a slight abuse of notation). So the approximate likelihood ratio is $\Pi_{i=1}^m\big(1-S^{(\tilde{y}_i)}(X_{n+i})\big)/\big(1-S^{(y_i)}(X_{n+i})\big)$. The numerator is evaluated using the maximum likelihood estimate for $\tilde{y}$ to obtain $G((X_{n+i})_{i\in[m]}, (y_i)_{i\in[m]})$.

**Proposition 3.1.** *The coverage of the batch prediction set $\mathcal{C}^m_{\alpha,G}$ provided in Algorithm 2 is at least $1-\alpha$ both in the iid model and in the conditional model. For the iid model, the restriction in line 4 of Algorithm 2 is not necessary.*

**Algorithm 2:** Constructing a batch prediction set using batch scores

```
Input: Calibration and test samples data
       $(X_i, Y_i)_{i\in[m]}$, $(X_{n+i})_{i\in[m]}$; a batch score
       function $G((x_i)_{i\in[m]}, (y_i)_{i\in[m]})$; level
       $\alpha \in (0,1)$; the number of permutations $B$
1 Initialize $C^m_{\alpha,G} \leftarrow \emptyset$
2 for each $y = (y_i)_{i\in[m]} \in [K]^m$ do
3     for each $b \in [B]$ do
4         Sample $m$ indices from the vector
            $(Y_1,\ldots,Y_n,y_1,\ldots,y_m)$. The $m$ indices are
            sampled with the restriction that the
            frequency of the classes in the 'test' sample is
            $\big(m_k((y_i)_{i\in[m]})\big)_{k\in[K]}$. Let
            $((x'_i)_{i\in[m(b)]}, (y'_i)_{i\in[m(b)]})$ denote the
            vectorized data in the 'test' sample.
5         Compute the $b$th null batch score
            $G_b := G((x'_i)_{i\in[m(b)]}, (y'_i)_{i\in[m(b)]})$.
6     end
7     The $p$-value for testing that $(Y_{n+i})_{i\in[m]} = y$, is
          $$p^{(y)} = \frac{1+\sum_{b=1}^B \mathbf{1}\{G_b \ge G((X_{n+i})_{i\in[m]}, (y_i)_{i\in[m]})\}}{B+1}.$$
8     If $p^{(y)} > \alpha$ then $C^m_{\alpha,G} \leftarrow C^m_{\alpha,G} \cup y$.
9 end
Output: Batch prediction set $\mathcal{C}^m_{\alpha,G}$.
```

The proof follows from standard theory on permutation tests, see, e.g., Theorem 2.4 in Angelopoulos et al. (2024). Specifically, for the class conditional model, the result follows since the non-coverage probability is equal to $\mathbb{P}\left(p^{(y)} \le \alpha \mid (Y_{n+i})_{i\in[m]} = y\right)$, which is $\le \alpha$ because the $B$ null batch scores generated for a specific $y$ in lines 3–6 of Algorithm 2 are exchangeable with the batch score test statistic when $(Y_{n+i})_{i\in[m]} = y$.

**Remark 3.1.** *We presented a computationally efficient shortcut for the bounds when using the p-value combining methods in § 2.6, and demonstrated in §D.4 that the bounds can be useful when m is large. Unfortunately, this shortcut is not possible for the estimated LRT statistic, since its (permutation) null distribution varies with the vector y being tested.*

## 4 Experiments

We study the performances of the following procedures: Bonferroni (6), Simes (8), Storey Simes (adaptive Simes (12) with the Storey estimator (16) where $\lambda = 1/2$), Fisher (15), and the estimated LRT (§ 3). We use the conditional setting, with class calibrated conformal $p$-values (5). The score function $S_k(x)$ is given by an estimator of the probability that $k$ is not the label of observation $x$.

### 4.1 Gaussian multivariate setting

We illustrate the substantial advantage of the new methods over Bonferroni for inferring on batch prediction sets in settings with different signal to noise ratio (SNR). We consider $K = 3$ categories, where the distribution of the covariate in each category is bivariate normal. The centers of the three categories are (0,0), (SNR,0), and (SNR,SNR). So the classification problem is more difficult as the SNR decreases. See § F.1 for one example of this data generation.

In Table 2 we show the results for a range of SNR values, in the setting with $n = 1200$, $m = 6$, and the calibration set and test sets have a fixed and equal number of examples from each of the three categories. As expected, using Simes is uniformly better than using Bonferroni. Adaptive Simes is far superior to both when the SNR is at most 2.5. For strong signal, using Simes produces slightly narrower batch prediction sets than using adaptive Simes. Fisher provides the narrowest batch prediction sets when the SNR is low. However, when the SNR is strong its performance is much worse even than Bonferroni. Thus, using Fisher is only recommended in situations where the batch prediction set is expected to be large. The estimated LRT statistic outperforms all the other practical methods when the SNR is $\geq 2.5$. Moreover, its batch prediction sets are only slightly wider than those obtained using the Fisher combining method when the SNR is $\leq 2.5$. All other methods, however, require less than 1/100 of the running time that is needed for the estimated LRT method. Thus it is the preferred method overall only if the practitioner has sufficient computing power.

| | | | Storey | | estimated |
|---|---|---|---|---|---|
| SNR | Bonf | Simes | Simes | Fisher | LRT |
| 1.00 | 410.52 | 384.66 | 327.55 | **274.36** | 277.58 |
| 1.50 | 217.69 | 187.36 | 142.98 | **107.85** | 113.88 |
| 2.00 | 81.63 | 65.52 | 49.12 | **37.40** | 37.76 |
| 2.50 | 23.51 | 17.98 | 15.08 | 14.60 | **11.91** |
| 3.00 | 6.42 | 5.35 | 5.18 | 7.78 | **4.35** |
| 3.50 | 2.46 | 2.24 | 2.27 | 5.20 | **2.02** |
| 4.00 | 1.39 | 1.34 | 1.37 | 4.38 | **1.28** |
| 4.50 | 1.07 | 1.06 | 1.08 | 4.03 | **1.03** |

Table 2: Average batch prediction set size at each SNR for different combining methods (columns). In bold, the combining method that produces the narrowest prediction region. $\alpha = 0.1$ and 2000 replications. For a single data generation, the average running time on a standard PC was less than 0.05 seconds for all methods but the estimated LRT, which has an average running time of 5.7 seconds.

In Appendix F.1, Table 7, the non-coverage probability is shown for each method, as well as the results for median Simes (adaptive Simes (12) with the 'median' estimator, see (18)), and oracle Simes that uses the true (unknown in practice) $m_0(y)$. As expected, oracle Simes leads to the narrowest batch prediction sets. For low SNR, the oracle statistic with the true $m_0(y)$ is far lower than all the practical test statistics. This suggests that optimizing the choice of estimate of $m_0(y)$ may improve the inference. As mentioned at the end of §2.5, one direction may be to use for $\hat{m}_0(y)$ the minimum of several good candidates. More generally, we could also use as combining function the minimum batch $p$-value from different combining functions. We leave for future work the investigation of the benefits from such a compound procedure.

In Appendix F.1, Tables 8 and 9, we show the bounds for each SNR. The bounds using Simes are slightly tighter than using Bonferroni. Interestingly, there seems to be no clear benefit for the bounds in using adaptive Simes or Fisher. However, the bounds using the method of combining batch scores with the estimated LRT are tighter when $SNR >= 2.5$. Appendix D.4 shows bounds in settings with $m$ large, which are computed using the available shortcut for the methods that combine conformal $p$-values, described in § D.

### 4.2 Real data sets

We use two datasets commonly used in the machine learning community, the USPS dataset (LeCun et al., 1989) with $K = 10$ digits and the CIFAR-10 dataset (Krizhevsky, 2009) restricted to $K = 3$ classes: "birds", "cats" and "dogs". For the USPS dataset, the calibration and batch sample sizes are $n = 700$ and $m = 3$, respectively. The score functions are derived using a support-vector classifier with the linear kernel (trained with 2431 examples). For the CIFAR-10 dataset, the calibration and batch sample sizes are $n = 2000$ and $m = 5$, respectively. We use a convolutional neural network with 8 layers, trained with 5666 examples with 10 epochs and the 'Adam' optimizer.

The coverage and violin plots of the size of the batch prediction sets for the different methods are displayed in Table 3 and Figure 2, respectively. For the USPS data set, the results strongly depend on the level $\alpha$ considered. For $\alpha = 0.01$, the batch prediction sets are all large and Fisher and LRT methods are the best. For $\alpha = 0.05$ and $\alpha = 0.1$, the best batch prediction sets are the LRT and the Simes methods. For the CIFAR data set, the sizes of the prediction sets are large for all $\alpha$ levels considered (meaning that the prediction task is more difficult on this data set). The Fisher combination and LRT method are comparable and better than the other ones, followed by the Storey Simes method.

These findings corroborate those of the previous section. Other qualitatively similar results are obtained in § F.2. Finally, we note that for $\alpha = 10\%$ the estimated coverage is less than one standard error (SE) below 90%.

| | USPS | | | CIFAR | | |
|---|---|---|---|---|---|---|
| Coverage | 0.99 | 0.95 | 0.90 | 0.99 | 0.95 | 0.90 |
| Bonf. | 1 | 0.966 | 0.932 | 0.993 | 0.958 | 0.896 |
| Simes | 1 | 0.966 | 0.928 | 0.993 | 0.958 | 0.890 |
| Storey | 1 | 0.986 | 0.942 | 0.993 | 0.954 | 0.893 |
| LRT | 0.990 | 0.962 | 0.916 | 0.992 | 0.957 | 0.892 |
| Fisher | 0.994 | 0.972 | 0.932 | 0.994 | 0.953 | 0.892 |

Table 3: Estimated coverage for $\alpha \in \{1\%, 5\%, 10\%\}$ and data sets USPS and CIFAR (in columns) and different procedures (in rows). Based on 500 simulation runs; all the standard errors are below 0.014

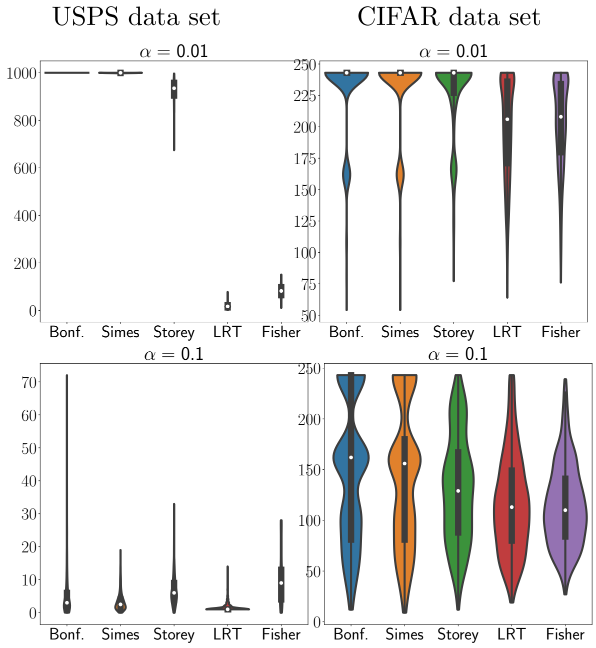


Figure 2: Violin plots for the size of the batch prediction set for $\alpha \in \{1\%, 10\%\}$ (rows) and data sets USPS and CIFAR (columns), see details in the text. The white dot inside the inter-quartile box of the violin plot is the median. The plots for $\alpha = 5\%$ (omitted) are qualitatively similar to the plots for $\alpha = 10\%$.

## 5 Discussion

For a batch of test points we provide, with a $(1-\alpha)$ coverage guarantee, a batch prediction set or bounds for the different classes, by testing that the batch label vector is $y \in [K]^m$ using two approaches: conformal $p$-value and batch score combination tests. We demonstrated that we can get much narrower batch prediction sets than using Bonferroni. For the bounds, the advantage over Bonferroni is modest, but nevertheless with Simes the improvement over Bonferroni is uniform. A further improvement is noted using the estimated LRT, the statistic suggested for the batch score combination test. However, the computation complexity is much larger with the estimated LRT, since the permutation null distribution of the batch score combination test depends on the $n + m$ scores. This is in contrast with Simes (and all other conformal $p$-value combination tests), for which the null distribution depends only on the number of examples from each class in the calibration sample, not on the realized scores, so in this sense it is distribution free.

As our numerical experiments show, there is no best method for combining the conformal $p$-values. Broadly, Fisher is good for weak signal and adaptive Simes is a better choice for the remaining cases. We can also consider combining the two using the algorithm in § 2.5 (a reasonable combining method is to take the minimum of the Fisher based and median Simes § 2.4). The lack of an overall best combining method is not surprising, since for every combining function that is reasonable there is a data generation that is optimal for it in a specific sense (Birnbaum, 1954; Heard and Rubin-Delanchy, 2018).

Our examples concentrated on a fairly small batch size $m$ and class size $K$. For $m$ or $K$ large we suggested, instead of testing all $y \in [K]^m$ to produce the bounds, to use a shortcut with computational complexity $O(K \cdot m^2)$. It is exact for $K = 2$, and appears tight for $K > 2$ in our numerical experiments. Specifically for Simes type combination tests, computationally efficient algorithms have been developed in the multiple testing literature (Goeman et al., 2019; Andreella et al., 2023). For large $m$ and $K$ it may be worthwhile to consider adapting their algorithms to our set-up for greater computational efficiency. A great challenge is to provide, for $m$ or $K$ large, efficient algorithms that directly target approximating the batch prediction set (rather than via the bounds). Relatedly, an open question is how to concisely summarize the batch prediction set when it is large.

## Acknowledgements

The authors acknowledge grants ANR-21-CE23-0035 (ASCAI) and ANR-23-CE40-0018-01 (BACKUP) of the French National Research Agency ANR, the Emergence project MARS of Sorbonne Université, and Israel Science Foundation grant no. 406/24.

## References

Andreella, A., Hemerik, J., Finos, L., Weeda, W., and Goeman, J. (2023). Permutation-based true discovery proportions for functional magnetic resonance imaging cluster analysis. *Statistics in Medicine*, 42(14):2311–2340.

Angelopoulos, A. N., Barber, R. F., and Bates, S. (2024). Theoretical foundations of conformal prediction. *arXiv preprint arXiv:2411.11824*.

Angelopoulos, A. N. and Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*.

Barber, R. F. and Ramdas, A. (2017). The p-filter: multilayer false discovery rate control for grouped hypotheses. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(4):1247–1268.

Bates, S., Candès, E., Lei, L., Romano, Y., and Sesia, M. (2023). Testing for outliers with conformal p-values. *Ann. Statist.*, 51(1):149–178.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. Ser. B*, 57(1):289–300.

Benjamini, Y., Krieger, A. M., and Yekutieli, D. (2006). Adaptive linear step-up procedures that control the false discovery rate. *Biometrika*, 93(3):491–507.

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Ann. Statist.*, 29(4):1165–1188.

Birnbaum, A. (1954). Combining independent tests of significance. *Journal of the American Statistical Association*, 49(267):559–574.

Blanchard, G. and Roquain, E. (2009). Adaptive false discovery rate control under independence and dependence. *J. Mach. Learn. Res.*, 10:2837–2871.

Bogomolov, M. (2023). Testing partial conjunction hypotheses under dependency, with applications to meta-analysis. *Electronic Journal of Statistics*, 17(1):102 – 155.

Gazin, U., Blanchard, G., and Roquain, E. (2024a). Transductive conformal inference with adaptive scores. In Dasgupta, S., Mandt, S., and Li, Y., editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 1504–1512. PMLR.

Gazin, U., Heller, R., Marandon, A., and Roquain, E. (2024b). Selecting informative conformal prediction sets with false coverage rate control. *arXiv preprint arXiv:2403.12295*.

Goeman, J. J., Meijer, R. J., Krebs, T. J. P., and Solari, A. (2019). Simultaneous control of all false discovery proportions in large-scale multiple hypothesis testing. *Biometrika*, 106(4):841–856.

Heard, N. A. and Rubin-Delanchy, P. (2018). Choosing between methods of combining-values. *Biometrika*, 105(1):239–246.

Heller, R. and Solari, A. (2023). Simultaneous directional inference. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(3):650–670.

Jin, Y. and Ren, Z. (2024). Confidence on the focal: Conformal prediction with selection-conditional coverage.

Johnstone, C. and Cox, B. (2021). Conformal uncertainty sets for robust optimization. In *Conformal and Probabilistic Prediction and Applications*, pages 72–90. PMLR.

Johnstone, C. and Ndiaye, E. (2022). Exact and approximate conformal inference in multiple dimensions. *arXiv preprint arXiv:2210.17405*.

Klenke, A. and Mattner, L. (2010). Stochastic ordering of classical discrete distributions. *Advances in Applied probability*, 42(2):392–410.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1989). Handwritten digit recognition with a back-propagation network. In *Neural Information Processing Systems*.

Lee, Y., Tchetgen, E. T., and Dobriban, E. (2024). Batch predictive inference.

Lehmann, E. L. and Romano, J. P. (2005). *Testing statistical hypotheses*. Springer Texts in Statistics. Springer, New York, third edition.

Lei, J., Rinaldo, A., and Wasserman, L. (2014). A conformal prediction approach to explore functional data. *Annals of Mathematics and Artificial Intelligence*, 74:29–43.

Marandon, A., Lei, L., Mary, D., and Roquain, E. (2024). Adaptive novelty detection with false discovery rate guarantee. *The Annals of Statistics*, 52(1):157–183.

Messoudi, S., Destercke, S., and Rousseau, S. (2020). Conformal multi-target regression using neural networks. In *Conformal and Probabilistic Prediction and Applications*, pages 65–83. PMLR.

Messoudi, S., Destercke, S., and Rousseau, S. (2021). Copula-based conformal prediction for multi-target regression. *Pattern Recognition*, 120:108101.

Papadopoulos, H., Proedrou, K., Vovk, V., and Gammerman, A. (2002). Inductive confidence machines for regression. In *13th European Conference on Machine Learning (ECML 2002)*, pages 345–356. Springer.

Romano, J. P. and Wolf, M. (2005). Exact and approximate stepdown methods for multiple hypothesis testing. *J. Amer. Statist. Assoc.*, 100(469):94–108.

Romano, Y., Barber, R. F., Sabatti, C., and Candès, E. (2020). With malice toward none: Assessing uncertainty via equalized coverage. *Harvard Data Science Review*, 2(2):4.

Sadinle, M., Lei, J., and Wasserman, L. (2019). Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 114(525):223–234.

Simes, R. J. (1986). An improved Bonferroni procedure for multiple tests of significance. *Biometrika*, 73(3):751–754.

Storey, J. D. (2002). A direct approach to false discovery rates. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 64(3):479–498.

Storey, J. D. (2003). The positive false discovery rate: a Bayesian interpretation and the $q$-value. *Ann. Statist.*, 31(6):2013–2035.

Vovk, V. (2013). Transductive conformal predictors. In *Artificial Intelligence Applications and Innovations: 9th IFIP WG 12.5 International Conference (AIAI 2013)*, pages 348–360. Springer.

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic learning in a random world*. Springer.

## Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes:** The goal is theoretical and methodological. We strived to present the material as clearly and transparently as possible.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Yes:** The issue of time complexity is discussed in § D.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes:** All the code are publicly available at `https://github.com/ulyssegazin/BatchCP_Classification` All codes used for the numerical experiments are made available in supplementary files: one is coded in R (and is devoted to the synthetic simulations) and one is in Python and is coded in a Jupyter notebook (and is devoted to the real data set experiments).

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes.**
   (b) Complete proofs of all theoretical results. **Yes.**
   (c) Clear explanations of any assumptions. **Yes:** the iid and conditional models are standard, and assuming no ties in the scores is common in conformal inference.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes:** the code is in the supplementary material and some simulation results are stored in '.npy' file.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Partially:** The values for training, calibration and test sizes are given. The parameters of the underlying machine learning black boxes are not explicitly discussed but can be found in the codes.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes:** we defined precisely

the quality criteria displayed in the experiments

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **No:** All the experiments were run on a standard laptop. Given the time complexity of the procedures, some simulations required several hours or days to complete.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. **Yes:** We provided them in the bibliography.

   (b) The license information of the assets, if applicable. **Yes:** Data sets from OpenML under (CC-BY) license and conformal prediction code is publicly available (scientific publications) without explicit license given by A. Marandon (see https://github.com/arianemarandon/infoconf).

   (c) New assets either in the supplemental material or as a URL, if applicable. **Yes.**

   (d) Information about consent from data providers/curators. **Not Applicable:** data sets from OpenML are under (CC-BY) License, with explicit public consent to reuse.

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not Applicable**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. **Not Applicable**

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable**

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable**

# Supplementary Material for "Powerful batch conformal prediction for classification"

This supplementary file contains additional details for the adaptive Simes procedure, additional experiments, the proofs of the results of the main paper (with auxiliary results) and more materials for the computational shortcut.

## A Estimators for $m_0(y)$

This section complements § 2.4.

### A.1 Storey and quantile type estimator

We first provide the general formula (10) for the Storey-type estimator $\hat{m}_0(y)$ that can accommodate any choice of $\lambda \in (0,1)$.

First, in the iid model, the modification corresponds to a simple rounding:

$$\hat{m}_0(y) := (1-\lambda)^{-1}\Big(1 + \sum_{i\in[m]} \mathbf{1}\{p_i^{(y_i)} \geq \lfloor (n+1)\lambda \rfloor/(n+1)\}\Big).$$

Clearly, the formula reduces to (11) when $(n+1)\lambda$ is an integer.

In the conditional model, the modification corresponds to a rounding on each class:

$$\hat{m}_0(y) := \kappa(y)\left(1 + \sum_{k\in[K]} \sum_{i:y_i=k} \mathbf{1}\{p_i^{(k)} \geq \lambda_k\}\right), \tag{16}$$

with $\lambda_k = \frac{\lfloor \lambda(n_k+1)\rfloor}{n_k+1}$ for $k \in [K]$. Above, the parameter $\kappa(y)$ is given by

$$\kappa(y) = \Big(1 - \min_{k\in[K]} \lambda_k\Big)^{\frac{1}{m-1}} \times \prod_{k\in[K]} \Big(\frac{1}{1-\lambda_k}\Big)^{\frac{m_k(y)}{m-1}}, \tag{17}$$

where we recall that $m_k(y)$ is given by (3). When $(n_k+1)\lambda$ is an integer for each $k \in [K]$, then $\lambda_k = \lambda$, $\kappa(y) = (1-\lambda)^{-1}$, and the formula reduces to (11).

Second, the 'quantile' estimator (Benjamini et al., 2006) is given by

$$\hat{m}_0(y) = \frac{m - \ell + 1}{1 - p_{(\ell)}(y)}, \tag{18}$$

for some $\ell \in [m]$, typically $\ell = \lceil m/2 \rceil$ for the 'median' estimator. The adaptive Simes batch prediction set using the quantile estimator satisfies the correct coverage in the iid model by Marandon et al. (2024). Proving such a coverage result in the class-conditional model is an open problem, although our numerical experiments seem to indicate that the control is maintained in that case (for the median estimator).[3]

[3] Recall that a valid coverage for the quantile Simes procedure can be ensured by using the empirical method of § 2.5 (not used in our numerical experiments).

### A.2 Choice of the tuning parameters

We discuss the choice of the parameter $\lambda \in (0,1)$ in the Storey estimator (16) (procedure denoted by $\lambda$-S for short), and of the parameter $\ell$ in the quantile estimator (18). In the latter, we let $\ell = \lceil qm \rceil$ and discuss rather the choice of $q$ (the corresponding procedure is denoted by $q$-Q for short).

The results are displayed in Figure 2 for the USPS and CIFAR data sets. For the Storey estimator, while no choice of $\lambda$ seems to be universally the best, this choice affects the performance of the method: we observe that choosing $\lambda = 1/2$ is better for the data set with weak signal (CIFAR) while choosing $\lambda$ small (and of the order of $\alpha$) is better for the data set with strong signal (USPS). This is coherent with the intuition behind the Storey estimator which implicitly supposes that the $p$-values above $\lambda$ are under the null. For the quantile procedure, the conclusion is similar to some extent, but the median procedure seems to have a good behavior for both data sets. Roughly, the latter can be seen as a Storey procedure with an adaptive choice $\lambda = p_{(\lceil m/2 \rceil)}(y)$, which is able to better adapt to the signal strength. These conclusions corroborate previous findings in the multiple testing literature under independence (Benjamini et al., 2006; Blanchard and Roquain, 2009), see also (Birnbaum, 1954; Heard and Rubin-Delanchy, 2018).

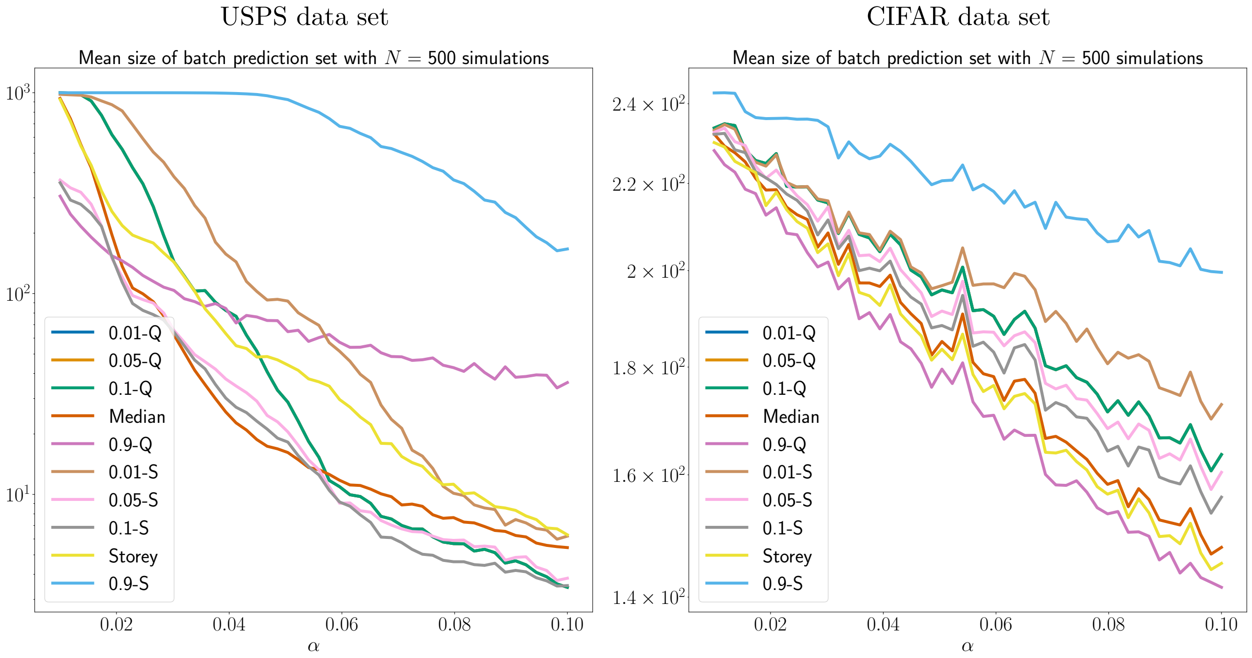


Figure 3: Averaged size of the batch prediction sets in function of $\alpha$ for different procedures (see text). Storey is 0.5-S and Median is 0.5-Q. Same setting as for Figure 2 .

## B Proofs

In this section, we prove Theorems 2.2, 2.3 and 2.4. The proofs follow from previous literature for the iid model (and full-calibrated $p$-values):

- Theorem 2.2 for the iid model is a consequence of Benjamini and Yekutieli (2001) and of the fact that the full-calibrated $p$-values are PRDS (Bates et al., 2023) (see definition below);
- Theorem 2.3 for the iid model is a consequence of Corollary 3.5 in Marandon et al. (2024);
- Theorem 2.4 for the iid model is a consequence of Corollary 3.7 in Marandon et al. (2024).

Below, we extend these arguments to the case of *the conditional model with class-calibrated $p$-values*. The main technical tool for the proof is Lemma C.2 (for comparison, we also recall Lemma C.1 that was obtained for the iid case with full calibrated $p$-values). On an intuitive point of view, the main idea of this extension is that, conditionally on $(Y_j)_{j\in[n+m]}$, each class-conditional conformal $p$-value $p_i^{(Y_{n+i})}$ depends on the $p$-values of the same class $(p_j^{(Y_{n+j})})_{j\in[m]\backslash\{i\}:Y_{n+j}=Y_{n+i}}$ exactly in the same way as for the iid case, and are independent of the $p$-values of the other classes $(p_j^{(Y_{n+j})})_{j\in[m]\backslash\{i\}:Y_{n+j}\neq Y_{n+i}}$.

Below, we write $p_i$ instead of $p_i^{(Y_{n+i})}$ for simplicity. Also, $n_i$ stands for $n_{Y_{n+i}}$ with a slight abuse of notation (recall that $n_k$ is the cardinal of $\mathcal{D}_{\text{cal}}^{(k)}$).

### B.1 Proof of Theorem 2.2

It is sufficient to establish the following Simes inequality for class-calibrated $p$-values:

$$\mathbb{P}(\exists \ell \in [m],\, p_{(\ell)} \leq \alpha\ell/m \mid (Y_j)_{j\in[n+m]}) \leq \alpha\,. \tag{19}$$

Since the families of class-calibrated $p$-values are marginally super-uniform (conditionally on $(Y_{n+i})_{i\in[m]}$), see Proposition 2.1, and by classical FDR controlling theory (Benjamini and Yekutieli, 2001), it is enough to prove that the following PRDS property on $m$ holds: for any nondecreasing[4] set $D\subset[0,1]^m$, the function

$$u \mapsto \mathbb{P}((p_i)_{i\in[m]} \in D \mid p_i = u, (Y_j)_{j\in[n+m]}),$$

is nondecreasing for all $i\in[m]$.

**Proposition B.1.** *In the conditional model, the family of class-calibrated $p$-values is PRDS on $[m]$.*

The proof relies on the general property of Lemma C.7, establishing that per-group PRDS for independent groups yields entire set PRDS.

*Proof.* Let us work conditionally on $(Y_j)_{j\in[n+m]}$ and consider the partition of $[m]$ given by $G_k = \{j \in [m] : Y_{n+j} = k\}$ then we know that for each $k\in[K]$, $(p_j)_{j\in G_k}$ is a family which is PRDS on $G_k$ (Bates et al., 2023). In addition, the $p$-values $(p_j)_{j\in G_k}$ and $(p_j)_{j\in G_{k'}}$ are independent for $k\neq k'$, because the calibration samples are not the same for each (since we use class-calibrated $p$-values). Hence, we can apply Lemma C.7 to conclude. □

### B.2 Proof of Theorem 2.3

To establish the result, we use the well known relationship between the Simes inequality and the FDR control of BH procedure under the full null, see, e.g., § 2.2 in Barber and Ramdas (2017). Let us denote for any $y=(y_i)_{i\in[m]}\in[K]^m$,

$$\widehat{\ell}(\mathbf{p}(y)) = \max\{\ell\in[m] \,:\, p_{(\ell)}(y) \leq \alpha\ell/m\}, \tag{20}$$

(with the convention $\widehat{\ell}(\mathbf{p}(y))=0$ if the set is empty) the number of rejections of the BH procedure (Benjamini and Hochberg, 1995) associated to the $p$-value family $\mathbf{p}(y)=(p_i^{(y_i)})_{i\in[m]}$. Observe that, $y\notin\mathcal{C}^m_{\alpha,\text{Simes}}$ if and only if $\widehat{\ell}(\mathbf{p}(y))\geq 1$. In addition, the latter holds if and only if $\sum_{i\in[m]}\mathbf{1}\{p_i^{(y_i)}\leq(\alpha/m)(1\vee\widehat{\ell}(\mathbf{p}(y)))\} = 1\vee\widehat{\ell}(\mathbf{p}(y))$.

Therefore, denoting $\mathbf{p}=(p_i)_{i\in[m]}$ the family of class-calibrated $p$-values, we can express the non-coverage probability as follows:

$$\mathbb{P}((Y_{n+i})_{i\in[m]}\notin\mathcal{C}^m_{\alpha,\text{Simes}} \mid (Y_j)_{j\in[n+m]}) = \sum_{i\in[m]}\mathbb{E}\Big[\frac{\mathbf{1}\{p_i\leq(\alpha/m)(1\vee\widehat{\ell}(\mathbf{p}))\}}{1\vee\widehat{\ell}(\mathbf{p})}\,\Big|\,(Y_j)_{j\in[n+m]}\Big]. \tag{21}$$

Consider $\mathbf{p}'=(p_i')_{i\in[m]}$ the vector defined in Lemma C.2 (v) with in addition $p_j'=p_j$ for $j\in[m]:Y_{n+j}\neq Y_{n+i}$. Combining Lemma C.2 (v) with Lemma C.3, we obtain

$$\{p_i\leq\alpha\widehat{\ell}(\mathbf{p})/m\} = \{p_i\leq\alpha\widehat{\ell}(\mathbf{p}')/m\}\subset\{\widehat{\ell}(\mathbf{p})=\widehat{\ell}(\mathbf{p}')\}.$$

[4] A set $D\subset[0,1]^m$ is nondecreasing if for $x=(x_j)_{1\leq j\leq m}\in D$ and $y=(y_j)_{1\leq j\leq m}\in\mathbb{R}^m$, $(\forall j\in[m], x_j\leq y_j)$ implies $y\in D$.

Hence, by letting $L_i = 1 \vee \widehat{\ell}(\mathbf{p}') \in [m]$, which is $W_i$-measurable (as defined in Lemma C.2), we have that (21) can be written as

$$\begin{aligned}\mathbb{P}((Y_{n+i})_{i\in[m]} \notin \mathcal{C}^m_{\alpha,\mathrm{Simes}} \mid (Y_j)_{j\in[n+m]}) &= \sum_{i\in[m]} \mathbb{E}\Big[\frac{\mathbf{1}\{p_i \le (\alpha/m)L_i\}}{L_i} \;\Big|\; (Y_j)_{j\in[n+m]}\Big] \\ &= \sum_{i\in[m]} \mathbb{E}\Big[\frac{\mathbb{P}(p_i \le (\alpha/m)L_i \mid W_i)}{L_i} \;\Big|\; (Y_j)_{j\in[n+m]}\Big].\end{aligned}$$

Now, by Lemma C.2 (ii), we have $\mathbb{P}(p_i \le (\alpha/m)L_i \mid W_i) = \frac{\lfloor (n_i+1)(\alpha/m)L_i \rfloor}{n_i+1} = (\alpha/m)L_i$ if $(n_i+1)(\alpha/m)$ is an integer for all $i \in [m]$. This finishes the proof.

### B.3 Proof of Theorem 2.4

Recall $\lambda_k = \frac{\lfloor \lambda(n_k+1)\rfloor}{n_k+1}$ for $k \in [K]$. For short, we sometimes write in this proof $\lambda_i$, $m_i$ and $n_i$ instead of $\lambda_{Y_{n+i}}$, $m_{Y_{n+i}}$ and $n_{Y_{n+i}}$ respectively, for all $i \in [m]$. Also, we write $\kappa$ instead of $\kappa((Y_{n+i})_{i\in[m]})$ and $m_k$ instead of $m_k((Y_{n+i})_{i\in[m]})$.

Let $G(\mathbf{p}) = \hat{m}_0((Y_{n+i})_{i\in[m]}) = \kappa(1 + \sum_{i=1}^m \mathbf{1}\{p_i \ge \lambda_i\})$ the estimator of $m_0$ at the true point $(Y_{n+i})_{i\in[m]}$ given in (16) (this means that this proof deals with the general case and not only the simple Storey estimator described in (11)). Similarly to (21), we have

$$\mathbb{P}((Y_{n+i})_{i\in[m]} \notin \mathcal{C}^m_{\alpha,\text{A-Simes}} \mid (Y_j)_{j\in[n+m]}) = \sum_{i\in[m]} \mathbb{E}\Big[\frac{\mathbf{1}\{p_i \le (\alpha/G(\mathbf{p}))(1 \vee \widehat{\ell}(\mathbf{p}))\}}{1 \vee \widehat{\ell}(\mathbf{p})} \;\Big|\; (Y_j)_{j\in[n+m]}\Big]$$

for $\widehat{\ell}(\mathbf{p}) = \max\{\ell \in [m] \;:\; p_{(\ell)} \le \alpha\ell/G(\mathbf{p})\}$ (with the convention $\widehat{\ell}(\mathbf{p}) = 0$ if the set is empty). Now we use Lemma C.2 and the notation therein, and we observe that

$$(p_j)_{j\in[m]\backslash\{i\}} = ((p_j)_{j\in[m]\backslash\{i\}:Y_{n+j}=Y_{n+i}}, (p_j)_{j\in[m]:Y_{n+j}\neq Y_{n+i}}) = (\Psi_i(p_i, W_i), (p_j)_{j\in[m]:Y_{n+j}\neq Y_{n+i}})$$

(with some abuse of notation in the ordering of the vector) is a function of $(p_i, W_i)$ which is nondecreasing in $p_i$. Hence, $1/G(\mathbf{p})$ and $1 \vee \widehat{\ell}(\mathbf{p})$ are functions of $(p_i, W_i)$, say $\Psi_2(p_i, W_i)$ and $\Psi_3(p_i, W_i)$ respectively, which are both nonincreasing in $p_i$. Now let

$$\begin{aligned}&c^*(W_i) = \max \mathcal{N}(W_i) \\ &\mathcal{N}(W_i) = \{a/(n_i+1) \;:\; a \in [n_i+1], a/(n_i+1) \le \alpha\Psi_2(a/(n_i+1), W_i)\Psi_3(a/(n_i+1), W_i)\},\end{aligned}$$

with the convention $c^*(W_i) = (n_i+1)^{-1}$ if $\mathcal{N}(W_i)$ is empty. Since $1 \vee \widehat{\ell}(\mathbf{p}) \ge 1 \vee \widehat{\ell}(c^*(W_i), (p_j)_{j\in[m]\backslash\{i\}})$, we have

$$\begin{aligned}\mathbb{P}((Y_{n+i})_{i\in[m]} \notin \mathcal{C}^m_{\alpha,\text{A-Simes}} \mid (Y_j)_{j\in[n+m]}) &\le \sum_{i\in[m]} \mathbb{E}\Big[\frac{\mathbb{P}(p_i \le c^*(W_i), p_i \in \mathcal{N}(W_i) \mid W_i)}{\Psi_3(c^*(W_i), W_i)} \;\Big|\; (Y_j)_{j\in[n+m]}\Big] \\ &\le \sum_{i\in[m]} \mathbb{E}\Big[\frac{c^*(W_i)}{\Psi_3(c^*(W_i), W_i)} \;\Big|\; (Y_j)_{j\in[n+m]}\Big] \\ &\le \alpha \sum_{i\in[m]} \mathbb{E}\Big[\Psi_2(1/(n_i+1), W_i)\Big|\; (Y_j)_{j\in[n+m]}\Big],\end{aligned}$$

where the first inequality comes from the definition of $\mathcal{N}(W_i)$ and $c^*(W_i)$ and from the fact that $\Psi_3(c^*(W_i), W_i)$ is $W_i$-measurable; the second inequality comes from Lemma C.2 (ii); and the third one comes from the fact that $c^*(W_i)$ is in $\mathcal{N}(W_i)$ and $\Psi_2(a/(n_i+1), W_i)$ is nonincreasing in $a$. Given the notation of Lemma C.2 (v), this leads to

$$\mathbb{P}((Y_{n+i})_{i\in[m]} \notin \mathcal{C}^m_{\alpha,\text{A-Simes}} \mid (Y_j)_{j\in[n+m]}) \le \alpha \sum_{i\in[m]} \mathbb{E}\Big[\frac{1}{G(\mathbf{p}')}\Big], \tag{22}$$

where $\mathbf{p}' = (p'_j)_{j\in[m]}$ is such that $p'_i = (n_i+1)^{-1}$, $(p'_j)_{j\in[m]:Y_{n+j}=Y_{n+i}} \sim \mathcal{L}_{i,(Y_j)_{j\in[n+m]}}$ and for each $k \neq Y_{n+i}$, $(p'_j)_{j\in[m]:Y_{n+j}=k} \sim \mathcal{L}^{(k)}_{(Y_j)_{j\in[n+m]}}$ where the distribution of $\mathcal{L}_{i,(Y_j)_{j\in[n+m]}}$ and $\mathcal{L}^{(k)}_{(Y_j)_{j\in[n+m]}}$ are defined in Lemma C.2.

Also note that $(p'_j)_{j\in[m]:Y_{n+j}=Y_{n+i}}$ and all $(p'_j)_{j\in[m]:Y_{n+j}=k}$, $k\neq Y_{n+i}$, are independent vectors, so that the distribution of $\mathbf{p}'$ is well specified. Now observe that (all expectations/probabilities below are taken implicitly conditionally on $(Y_j)_{j\in[n+m]}$)

$$\begin{aligned}\mathbb{E}\Big[\frac{1}{G(\mathbf{p}')}\Big] &= \mathbb{E}\Big[\frac{1/\kappa}{1+\sum_{j=1}^m \mathbf{1}\{p'_j\geq \lambda_j\}}\Big]\\ &= \mathbb{E}\Big[\frac{1/\kappa}{1+\sum_{k\neq Y_{n+i}}\sum_{j:Y_{n+j}=k}\mathbf{1}\{p'_j\geq \lambda_j\}+\sum_{j\in[m]\setminus\{i\}:Y_{n+j}=Y_{n+i}}\mathbf{1}\{p'_j\geq \lambda_j\}}\Big]\\ &= \mathbb{E}\Big[\frac{1/\kappa}{1+\sum_{k\neq Y_{n+i}}\mathcal{B}(m_k,\nu_k)+\mathcal{B}(m_i-1,\nu'_i)}\Big],\end{aligned}$$

by using Lemma C.2 (iii), (iv), where $\mathcal{B}(a,b)$ denotes (independent) binomial variables of parameters $a$ and $b$, and where $\nu_k=U^{(k)}_{(\lfloor(n_k+1)\lambda\rfloor-1)}$ (with the convention $\nu_k=1$ if $\lfloor(n_k+1)\lambda\rfloor\leq 1$) and $\nu'_i=U_{(\lfloor(n_i+1)\lambda\rfloor)}$ (with the convention $\nu'_i=1$ if $\lfloor(n_i+1)\lambda\rfloor=0$). The latter comes from the fact that for $j\in[m]$ such that $Y_{n+j}=k\neq Y_{n+i}$,

$$\begin{aligned}\mathbb{P}(p'_j\geq\lambda_j\mid(U^{(k)}_{(1)},\dots,U^{(k)}_{(n_k)})) &= \mathbb{P}\Big(\sum_{s\in A_i}\mathbf{1}\{s\geq S_{n+j}\}>\lfloor\lambda(n_k+1)\rfloor-1\;\Big|\;(U^{(k)}_{(1)},\dots,U^{(k)}_{(n_k)})\Big)\\ &= 1-(1-U^{(k)}_{(\lfloor(n_k+1)\lambda\rfloor-1)})=U^{(k)}_{(\lfloor(n_k+1)\lambda\rfloor-1)}.\end{aligned}$$

Similarly, for $j\neq i$ such that $Y_{n+j}=Y_{n+i}$, $\mathbb{P}(p'_j\geq\lambda_j\mid(U_{(1)},\dots,U_{(n_i+1)}))=U_{(\lfloor(n_i+1)\lambda\rfloor)}$.

Now, by Lemma C.4, we have $\nu_k\sim\beta(n_k+2-\lfloor(n_k+1)\lambda\rfloor,\lfloor(n_k+1)\lambda\rfloor-1)$ and $\nu'_i\sim\beta(n_i+2-\lfloor(n_i+1)\lambda\rfloor,\lfloor(n_i+1)\lambda\rfloor)$. Let $\nu$ be the random variable

$$\nu=(\nu'_i)^{m_i/m}\prod_{k\neq Y_{n+i}}(\nu_k)^{m_k/m}.$$

By the stochastic domination argument of Lemma C.5, we have

$$\begin{aligned}&\mathbb{E}\Big[\frac{1}{1+\sum_{k\neq Y_{n+i}}\mathcal{B}(m_k,\nu_k)+\mathcal{B}(m_i-1,\nu'_i)}\;\Big|\;(\nu_k)_{k\neq Y_{n+i}},\nu'_i\Big]\\ &\leq \mathbb{E}\Big[\frac{1}{1+\sum_{k\neq Y_{n+i}}\mathcal{B}(m-1,\nu)}\;\Big|\;(\nu_k)_{k\neq Y_{n+i}},\nu'_i\Big]\leq 1/(m\nu),\end{aligned}$$

where we used Lemma C.6 in the last inequality. As a result,

$$\begin{aligned}\sum_{i\in[m]}\mathbb{E}\Big[\frac{1}{G(\mathbf{p}')}\Big] &\leq \kappa^{-1}m^{-1}\sum_{i\in[m]}\mathbb{E}\Big((\nu'_i)^{-(m_i-1)/(m-1)}\prod_{k\neq Y_{n+i}}(\nu_k)^{-m_k/(m-1)}\Big)\\ &= \kappa^{-1}m^{-1}\sum_{i\in[m]}\mathbb{E}\big((\nu'_i)^{-(m_i-1)/(m-1)}\big)\prod_{k\neq Y_{n+i}}\mathbb{E}\big((\nu_k)^{-m_k/(m-1)}\big),\end{aligned}$$

by using the independence between the variables $\nu'_i$, $\nu_k$, $k\neq Y_{n+i}$. By Jensen's inequality, the last display is at most

$$\begin{aligned}&m^{-1}\sum_{i\in[m]}\kappa^{-1}(\mathbb{E}((\nu'_i)^{-1}))^{(m_i-1)/(m-1)}\prod_{k\neq Y_{n+i}}(\mathbb{E}((\nu_k)^{-1})))^{m_k/(m-1)}\\ &= m^{-1}\sum_{i\in[m]}\kappa^{-1}\Big(\frac{n_i+1}{n_i+1-\lfloor(n_i+1)\lambda\rfloor}\Big)^{(m_i-1)/(m-1)}\prod_{k\neq Y_{n+i}}\Big(\frac{n_k}{n_k+1-\lfloor(n_k+1)\lambda\rfloor}\Big)^{m_k/(m-1)}\\ &\leq m^{-1}\sum_{i\in[m]}\kappa^{-1}\Big(\frac{1}{1-\lambda_i}\Big)^{(m_i-1)/(m-1)}\prod_{k\neq Y_{n+i}}\Big(\frac{1}{1-\lambda_k}\Big)^{m_k/(m-1)}\leq 1,\end{aligned}$$

because $\mathbb{E}(\nu_k^{-1})=\frac{n_k}{n_k+1-\lfloor(n_k+1)\lambda\rfloor}\leq\frac{n_k+1}{n_k+1-\lfloor(n_k+1)\lambda\rfloor}$ and $\mathbb{E}((\nu'_i)^{-1})=\frac{n_i+1}{n_i+1-\lfloor(n_i+1)\lambda\rfloor}$ by Lemma C.4 and by the definition (17) of $\kappa$. Combining the latter with (22) gives the result.

### B.4 Proof of Theorem 2.5

Let us prove the result for the iid model (the proof for the conditional model is similar). Recall the definition of $(\hat{p}_{i,b})_{i\in[m]} = (\hat{p}_{i,b}^{(z_i^h)})_{i\in[m]}$ (not depending on $z^h$ for the iid model, see Algorithm 5), for $1 \leq b \leq B$ in Algorithm 1. Since the scores $S_{Y_i}(X_i)$, $i \in [n+m]$, are iid and have no ties, and $p$-values $(p_i)_{i\in[m]} = (p_i^{(Y_{n+i})})_{i\in[m]}$ involve only ranks between those scores, we have that the variables $(\hat{p}_{i,b})_{i\in[m]}$, $1 \leq b \leq B$, and $(p_i)_{i\in[m]}$ are iid. This means that $\xi_b = \xi_b^h$, $1 \leq b \leq B$, and $\xi := F((p_i)_{i\in[m]})$ are iid and thus exchangeable. Letting

$$\hat{q} = (B+1)^{-1}\Big(1 + \sum_{b=1}^{B} \mathbf{1}\{\xi_b \leq \xi\}\Big), \tag{23}$$

we thus have by Romano and Wolf (2005) that $\mathbb{P}(\hat{q} \leq \alpha) \leq \alpha$. Now, we have

$$\mathbb{P}((Y_{n+i})_{i\in[m]} \notin \mathcal{C}_{t,F}^{m}) = \mathbb{P}(F((p_i^{(Y_{n+i})})_{i\in[m]}) < \xi_{(\lfloor (B+1)\alpha)\rfloor)}) = \mathbb{P}(\xi < \xi_{(\lfloor (B+1)\alpha)\rfloor)}) = \mathbb{P}(\hat{q} \leq \alpha) \leq \alpha,$$

which concludes the proof.

## C Technical results

The next result is a variation of results in appendices of Marandon et al. (2024); Gazin et al. (2024b).

**Lemma C.1** (For full-calibrated $p$-values)**.** *Let us consider the scores $S_j = S_{Y_j}(X_j)$, $j \in [n+m]$, and assume them to be exchangeable and have no ties almost surely. Consider the full-calibrated $p$-values* (5) $p_i := p_i^{(Y_{n+i})}$*, $i \in [m]$, and let for any fixed $i \in [m]$,*

$$\begin{aligned} W_i &:= (A_i, (S_{n+j})_{j\in[m]\setminus\{i\}}); \\ A_i &:= \{S_j, j \in [n]\} \cup \{S_{n+i}\} =: \{a_{i,(1)}, \dots, a_{i,(n+1)}\}; \\ \Psi_i(u, W_i) &:= \left(\frac{1}{n+1}\Big(\mathbf{1}\{a_{i,(\lceil u(n+1)\rceil)} < S_{n+j}\} + \sum_{s\in A_i} \mathbf{1}\{s \geq S_{n+j}\}\Big)\right)_{j\in[m]\setminus\{i\}}, \end{aligned}$$

*with $a_{i,(1)} > \cdots > a_{i,(n+1)}$. Then we have*

*(i) $\mathbf{p}_{-i} := (p_j)_{j\in[m]\setminus\{i\}}$ is equal to $\Psi_i(p_i, W_i)$ and $u \in [0,1] \mapsto \Psi_i(u, W_i) \in \mathbb{R}^{m-1}$ is a nondecreasing function (in a coordinate-wise sense for the image space);*

*(ii) $(n+1)p_i$ is uniformly distributed on $[n+1]$ and independent of $W_i$;*

*(iii) the distribution of $\mathbf{p}_{-i}$ conditionally on $p_i = (n+1)^{-1}$ is the same as if all the scores were all iid $U(0,1)$. In particular, this distribution is equal to a distribution $\mathcal{D}_i$ which is defined as follows: $\mathbf{p}'_{-i} := (p'_j)_{j\in[m]\setminus\{i\}} \sim \mathcal{D}_i$ if, conditionally on the ordered statistics $U_{(1)} > \cdots > U_{(n+1)}$ of an iid sample of uniform random variables $(U_1, \dots, U_{n+1})$, the variables $(p'_j)_{j\in[m]\setminus\{i\}}$ are iid with common cdf $F(x) = (1 - U_{(\lfloor (n+1)x\rfloor + 1)})\mathbf{1}\{(n+1)^{-1} \leq x < 1\} + \mathbf{1}\{x \geq 1\}$.*

*(iv) Let $(p'_j)_{j\in[m]}$ such that $p'_i = (n+1)^{-1}$ and $p'_j = (n+1)^{-1}\sum_{s\in A_i} \mathbf{1}\{s \geq S_{n+j}\}$ for $j \neq i$. Then, $(p'_j)_{j\in[m]}$ is $W_i$-measurable and almost surely, for all $j \neq i$, $p'_j \leq p_j$ when $p_j \leq p_i$ and $p'_j = p_j$ when $p_j > p_i$.*

The next lemma adapts Lemma C.1 to the class conditional model (with class-calibrated $p$-values). In a nutshell, it says that the previous lemma applies within each class and uses the independence between scores of different classes (conditionally on all the labels).

**Lemma C.2** (For class-calibrated $p$-values)**.** *Let us consider the scores $S_j = S_{Y_j}(X_j)$, $j \in [n+m]$, and assume that for all $k \in [K]$, the scores $S_j$, $j \in [n+m] : Y_j = k$, are exchangeable, independent of the scores $S_j$, $j \in [n+m] : Y_j \neq k$ and have no ties almost surely. Consider the class-calibrated $p$-values* (5) $p_i := p_i^{(Y_{n+i})}$,

*$i \in [m]$, and let for any fixed $i \in [m]$, $n_i = |\mathcal{D}_{cal}^{(Y_{n+i})}|$ and*

$$\begin{aligned}
W_i &:= (A_i, (S_{n+j})_{j\in[m]\backslash\{i\}}, (S_j)_{j\in[n]:Y_j\neq Y_{n+i}});\\
A_i &:= \{S_j, j \in \mathcal{D}_{cal}^{(Y_{n+i})}\} \cup \{S_{n+i}\} = \{a_{i,(1)}, \dots, a_{i,(n_i+1)}\};\\
\Psi_i(u, W_i) &:= \left(\frac{1}{n_i+1}\Big(\mathbf{1}\{a_{i,(\lceil u(n_i+1)\rceil)} < S_{n+j}\} + \sum_{s\in A_i} \mathbf{1}\{s \geq S_{n+j}\}\Big)\right)_{j\in[m]\backslash\{i\}:Y_{n+j}=Y_{n+i}},
\end{aligned}$$

*with $a_{i,(1)} > \dots > a_{i,(n_i+1)}$. Then we have*

*(i) $(p_j)_{j\in[m]\backslash\{i\}:Y_{n+j}=Y_{n+i}}$ is equal to $\Psi_i(p_i, W_i)$ and $u \in [0,1] \mapsto \Psi_i(u, W_i)$ is a nondecreasing function (in a coordinate-wise sense for the image space);*

*(ii) Conditionally on $(Y_j)_{j\in[n+m]}$, the variable $(n_i+1)p_i$ is uniformly distributed on $[n_i+1]$ and independent of $W_i$ and $(p_j)_{j\in[m]:Y_{n+j}\neq Y_{n+i}}$;*

*(iii) the distribution of $(p_j)_{j\in[m]\backslash\{i\}:Y_{n+j}=Y_{n+i}}$ conditionally on $p_i = (n_i+1)^{-1}$ and $(Y_j)_{j\in[n+m]}$ is the same as if all the scores were all iid $U(0,1)$. In particular, this distribution is equal to a distribution $\mathcal{L}_{i,(Y_j)_{j\in[n+m]}}$ which is defined as follows: $(p'_j)_{j\in[m]\backslash\{i\}:Y_{n+j}=Y_{n+i}} \sim \mathcal{L}_{i,(Y_j)_{j\in[n+m]}}$ if, conditionally on the ordered statistics $U_{(1)} > \dots > U_{(n_i+1)}$ of an iid sample of uniform random variables $(U_1, \dots, U_{n_i+1})$ (independent of everything else), the variables $(p'_j)_{j\in[m]\backslash\{i\}:Y_{n+j}=Y_{n+i}}$ are iid with common cdf*

$$F(x) = (1 - U_{(\lfloor(n_i+1)x\rfloor+1)})\mathbf{1}\{(n_i+1)^{-1} \leq x < 1\} + \mathbf{1}\{x \geq 1\}.$$

*(iv) For $k \neq Y_{n+i}$, conditionally on $(Y_j)_{j\in[n+m]}$, the distribution of $(p_j)_{j\in[m]:Y_{n+j}=k}$ is the same as if all the scores were all iid $U(0,1)$. In particular, this distribution is equal to a distribution $\mathcal{L}^{(k)}_{(Y_j)_{j\in[n+m]}}$ which is defined as follows: $(p'_j)_{j\in[m]:Y_{n+j}=k} \sim \mathcal{L}^{(k)}_{(Y_j)_{j\in[n+m]}}$ if, conditionally on the ordered statistics $U^{(k)}_{(1)} > \dots > U^{(k)}_{(n_k)}$ of an iid sample of uniform random variables $(U^{(k)}_1, \dots, U^{(k)}_{n_k})$ (independent of everything else), the variables $(p'_j)_{j\in[m]:Y_{n+j}=k}$ are iid with common cdf*

$$F^{(k)}(x) = (1 - U^{(k)}_{(\lfloor(n_k+1)x\rfloor)})\mathbf{1}\{(n_k+1)^{-1} \leq x < 1\} + \mathbf{1}\{x \geq 1\}.$$

*(v) Let $(p'_j)_{j\in[m]:Y_{n+j}=Y_{n+i}}$ such that $p'_i = (n_i+1)^{-1}$ and $p'_j = (n_i+1)^{-1}\sum_{s\in A_i}\mathbf{1}\{s \geq S_{n+j}\}$ for $j \neq i$ with $Y_{n+j} = Y_{n+i}$. Then, $(p'_j)_{j\in[m]:Y_{n+j}=Y_{n+i}}$ is $W_i$-measurable and almost surely, for all $j \neq i$ with $Y_{n+j} = Y_{n+i}$, $p'_j \leq p_j$ when $p_j \leq p_i$ and $p'_j = p_j$ when $p_j > p_i$.*

*Proof.* Let us prove (i), we have for $j \in [m]\backslash\{i\}$ with $Y_{n+j} = Y_{n+i}$,

$$\begin{aligned}
p_j &= \frac{1}{|\mathcal{D}_{\text{cal}}^{(Y_{n+j})}|+1}\Big(1 + \sum_{\ell\in\mathcal{D}_{\text{cal}}^{(Y_{n+j})}} \mathbf{1}\{S_\ell \geq S_{n+j}\}\Big)\\
&= \frac{1}{|\mathcal{D}_{\text{cal}}^{(Y_{n+i})}|+1}\Big(1 + \sum_{s\in A_i} \mathbf{1}\{s \geq S_{n+j}\} - \mathbf{1}\{S_{n+i} \geq S_{n+j}\}\Big), \qquad (24)
\end{aligned}$$

which gives the relation because $S_{n+i} = a_{i,(p_i(n_i+1))}$. Since the monotonicity property is clear, this gives (i).

Point (ii) comes from the fact that the scores $\{S_j, j \in \mathcal{D}_{\text{cal}}^{(Y_{n+i})}\} \cup \{S_{n+i}\}$ have not ties and are exchangeable conditionally on all other scores (and of $(Y_j)_{j\in[n+m]}$).

For proving (iii), we first note that the calibrated $p$-values are ranks of exchangeable scores with not ties. Hence, the distribution of the $p$-value vector is free from the distribution scores and thus is the same as if the scores were generated as iid $U(0,1)$. Hence, the latter assumption is made for the rest of the proof. Now, by (i), we have for all $j \in [m]\backslash\{i\}$ with $Y_{n+j} = Y_{n+i}$, and if $p_i = (n_i+1)^{-1}$,

$$p_j = \frac{1}{n_i+1}\Big(1 + \sum_{s\in A_i\backslash\{a_{i,(1)}\}} \mathbf{1}\{s \geq S_{n+j}\}\Big),$$

because $a_{i,(1)} = S_{n+i}$ in that case. Hence, the $p_j$'s, $j \in [m]\backslash\{i\}$ with $Y_{n+j} = Y_{n+i}$, are iid conditionally on $A_i$ and $(Y_j)_{j\in[n+m]}$. In addition, the common marginal cdf at a point $x$ is given by

$$\mathbb{P}\Big(1 + \sum_{s\in A_i\backslash\{a_{i,(1)}\}} \mathbf{1}\{s \geq S_{n+j}\} \leq x(n_i+1)\Big) = \mathbb{P}\Big(\sum_{s\in A_i\backslash\{a_{i,(1)}\}} \mathbf{1}\{s \geq S_{n+j}\} < \lceil x(n_i+1)\rceil\Big)$$
$$= \mathbb{P}\Big(a_{i,(\lceil x(n_i+1)\rceil+1)} < S_{n+j}\Big),$$

provided that $1 \leq x(n_i+1) < n_i+1$ and the above probabilities being taken conditionally on $A_i$ and $(Y_j)_{j\in[n+m]}$. The result follows because we considered uniformly distributed scores.

Point (iv) is similar to point (iii), starting directly from the following relation: for all $j \in [m]$ with $Y_{n+j} = k$,

$$p_j = \frac{1}{n_k+1}\Big(1 + \sum_{s\in\{U^{(k)}_{(1)},\dots,U^{(k)}_{(n_k)}\}} \mathbf{1}\{s \geq S_{n+j}\}\Big),$$

where $U^{(k)}_1 > \cdots > U^{(k)}_{n_k}$ are the ordered elements of $\{S_j, j \in \mathcal{D}^{(k)}_{\text{cal}}\}$.

Finally, let us prove point (v): first $p'_j \leq p_j$ is obvious from (24). Second, if $j \in [m]\backslash\{i\}$ with $Y_{n+j} = Y_{n+i}$ is such that $p_j > p_i$, this means $S_{n+j} < S_{n+i}$ and thus $p'_j = p_j$ from (24). The result is proved. □

**Lemma C.3** (Lemma D.6 of Marandon et al. (2024)). *Write $\widehat{\ell} = \widehat{\ell}(\mathbf{p})$ for (20) with any p-value family $\mathbf{p} = (p_i)_{i\in[m]}$. Fix any $i \in \{1,\dots,m\}$ and consider two collections $\mathbf{p} = (p_i)_{i\in[m]}$ and $\mathbf{p}' = (p'_i)_{i\in[m]}$ which satisfy almost surely that*

$$\forall j \in [m], \begin{cases} p'_j \leq p_j & \text{if } p_j \leq p_i; \\ p'_j = p_j & \text{if } p_j > p_i. \end{cases} \tag{25}$$

*Then we have almost surely $\{p_i \leq \alpha\widehat{\ell}(\mathbf{p})/m\} = \{p_i \leq \alpha\widehat{\ell}(\mathbf{p}')/m\} \subset \{\widehat{\ell}(\mathbf{p}) = \widehat{\ell}(\mathbf{p}')\}$.*

**Lemma C.4.** *For $V_{(1)} > \cdots > V_{(\ell)}$ the order statistics of $\ell$ iid uniform variables on $[0,1]$, we have for all $a \in [\ell]$, $V_{(a)} \sim \beta(\ell+1-a, a)$. In addition, if $a < \ell$, $\mathbb{E}(1/V_{(a)}) = \ell/(\ell-a)$.*

**Lemma C.5** (Klenke and Mattner (2010)). *For $Z_1,\dots,Z_m$ independent Bernoulli variables of respective parameters $\nu_i \in [0,1]$, $i \in [m]$, the Poisson binomial variable $\sum_{i\in[m]} Z_i$ is stochastically larger than a binomial variable of parameters $m$ and $\nu = \prod_{i\in[m]} \nu_i^{1/m}$.*

**Lemma C.6** (Lemma 1 of Benjamini et al. (2006)). *If $T$ is a Binomial variable with parameter $m-1 \geq 0$ and $\nu \in (0,1]$, we have*

$$\mathbb{E}[1/(T+1)] = (1-(1-\nu)^m)/(m\nu) \leq 1/(m\nu).$$

The next lemma has been suggested by an anonymous referee. Recall the definition of PRDS given in § B.1.

**Lemma C.7** (Lemma A.2 of Bogomolov (2023)). *Let $(p_i)_{i\in[m]}$ be a p-value family, $(G_k)_{k\in[K]}$ a partition of $[m]$ and assume the following:*

- *Independence between groups: for $k \neq k'$, $(p_i)_{i\in G_k}$ is independent of $(p_i)_{i\in G_{k'}}$;*
- *PRDS inside each group: for $k \in [m]$, $(p_i)_{i\in G_k}$ is a p-value family which is PRDS on $G_k$.*

*Then the p-value family $(p_i)_{i\in[m]}$ is PRDS on $[m]$.*

We provide a proof for completeness.

*Proof.* Fix $i \in [m]$ and a nondecreasing set $D \subset [0,1]^m$, and prove that the function $u \mapsto \mathbb{P}((p_i)_{i\in[m]} \in D \,|\, p_i = u)$ is nondecreasing. Denote $k_i$ the unique $k$ such that $i \in G_k$ and let

$$D_{k_i,(p_j)_{j\notin G_{k_i}}} = \{(p_j)_{j\in G_{k_i}} \in [0,1]^{G_{k_i}} \;:\; (p_j)_{j\in[m]} \in D\}$$

which is clearly a nondecreasing (measurable) set of $[0,1]^{G_{k_i}}$. By using the two assumptions, we have that $(p_i)_{i\in G_{k_i}}$ is PRDS on $G_k$ conditionally on $(p_j)_{j\notin G_{k_i}}$. Hence,

$$u \mapsto \mathbb{P}((p_j)_{j\in G_{k_i}} \in D_{k_i,(p_j)_{j\notin G_{k_i}}} \mid p_i = u, (p_j)_{j\notin G_{k_i}})$$

is nondecreasing. We obtain the result by integrating with respect to $(p_j)_{j\notin G_{k_i}}$. □

# D Computational shortcut for the combinations of conformal $p$-values method

Computing the batch prediction set for our methods is in general of complexity of order $K^m$ times the cost of computing the combining function (e.g., order $m$ for Fisher, or $m\log m$ for Simes or adaptive Simes)[5]. The aim of this section is to reduce this complexity when the user only want to report lower/upper bounds for $m_k(Y)$, $k\in[K]$ (3). We also discuss the issue of reconstructing the batch prediction set from these bounds.

## D.1 Shortcut for computing the bounds

Naively computing the bounds $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$, $k\in[K]$, in (4), which are derived from the Simes conformal prediction set in (8) or its adaptive version in (12), results in an exponential complexity of $O(K^m)$. This quickly becomes impractical for large batch sizes. To address this issue, we introduce a novel shortcut that allows for a more efficient computation of these bounds, with a computational complexity of at most $O(K\times m^2)$.

This shortcut applies to both the full-calibrated and class-calibrated conformal $p$-values. Proposition D.1 shows that it is exact when $K=2$ and the scores produced by the machine learning model are probabilities. However, when $K>2$ or when arbitrary scores are used, the shortcut becomes conservative, potentially yielding wider bounds but never narrower ones. This ensures that the coverage guarantee of at least $1-\alpha$ probability is maintained.

Algorithm 3 provides the pseudocode for the shortcut to compute the bounds $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$ derived from the (adaptive) Simes conformal prediction set.

**Algorithm 3:** Shortcut for computing the bounds $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$, $k\in[K]$, with (adaptive) Simes predition set.

```
Input: Full-calibrated or class-calibrated conformal p-values $(p_i^{(k)})_{i\in[m],k\in[K]}$, level $\alpha\in(0,1)$, an estimator
       $\hat m_0(\mathbf{p})$ that is monotone in the p-values $\mathbf{p}=(p_i)_{i\in[m]}$.
for each $k\in[K]$ do
    Sort $(p_i^{(k)})_{i\in[m]}$ in decreasing order and store as $a_1\geq\ldots\geq a_m$;
    Sort $(\max\{p_i^{(j)}, j\neq k\})_{i\in[m]}$ in decreasing order and store as $b_1\geq\ldots\geq b_m$;
    for each $v\in\{m,\ldots,0\}$ do
        $(q_1,\ldots,q_m)\leftarrow(a_1,\ldots,a_v,b_1,\ldots,b_{m-v})$;
        Sort $(q_i)_{i\in[m]}$ in increasing order and store as $q_{(1)}\leq\ldots\leq q_{(m)}$;
        $h_{v,k}\leftarrow\min\left(\frac{\hat m_0(\mathbf{q})}{\ell}q_{(\ell)}, \ell\in[m]\right)$
    end
    $\ell_\alpha^{(k)}\leftarrow\min(v\in\{0,\ldots,m\}: h_{v,k}>\alpha)$;
    $u_\alpha^{(k)}\leftarrow\max(v\in\{0,\ldots,m\}: h_{v,k}>\alpha)$;
end
Output: $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$, $k\in[K]$
```

**Proposition D.1.** *For any $\alpha\in(0,1)$, let $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$, $k\in[K]$ be the bounds defined by (4), derived from the Simes prediction sets in (8) or its adaptive version in (12). Algorithm 3 returns the bounds $[\tilde\ell_\alpha^{(k)}, \tilde u_\alpha^{(k)}]$ such that $\tilde\ell_\alpha^{(k)}\leq\ell_\alpha^{(k)}$ and $\tilde u_\alpha^{(k)}\geq u_\alpha^{(k)}$ for all $k\in[K]$, with a computational complexity of at most $O(K\times m^2)$. In addition, when $K=2$ and the scores produced by the machine learning model are probabilities, i.e., $S_k(x_{n+i}) = 1-S_{3-k}(x_{n+i})$ for $k\in\{1,2\}$ and $i\in[m]$, it holds that $\tilde\ell_\alpha^{(k)}=\ell_\alpha^{(k)}$ and $\tilde u_\alpha^{(k)}=u_\alpha^{(k)}$ for all $k\in[K]$.*

[5] In general, the cost of computing the $p$-value family $(p_i^{(k)}, k\in[K], i\in[m])$ is negligible wrt $K^m$.

*Proof.* First, let us establish that the time complexity of the algorithm is $O(K \times m^2)$. To produce the sorted concatenation of two sorted vectors $a_1, \dots, a_{m-i}$ and $b_1, \dots, b_i$ takes linear time, i.e. $O(m)$. This merging process, which generates the sorted concatenation, is repeated $m+1$ times for each $k$. As a result, for each $k$, this step contributes $O(m^2)$, leading to an overall complexity of $O(K \times m^2)$.

We first discuss the case where $\hat{m}_0 = m$, meaning the estimator is the constant $m$. Let $\mathbf{p} = (p_i)_{i \in [m]}$ denotes a vector of $p$-values, with the sorted values represented as $p_{(1)} \le \ldots \le p_{(m)}$. Simes' test is defined as $F_{\text{Simes}}(\mathbf{p}) = \min\left(\frac{m}{\ell} p_{(\ell)}, \ell \in [m]\right)$. This test is monotonic, meaning that if $\mathbf{p} \le \mathbf{q}$ componentwise (i.e. $p_{(i)} \le q_{(i)}$ for all $i \in [m]$), then $F_{\text{Simes}}(\mathbf{p}) \le F_{\text{Simes}}(\mathbf{q})$.

By definition, $v \notin \mathcal{N}_k(\mathcal{C}^m_{\alpha,\text{Simes}})$ if $F_{\text{Simes}}(\mathbf{p}(y)) \le \alpha$ for all $y \in [K]^m$ such that $m_k(y) = v$, for any $v \in \{0, \dots, m\}$.

Then, for some $\mathbf{q} = (q_i)_{i \in [m]}$ with $\mathbf{q} \ge \mathbf{p}(y)$ for all $y \in [K]^m$ such that $m_k(y) = v$, $F_{\text{Simes}}(\mathbf{q}) \le \alpha$ implies $v \notin \mathcal{N}_k(\mathcal{C}^m_{\alpha,\text{Simes}})$. However, $F_{\text{Simes}}(\mathbf{q}) > \alpha$ does not necessarily imply $v \in \mathcal{N}_k(\mathcal{C}^m_{\alpha,\text{Simes}})$.

Given $k$ and $v$, Algorithm 3 identifies a suitable vector $\mathbf{q} = \mathbf{q}_{v,k}$ such that $\mathbf{q} \ge \mathbf{p}(y)$ for all $y \in [K]^m$ where $m_k(y) = v$. Then we let

$$\tilde{\mathcal{N}}_k = \{v \in \{0, \dots, m\} \,:\, F_{\text{Simes}}(\mathbf{q}_{v,k}) > \alpha\},$$

which ensures $\tilde{\mathcal{N}}_k \supseteq \mathcal{N}_k(\mathcal{C}^m_{\alpha,\text{Simes}})$ The resulting bounds are given by $[\tilde{\ell}^{(k)}_\alpha, \tilde{u}^{(k)}_\alpha] = [\min\tilde{\mathcal{N}}_k, \max\tilde{\mathcal{N}}_k]$, which guarantees that $\tilde{\ell}^{(k)}_\alpha \le \ell^{(k)}_\alpha$ and $\tilde{u}^{(k)}_\alpha \ge u^{(k)}_\alpha$ for every $k \in [K]$.

We now need to demonstrate that Algorithm 3 produces a vector $\mathbf{q}$ such that $\mathbf{q} \ge \mathbf{p}(y)$ for all $y \in [K]^m$ such that $m_k(y) = v$.

For any $y \in [K]^m$ such that $m_k(y) = v$, the vector $\mathbf{p}(y)$ consists of $v$ conformal $p$-values $p^{(k)}_{i_1}, \dots, p^{(k)}_{i_v}$ and $m - v$ conformal $p$-values $p^{(j_1)}_{i_{v+1}}, \dots, p^{(j_{m-v})}_{i_m}$, where $i_1, \dots, i_m$ is a permutation of $[m]$ and $j_1, \dots, j_{m-v} \in [K] \setminus \{k\}$. If we consider the vector $\mathbf{p}(\tilde{y})$, which is formed by $p^{(k)}_{i_1}, \dots, p^{(k)}_{i_v}$ and the maximum values $\max(p^{(j)}_{i_{v+1}}, j \ne k), \dots, \max(p^{(j)}_{i_{v+m}}, j \ne k)$, we can conclude that $\mathbf{p}(\tilde{y}) \ge \mathbf{p}(y)$. Since the vector $\mathbf{q}$ in Algorithm 3 is constructed using the largest $v$ values from $(p^{(k)}_i)_{i \in [m]}$ and the largest $m - v$ values from $(\max(p^{(j)}_i, j \ne k))_{i \in [m]}$, it follows that $\mathbf{q} \ge \mathbf{p}(\tilde{y}) \ge \mathbf{p}(y)$ for all $y \in [K]^m$ such that $m_k(y) = v$. This establishes the conservativeness of the shortcut for $K \ge 2$ and for any scores produced by the machine learning model.

If $K = 2$ and the scores produced by the machine learning model are probabilities, then we have the relationship $S_k(x_{n+i}) = 1 - S_{3-k}(x_{n+i})$ for $k \in \{1, 2\}$ and $i \in [m]$. Given this relationship, there exists a permutation $i_1, \dots, i_m$ such that the sequence $S_k(x_{n+i_{j_1}}) \le \ldots \le S_k(x_{n+i_{j_m}})$ is nondecreasing, while the sequence $S_{3-k}(x_{n+i_{j_1}}) \ge \ldots \ge S_{3-k}(x_{n+i_{j_m}})$ is nonincreasing. Consequently, the ranks of $S_k(x_{n+j_1}), \dots, S_k(x_{n+j_m})$ within the set $(S_{y_j}(x_j))_{j \in \mathcal{D}^{(k)}_{\text{cal}}}$ will be nondecreasing, while the ranks of $S_{3-k}(x_{n+j_1}), \dots, S_{3-k}(x_{n+j_m})$ within the set $(S_{y_j}(x_j))_{j \in \mathcal{D}^{(3-k)}_{\text{cal}}}$ will be nonincreasing. Since these ranks are proportional to the conformal $p$-values, it follows that $p^{(k)}_{i_1} \le \ldots \le p^{(k)}_{i_m}$ and $p^{(3-k)}_{i_1} \ge \ldots \ge p^{(3-k)}_{i_m}$.

Consider $y \in [K]^m$ such that $m_k(y) = v$. Let the vector $\mathbf{p}(y^*)$ consist of the $v$ largest values from $(p^{(k)}_i)_{i \in [m]}$, specifically $p^{(k)}_{i_{m-v+1}}, \dots, p^{(k)}_{i_m}$. Consequently, the remaining $m - v$ values in $\mathbf{p}(y^*)$ are $p^{(3-k)}_{i_1}, \dots, p^{(3-k)}_{i_{m-v}}$, i.e. the largest $m - v$ values from $(p^{(3-k)}_i)_{i \in [m]}$. Thus, we have $\mathbf{p}(y^*) \ge \mathbf{p}(y)$ for all $y \in [K]^m$ such that $m_k(y) = v$. Furthermore, by construction, $\mathbf{q}$ in Algorithm 3 is equal to $\mathbf{p}(y^*)$. Therefore $F_{\text{Simes}}(\mathbf{q}) \le \alpha$ if and only if $F_{\text{Simes}}(\mathbf{p}(y)) \le \alpha$ for all $y \in [K]^m$ such that $m_k(y) = v$. This establishes the exactness of the shortcut when $K = 2$ and $S_k(x_{n+i}) = 1 - S_{3-k}(x_{n+i})$ for $k \in \{1, 2\}$ and $i \in [m]$.

The validity of the shortcut for the adaptive version of Simes follows from the required monotonicity of the estimator: if $\mathbf{p}(y) \le \mathbf{q}$, then $\hat{m}_0(\mathbf{p}(y)) \le \hat{m}_0(\mathbf{q})$ holds for any $y \in [K]^m$. This, combined with $F_{\text{A-Simes}}(\mathbf{p}(y)) \le \alpha$ if and only if $F_{\text{Simes}}(\mathbf{p}(y)) \le m\alpha / \hat{m}_0(\mathbf{p}(y))$ yields the desired result. □

### D.2 Extension to other combining functions

Algorithm 4 presents a more general approach for any $p$-value vector combining function $F(\mathbf{p})$, which is symmetric and monotone in the $p$-values $\mathbf{p} = (p_i)_{i \in [m]}$. It requires the empirical threshold $t = \xi_{(\lfloor (B+1)\alpha \rfloor)}$ from Theorem

2.5, which depends on $(m_k)_{k\in[K]}$ in the conditional model, i.e. $t = t(\alpha, (m_k)_{k\in[K]})$. The proof that Algorithm 4 yields conservative yet valid bounds is analogous to the previous result and is therefore omitted.

**Algorithm 4:** General shortcut for computing the bounds $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$, $k \in [K]$.

```
Input: Full-calibrated or class-calibrated conformal p-values $(p_i^{(k)})_{i\in[m],k\in[K]}$, level $\alpha \in (0,1)$, p-value vector
       combining function $F(\mathbf{p})$ that is symmetric and monotone in the p-values $\mathbf{p} = (p_i)_{i\in[m]}$ and the
       corresponding critical value $t = t(\alpha, (m_k)_{k\in[K]})$.
for each $k \in [K]$ do
    Sort $(p_i^{(k)})_{i\in[m]}$ in decreasing order and store as $a_1 \geq \ldots \geq a_m$;
    Sort $(\max\{p_i^{(j)}, j \neq k\})_{i\in[m]}$ in decreasing order and store as $b_1 \geq \ldots \geq b_m$;
    for each $v \in \{m, \ldots, 0\}$ do
        $(q_1, \ldots, q_m) \leftarrow (a_1, \ldots, a_v, b_1, \ldots, b_{m-v})$;
        Sort $(q_i)_{i\in[m]}$ in increasing order and store as $q_{(1)} \leq \ldots \leq q_{(m)}$;
        $h_{v,k} \leftarrow \mathbf{1}\{F(\mathbf{q}) \geq \min\{t(\alpha, m_k = v, m_j), j \neq k\}\}$
    end
    $\ell_\alpha^{(k)} \leftarrow \min(v \in \{0, \ldots, m\} : h_{v,k} > 0)$;
    $u_\alpha^{(k)} \leftarrow \max(v \in \{0, \ldots, m\} : h_{v,k} > 0\}$;
end
Output: $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$, $k \in [K]$
```

### D.3 Batch prediction set reconstruction from the bounds

As described in the previous subsections, from the bounds $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$, $k \in [K]$, it is straightforward to produce a conservative batch prediction set $\tilde{\mathcal{C}}_\alpha^m$ such that $\tilde{\mathcal{C}}_\alpha^m \supseteq \mathcal{C}_\alpha^m$. The cardinality of the conservative set $\tilde{\mathcal{C}}_\alpha^m$ is the sum of all valid assignments of $(m_1, \ldots, m_K)$ occurrences, where $\ell_\alpha^{(k)} \leq m_k \leq u_\alpha^{(k)}$ for each $k \in \{1, \ldots, K\}$, and $m_1 + \cdots + m_K = m$, with each valid assignment counted by the multinomial coefficient $\binom{m}{m_1, m_2, \ldots, m_K}$:

$$|\tilde{\mathcal{C}}_\alpha^m| = \sum_{\substack{(m_1,\ldots,m_K)\,:\,\sum_{k=1}^K m_k = m, \\ \ell_\alpha^{(k)} \leq m_k \leq u_\alpha^{(k)}\ \forall k\in[K]}} \binom{m}{m_1, m_2, \ldots, m_K}.$$

For the reading zip code example, from Table 1, we derive the bounds $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$ with $\alpha = 0.05$, which are as follows:

$$[1,2],\ [0,0],\ [0,0],\ [0,0],\ [1,1],\ [0,2],\ [0,2],\ [0,0],\ [0,1],\ [0,0] \quad \text{for} \quad k = 1, \ldots, 10.$$

The assignments $(m_1, \ldots, m_{10})$ that satisfy $m_1 + \ldots + m_{10} = 5$ and $\ell_\alpha^{(k)} \leq m_k \leq u_\alpha^{(k)}$ for each $k \in \{1, \ldots, 10\}$ are ten:

$$\begin{array}{lll} (1,0,0,0,1,0,2,0,1,0), & (1,0,0,0,1,1,1,0,1,0), & (1,0,0,0,1,1,2,0,0,0), \\ (1,0,0,0,1,2,0,0,1,0), & (1,0,0,0,1,2,1,0,0,0), & (2,0,0,0,1,0,1,0,1,0), \\ (2,0,0,0,1,0,2,0,0,0), & (2,0,0,0,1,1,0,0,1,0), & (2,0,0,0,1,1,1,0,0,0), \\ (2,0,0,0,1,2,0,0,0,0). & & \end{array}$$

The corresponding multinomial coefficients are 60, 120, 60, 60, 60, 60, 30, 60, 60 and 30, respectively. This results in a cardinality of the conservative set $|\tilde{\mathcal{C}}_{\alpha,\mathrm{Simes}}^m| = 600$, compared to $|\mathcal{C}_{\alpha,\mathrm{Simes}}^m| = 6$ given in Table 1. This indicates that reconstructing the prediction set solely from the bounds is quite imprecise. For instance, the assignment $(2,0,0,0,1,2,0,0,0,0)$ corresponds to $\binom{5}{2,0,0,0,1,2,0,0,0,0} = 30$ vectors of size 5, which include two 0s, one 4, and two 5s.

While $\tilde{\mathcal{C}}_\alpha^m$ is not accurate in general, we can combine this information with individual conformal prediction sets $\mathcal{C}_{i,\alpha}^m$, $i \in [m]$ to allows for a more accurate batch prediction set reconstructed from the bounds. For this, specific shortcuts can be investigated to compute the individual conformal prediction sets $\mathcal{C}_{i,\alpha}^m$, $i \in [m]$. More specifically, for Simes' method, we can always use the Bonferroni individual prediction set to obtain a new batch prediction set from the bounds *both with low complexity that can only improve over $\mathcal{C}^m_{\alpha,Bonf}$*. In addition, the following example shows that this improvement can be strict: we see this as an important 'proof of concept'.

For the example of one batch of the CIFAR dataset given in Figure 6 with $m = 10$, $K = 3$, and $\alpha = 0.1$, the Bonferroni individual conformal prediction sets $\mathcal{C}^m_{i,\alpha}$ are $\{3\}$ for $i = 8$ and $\{1, 2, 3\}$ for $i = 1, 2, 3, 4, 5, 6, 7, 9, 10$. On the other hand, the Simes bounds $[\ell^{(k)}_\alpha, u^{(k)}_\alpha]$ are $[0, 8]$, $[0, 9]$, and $[1, 10]$ for $k = 1, 2, 3$, which improve upon Bonferroni's $[0, 9]$, $[0, 9]$, and $[1, 10]$. Consequently, the vector $(1, 1, 1, 1, 1, 1, 1, 3, 1, 1)$ must be excluded from $\mathcal{C}^m_{\alpha,\text{Bonf}}$ because it violates the constraint that the number of 1s must not exceed 8.

### D.4 Simulation results for large batches of test points

In order to demonstrate the feasibility and usefulness of the shortcut, we carried out simulations with test samples of size $m \in \{200, 2000\}$. Specifically, we considered the Gaussian multivariate setting described in § 4.1 with classes one and two only. The calibration sample has an equal number of examples from each of the two classes.

Table 4 shows results for the case that the test sample has an equal number of examples from each of the two classes. Since the calibration set in each class is 400 examples, the smallest possible class conditional conformal $p$-value is 1/401, so the Bonferroni adjusted $p$-value is at least $m \cdot 1/401$ and the lower bounds are zero. However, Simes and modified Simes have informative lower bounds, and the tightness of the lower bounds increases with the SNR. The computational complexity is very reasonable, running in less than 0.01 seconds for $m = 200$, and in 0.20-0.22 seconds for $m = 2000$.

| | $m = 200$ | | | $m = 2000$ | | |
|---|---|---|---|---|---|---|
| | | | Storey- | | | Storey- |
| SNR | Bonf | Simes | Simes | Bonf | Simes | Simes |
| 1.00 | 0.00 | 1.12 | 2.13 | 0.00 | 9 | 20 |
| 1.50 | 0.00 | 6.98 | 10.56 | 0.00 | 62 | 103 |
| 2.00 | 0.00 | 23.31 | 26.70 | 0.00 | 212 | 253 |
| 2.50 | 0.00 | 40.73 | 41.85 | 0.00 | 413 | 428 |
| 3.00 | 0.00 | 58.02 | 57.60 | 0.00 | 729 | 730 |
| 3.50 | 0.00 | 73.28 | 72.85 | 0.00 | 830 | 829 |
| 4.00 | 0.00 | 84.19 | 83.83 | 0.00 | 830 | 829 |
| 4.50 | 0.00 | 90.75 | 90.22 | 0.00 | 897 | 896 |

Table 4: The average lower bound for class one at each SNR, for $m = 200$ (columns 2–4) and for $m = 2000$ (columns 5–7), at level $\alpha = 0.1$, for the following $p$-value combining functions: Bonferroni, Simes, and adaptive Simes using Storey's estimator (see detailed data generation in text). The fraction of test sample examples from class one is half. Based on 100 simulations.

Table 5 provides results when the distribution of the classes is uneven in the test sample. As in the setting of Table 4, Bonferroni's combination method provides only trivial lower bounds so it is not shown. We also omit adaptive Simes since the performance is very similar to that of using Simes combining function. With Simes combining function, we see that as the fraction of test samples from class two increases or the signal strengthens, the probability of detecting that there are examples from class two increases and the expected lower bound increases. The detection of a lower bound being positive is important in many application. For example, in ecology, this is proof that an animal population is not extinct in an area. In medicine, the detection of evidence that a treatment can be positive (class one) in some patients but negative (class two) in other patients suggests a qualitative interaction that can prompt further investigation.

## E General $p$-value combining prediction set algorithm for the iid model

In § 2.5 we provided the most general method of obtaining $1 - \alpha$ level prediction sets using combinations of conformal $p$-values. Algorithm 1 shows the construction for the class conditional model. Its computational complexity is $B$ times the number of unique frequency distributions of $[K]^m$ vectors. For completeness, we provide in Algorithm 5 an algorithm for the iid model, which requires only $B$ permutations.

| | Fraction in test sample | Average lower bound | | Probability of non-zero lower bound | |
|---|---|---|---|---|---|
| SNR | from class one | class 1 | class 2 | class 1 | class 2 |
| 2 | 1 | 61.9 | 0.004 | 1.000 | 0.002 |
| | 0.95 | 57.9 | 0.258 | 1.000 | 0.1334 |
| | 0.9 | 54.0 | 1.225 | 1.000 | 0.4242 |
| | 0.7 | 37.39 | 10.12 | 1.000 | 0.9909 |
| 3 | 1 | 133.6 | 0.008 | 1.000 | 0.004 |
| | 0.95 | 126.3 | 2.028 | 1.000 | 0.680 |
| | 0.9 | 118.3 | 7.242 | 1.000 | 0.9756 |
| | 0.7 | 88.05 | 31.99 | 1.000 | 1.000 |
| 4 | 1 | 175.7 | 0.006 | 1.000 | 0.003 |
| | 0.95 | 166.3 | 4.828 | 1.000 | 0.9832 |
| | 0.9 | 157.0 | 13.24 | 1.000 | 0.9999 |
| | 0.7 | 119.9 | 47.88 | 1.000 | 1.000 |

Table 5: The average lower bound (columns 2 and 3) and probability that the lower bound is non-trivial (columns 4 and 5) using the Simes combination function, for each class at SNR=3, for $m = 200$, at level $\alpha = 0.1$. For each SNR, each row has a different relative frequency of the number of examples from class one (in the calibration set, half the examples are from class one, see text for details). Based on 10000 simulations.

**Algorithm 5:** Constructing a $1-\alpha$ level batch prediction set, using combinations of conformal $p$-values, for the iid model

```
Input: Number of examples in the calibration set $n$; combining function $F$; level $\alpha \in (0,1)$; number of permutations $B$; conformal $p$-values $(p_i^{(y_i)})_{i\in[m]}$.
for each $b \in [B]$ do
    Generate a random permutation $\pi_b$ of $[n+m]$;
    Compute null conformal $p$-values:
        $$\hat{p}_{i,b} \leftarrow \frac{1+\sum_{j=1}^{n} \mathbf{1}\{\pi_b(j) \geq \pi_b(n+i)\}}{n+1}$$
     for $i \in [m]$;
    Compute combined statistic:
    $\xi_b \leftarrow F((\hat{p}_{i,b}, i \in [m]))$;
end
Compute threshold:
$t \leftarrow \xi_{(\lfloor (B+1)\alpha \rfloor)}$,
where $\xi_{(1)} \leq \ldots \leq \xi_{(B)}$ are the ordered test statistics and $\xi_{(0)} = -\infty$;
Construct batch prediction set:
$\mathcal{C}^m_{t,F} \leftarrow \{y = (y_i)_{i\in[m]} \in [K]^m \ : \ F((p_i^{(y_i)})_{i\in[m]}) \geq t\}$;
Output: Batch prediction set $\mathcal{C}^m_{t,F}$.
```

# F Additional numerical experiments

## F.1 Gaussian multivariate setting

We provide more results for the data generation described in § 4.1. Figure 4 shows the data available in one data generation. Table 6 shows the batch prediction set for this batch using Bonferroni at $\alpha = 0.1$, as well as the Bonferroni and Simes $p$-values for each $y$ in the batch. Had the analyst used Simes instead of Bonferroni at $\alpha = 0.1$, the batch prediction set size would have been 25% smaller.

Table 7 adds the *median* and the *oracle* adaptive Simes procedure, that uses respectively (18) with $\ell = \lceil m/2 \rceil$ and $\hat{m}_0(y) = m_0(y)$ as estimator, to the comparison in Table 2. It also provides the estimated non-coverage for each method. Using oracle adaptive Simes is by far the best, but this is not a practical method since $m_0(y)$ is unknown.

Table 8 provides the average sum of lower and upper bounds for the three classes by the different methods. The

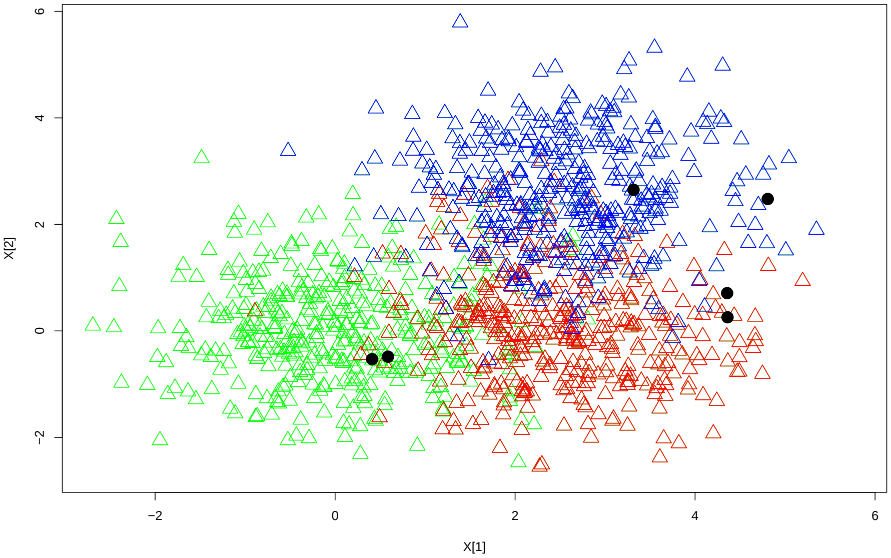


Figure 4: Illustration of one data generation with SNR = 2.5. The batch of six test samples are in black. There are 400 calibration examples from each class (class one in green, class two in red, and class three in blue). At $\alpha = 0.1$, the size of the prediction set using Bonferroni and Simes is 32 and 24, respectively.

goal in the comparisons in this table are two fold. First, to assess how conservative the shortcut suggested in § D for computational efficiency is. Using Simes (columns 3 and 4), it appears that the shortcut produces almost the same exact bounds for low SNR, and the inflation (i.e., smaller lower bounds and higher upper bounds with the shortcut) for high SNR is tiny. Using adaptive Simes (columns 6 and 7), it appears that there is a light inflation for all SNRs, and it is larger than using Simes. The second goal is to compare the efficiency of each combining method. For this purpose, we also provide Table 5 that includes the estimated LRT but is based on a smaller number of simulations (since the bounds take 100 times longer to compute with the estimated LRT). As expected, the bounds using Simes are tighter than using Bonferroni, but the advantage is small. A more pronounced difference is with respect to oracle Simes, but it is not a practical method since $m_0(y)$ is unknown in practice. The bounds using Fisher is worse than other methods for SNR $\geq 2.5$, and better for the upper bound if SNR $\leq 2$. The bounds using the estimated LRT tend to be the tightest among the practical methods considered.

## F.2 USPS and CIFAR data sets

To obtain a visualization different from the one of § 4.2, Figure 5 displays the averaged size of batch prediction sets as well as an estimation of the coverage in function of $\alpha$ in the same setting as Figure 2. The conclusions are analogue. The fluctuations of the coverage around $\alpha$ in Figure 5 for the Bonferroni, Simes and Storey procedures is due to the uncertainty of the empirical estimation of the coverage.

## F.3 Survey animal populations for CIFAR data set

In this section, we illustrate the task of predicting the counts for each category (task (ii) in the main text) for the batch displayed in Figure 6. The lower and upper bounds for the number of each animal in this batch are given in Table 10. As in the previous section, while the improvement of the new methods are significant for the size of the batch prediction sets, it is more modest for the bounds.

## F.4 Full versus class calibrated $p$-values under label shift

In this section, we illustrate the importance of the *class conditional model* and the conditional guarantee (2) with the CIFAR dataset. The calibration sample is of size $n = 2000$, with 10% of birds, 30% of cats and 60% of dogs. The test sample is $m = 5$ with 2 birds (40%), 3 cats (60%), hence without dogs. They are both drawn without

| $Y_1 = 1$ | $Y_2 = 1$ | $Y_3 = 2$ | $Y_4 = 2$ | $Y_5 = 3$ | $Y_6 = 3$ | Bonf | Simes |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 2 | 3 | 0.12 | 0.07 |
| 2 | 1 | 2 | 3 | 2 | 3 | 0.12 | 0.07 |
| 1 | 2 | 2 | 3 | 2 | 3 | 0.12 | 0.07 |
| 2 | 2 | 2 | 3 | 2 | 3 | 0.12 | 0.07 |
| 1 | 1 | 3 | 3 | 2 | 3 | 0.12 | 0.07 |
| 2 | 1 | 3 | 3 | 2 | 3 | 0.12 | 0.07 |
| 1 | 2 | 3 | 3 | 2 | 3 | 0.12 | 0.07 |
| 2 | 2 | 3 | 3 | 2 | 3 | 0.12 | 0.07 |
| 1 | 1 | 2 | 2 | 2 | 3 | 0.12 | 0.12 |
| 2 | 1 | 2 | 2 | 2 | 3 | 0.12 | 0.12 |
| 1 | 2 | 2 | 2 | 2 | 3 | 0.12 | 0.12 |
| 2 | 2 | 2 | 2 | 2 | 3 | 0.12 | 0.12 |
| 1 | 1 | 3 | 2 | 2 | 3 | 0.12 | 0.12 |
| 2 | 1 | 3 | 2 | 2 | 3 | 0.12 | 0.12 |
| 1 | 2 | 3 | 2 | 2 | 3 | 0.12 | 0.12 |
| 2 | 2 | 3 | 2 | 2 | 3 | 0.12 | 0.12 |
| 2 | 2 | 3 | 3 | 3 | 3 | 0.15 | 0.12 |
| 1 | 2 | 3 | 3 | 3 | 3 | 0.15 | 0.12 |
| 1 | 1 | 2 | 3 | 3 | 3 | 0.15 | 0.15 |
| 2 | 1 | 2 | 3 | 3 | 3 | 0.15 | 0.15 |
| 1 | 2 | 2 | 3 | 3 | 3 | 0.15 | 0.15 |
| 2 | 2 | 2 | 3 | 3 | 3 | 0.15 | 0.15 |
| 1 | 1 | 3 | 3 | 3 | 3 | 0.15 | 0.15 |
| 2 | 1 | 3 | 3 | 3 | 3 | 0.15 | 0.15 |
| 2 | 2 | 3 | 2 | 3 | 3 | 0.33 | 0.16 |
| 1 | 2 | 3 | 2 | 3 | 3 | 0.33 | 0.19 |
| 2 | 2 | 2 | 2 | 3 | 3 | 0.33 | 0.24 |
| 2 | 1 | 3 | 2 | 3 | 3 | 0.37 | 0.24 |
| 1 | 2 | 2 | 2 | 3 | 3 | 0.33 | 0.33 |
| 1 | 1 | 3 | 2 | 3 | 3 | 0.37 | 0.37 |
| 2 | 1 | 2 | 2 | 3 | 3 | 0.48 | 0.48 |
| 1 | 1 | 2 | 2 | 3 | 3 | 1 | 0.65 |

Table 6: The batch prediction set using Bonferroni at $\alpha = 0.1$, as well as the Bonferroni and Simes $p$-values for each $y$.

replacement in the CIFAR data set. While the distribution of $X$ given $Y$ is the same, there is a significant label shift between the calibration and test samples. Hence, using full-calibrated $p$-values is not appropriate and we should rely on class-conditional $p$-values to retain the guarantees (2) and thus the $(1 - \alpha)$-coverage under this specific data-generation process.

The coverage of the different approaches are approximated with 1000 replications and reported in Figure 7. This corroborates the theoretical findings: the full calibrated approaches can miss the nominal coverage by a lot in this case, whereas the class calibrated approaches ensure the correct coverage.

### F.5 LRT computation time

Table 11 provides the time to compute one batch prediction set for different methods with the CIFAR and USPS datasets, averaged over 500 simulations and in the simulation setting of § 4.2. As one can see, the LRT is by far the most computationally demanding method.

| | Expected size of batch prediction set | | | | | | | Probability of non-coverage | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Storey | Median | Oracle | | | | | Storey | Median | Oracle | | |
| SNR | Bonf | Simes | Simes | Simes | Simes | Fisher | LRT | Bonf | Simes | Simes | Simes | Simes | Fisher | LRT |
| 1.00 | 410.52 | 384.66 | 327.55 | 346.09 | *160.22* | **274.36** | 277.58 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.11 |
| 1.50 | 217.69 | 187.36 | 142.98 | 154.47 | *70.56* | **107.85** | 113.88 | 0.09 | 0.09 | 0.10 | 0.09 | 0.09 | 0.10 | 0.10 |
| 2.00 | 81.63 | 65.52 | 49.12 | 50.35 | *26.32* | **37.40** | 37.76 | 0.08 | 0.08 | 0.09 | 0.09 | 0.08 | 0.09 | 0.10 |
| 2.50 | 23.51 | 17.98 | 15.08 | 14.53 | *9.05* | 14.60 | **11.91** | 0.10 | 0.11 | 0.11 | 0.11 | 0.11 | 0.10 | 0.11 |
| 3.00 | 6.42 | 5.35 | 5.18 | 4.90 | *3.57* | 7.78 | **4.35** | 0.08 | 0.09 | 0.09 | 0.08 | 0.09 | 0.09 | 0.09 |
| 3.50 | 2.46 | 2.24 | 2.27 | 2.21 | *1.79* | 5.20 | **2.02** | 0.08 | 0.08 | 0.09 | 0.09 | 0.08 | 0.08 | 0.09 |
| 4.00 | 1.39 | 1.34 | 1.37 | 1.38 | *1.22* | 4.38 | **1.28** | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 | 0.09 |
| 4.50 | 1.07 | 1.06 | 1.08 | 1.09 | *1.03* | 4.03 | **1.03** | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 | 0.10 | 0.09 |

Table 7: The average batch prediction set size at each SNR (columns 2–8) and probability of non-coverage (columns 9–15) for the batch conformal prediction inference at level $\alpha = 0.1$, for the following $p$-value combining functions: Bonferroni, Simes, adaptive Simes using Storey's estimator and the median estimator (see detailed data generation in text), oracle Simes, Fisher, and the estimated LRT. In bold, the (practical) combining method that produces the narrowest prediction region (oracle adaptive Simes is in italic). Based on 2000 simulations. For a single data generation, the average running time on a standard PC was less than 0.05 seconds for all methods but the estimated LRT, which has an average running time of 5.7 seconds.

| | | | Shortcut | Oracle | Storey | Shortcut Storey | |
|---|---|---|---|---|---|---|---|
| SNR | Bonf | Simes | Simes | Simes | Simes | Simes | Fisher |
| 1 | 0.1735 | **0.1799** | **0.1799** | *0.3056* | 0.1601 | 0.1598 | 0.0959 |
| 1.5 | 0.5731 | 0.5923 | 0.5923 | *0.8769* | **0.5998** | 0.5973 | 0.4691 |
| 2 | 1.3846 | 1.4423 | 1.4423 | *1.8984* | **1.4692** | 1.4665 | 1.3304 |
| 2.5 | 2.6567 | **2.7494** | **2.7494** | *3.2361* | 2.7424 | 2.7375 | 2.4744 |
| 3 | 3.9335 | **4.0222** | **4.0222** | *4.4062* | 3.9831 | 3.9718 | 3.4714 |
| 3.5 | 5.0332 | **5.0740** | **5.0740** | *5.2971* | 5.0384 | 5.0297 | 4.2149 |
| 4 | 5.6546 | **5.6741** | 5.6725 | *5.7897* | 5.6505 | 5.6431 | 4.6495 |
| 4.5 | 5.9349 | **5.9403** | 5.9320 | *5.9729* | 5.9112 | 5.9031 | 4.9124 |
| 1 | 16.4065 | 16.2350 | 16.2350 | *14.6186* | 15.9986 | 16.0034 | **15.5516** |
| 1.5 | 14.6781 | 14.3764 | 14.3764 | *12.8131* | 14.1638 | 14.2222 | **13.6339** |
| 2 | 12.3595 | 11.9946 | 11.9946 | *10.8056* | 11.9328 | 12.0616 | **11.6715** |
| 2.5 | 10.0392 | **9.7815** | **9.7815** | *9.0074* | 9.8433 | 9.9388 | 10.0506 |
| 3 | 8.2403 | **8.0921** | **8.0921** | *7.6426* | 8.1661 | 8.2092 | 8.8527 |
| 3.5 | 6.9937 | **6.9344** | 6.9348 | *6.7016* | 6.9844 | 6.9952 | 8.0839 |
| 4 | 6.3479 | **6.3242** | 6.3280 | *6.2107* | 6.3514 | 6.3611 | 7.6670 |
| 4.5 | 6.0651 | **6.0595** | 6.0693 | *6.0271* | 6.0884 | 6.0979 | 7.4120 |

Table 8: Sum of average lower bounds $\sum_{k=1}^{3} \ell_\alpha^{(k)}$ (rows 1–8) and upper bounds $\sum_{k=1}^{3} u_\alpha^{(k)}$ (rows 9–16) of $\sum_{k=1}^{3} m_k(Y) = 3$ at each SNR for different batch conformal prediction inferences at level $\alpha = 0.1$. Estimation with an average over 10000 replications. The most informative practical bound has highest lower bounds / lowest upper bounds among the practical methods (in bold). Oracle Simes is in italic.

| | Sum of the average lower bounds | | | | | Sums of the average upper bounds | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Storey | | Estimated | | | Storey | | Estimated |
| SNR | Bonf | Simes | Simes | Fisher | LRT | Bonf | Simes | Simes | Fisher | LRT |
| 1 | 0.21 | **0.22** | 0.20 | 0.13 | 0.14 | 16.41 | 16.21 | 15.96 | 15.54 | **15.47** |
| 1.5 | 0.58 | 0.59 | **0.60** | 0.45 | 0.55 | 14.66 | 14.32 | 14.14 | 13.60 | **13.45** |
| 2 | 1.36 | 1.41 | 1.46 | 1.32 | **1.55** | 12.34 | 12.01 | 11.94 | 11.70 | **11.37** |
| 2.5 | 2.58 | 2.67 | 2.70 | 2.47 | **2.81** | 10.07 | 9.83 | 9.83 | 10.04 | **9.57** |
| 3 | 4.03 | 4.10 | 4.05 | 3.49 | **4.25** | 8.16 | 8.03 | 8.11 | 8.86 | **7.92** |
| 3.5 | 5.00 | 5.05 | 5.05 | 4.22 | **5.20** | 7.04 | 6.96 | 6.99 | 8.09 | **6.85** |
| 4 | 5.67 | 5.70 | 5.65 | 4.64 | **5.76** | 6.33 | 6.30 | 6.35 | 7.67 | **6.25** |
| 4.5 | 5.93 | 5.94 | 5.93 | 4.95 | **5.98** | 6.07 | 6.06 | 6.07 | 7.38 | **6.02** |

Table 9: Sum of average lower bounds $\sum_{k=1}^{3} \ell_\alpha^{(k)}$ (columns 2–6) and upper bounds $\sum_{k=1}^{3} u_\alpha^{(k)}$ (columns 7–11) for $\sum_{k=1}^{3} m_k(Y) = m$ at each SNR for the different batch conformal prediction inferences with average batch size presented in Table 2, at level $\alpha = 0.1$. Estimation with an average over 2000 replications. The most informative bound has highest lower bounds / lowest upper bounds (in bold).

| | Simes | Bonf. | Storey | Median | Fisher | LRT |
|---|---|---|---|---|---|---|
| Bird | 0 ; 9 | 0 ; 9 | 0 ; 9 | 0 ; 8 | 0 ; 8 | 0 ; 7 |
| Cat | 0 ; 10 | 0 ; 9 | 0 ; 9 | 0 ; 8 | 0 ; 8 | 0 ; 9 |
| Dog | 0 ; 10 | 0 ; 10 | 0 ; 10 | 0 ; 10 | 0 ; 10 | 0 ; 10 |
| Size | 27216 | 39366 | 24459 | 20680 | 12653 | 11313 |

Table 10: Bounds for the particular batch of Figure 6 from the CIFAR data set at level $\alpha = 0.1$. The number of birds, cats, and dogs in the batch is 5, 3, and 2, respectively.

| | Targeted coverage | | | | | |
|---|---|---|---|---|---|---|
| | USPS dataset | | | CIFAR dataset | | |
| | 0.99 | 0.95 | 0.90 | 0.99 | 0.95 | 0.90 |
| Simes | 0.027 | 0.027 | 0.026 | 0.008 | 0.007 | 0.006 |
| Bonf. | 0.024 | 0.023 | 0.023 | 0.003 | 0.003 | 0.003 |
| Storey | 0.030 | 0.030 | 0.030 | 0.005 | 0.005 | 0.005 |
| Median | 0.028 | 0.027 | 0.027 | 0.004 | 0.004 | 0.004 |
| Fisher | 0.072 | 0.072 | 0.072 | 0.015 | 0.015 | 0.015 |
| LRT | 5.690 | 5.668 | 5.656 | 7.475 | 7.461 | 7.507 |

Table 11: Mean time (in second) over 500 replications of different procedures (in rows) and for different targeted $1 - \alpha$ (in columns). The setting is the same as the one in § 4.2.

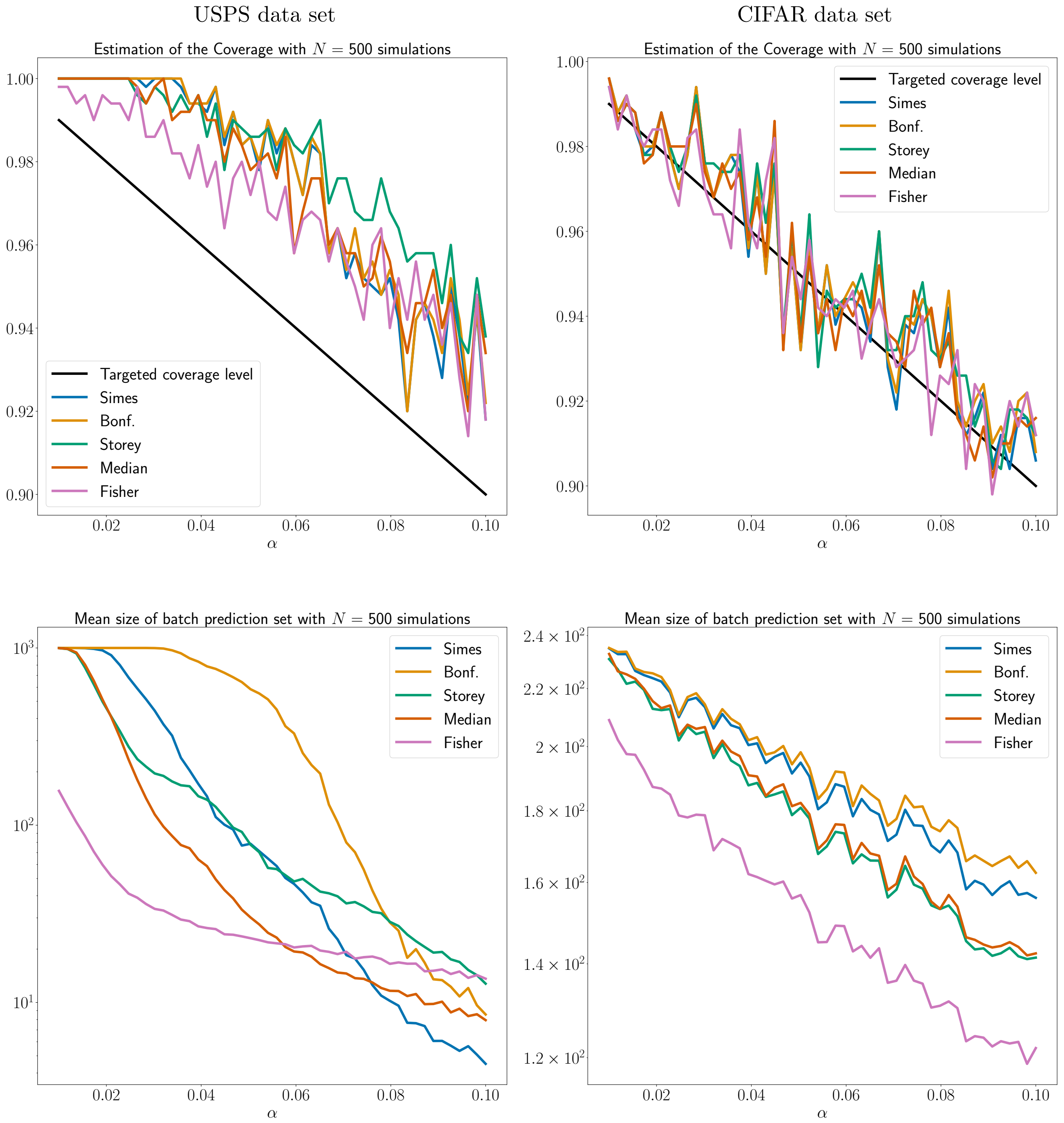


Figure 5: Top: averaged coverage of the batch prediction sets in function of $\alpha$ for different procedures. Bottom: averaged size of the batch prediction sets in function of $\alpha$ for different procedures. Same setting as for Figure 2. The standard error for the USPS dataset is below 0.013, and is below 0.014 for the CIFAR dataset.

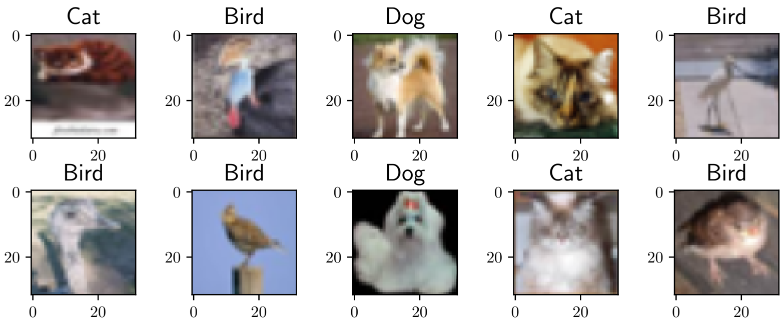


Figure 6: One batch of the CIFAR dataset (Krizhevsky, 2009).

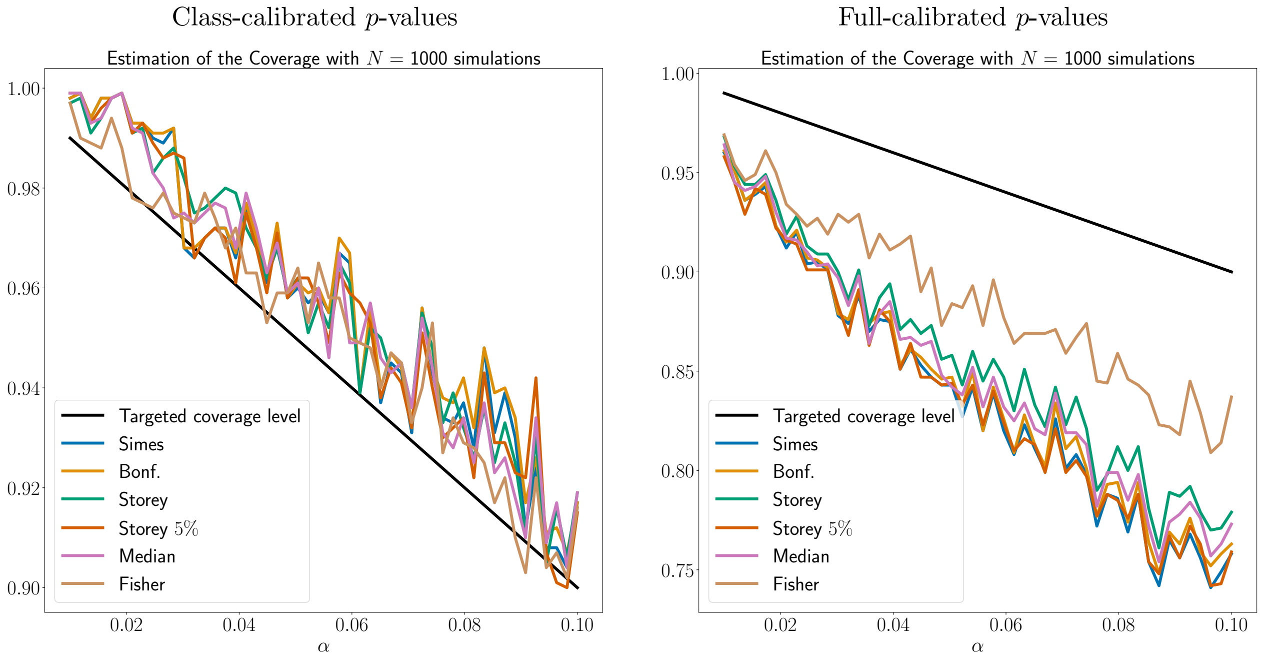


Figure 7: Estimated coverage for class calibrated $p$-values (left) and full calibrated $p$-values (right) in the label shift setting described in § F.4. The estimated standard errors are below 0.009 with the class-calibrated $p$-values, and are all below 0.014 with the full-calibrated $p$-values.