Powerful batch conformal prediction for classification

Ulysse Gazin, Ruth Heller, Etienne Roquain, Aldo Solari

▶ **To cite this version:**

Ulysse Gazin, Ruth Heller, Etienne Roquain, Aldo Solari. Powerful batch conformal prediction for classification. 2024. hal-04803539

HAL Id: hal-04803539
https://hal.science/hal-04803539v1

Preprint submitted on 25 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Powerful batch conformal prediction for classification

Ulysse Gazin* Ruth Heller† Etienne Roquain‡ Aldo Solari§

November 25, 2024

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. In this work, we explore the case where a 'batch' of m independent such unlabeled examples is given, and a batch prediction set with $1-\alpha$ coverage should be provided for this batch. Hence, the batch prediction set takes the form of a collection of label vectors of size m, while the calibration sample only contains univariate labels. Using the Bonferroni correction consists in concatenating the individual prediction sets at level $1-\alpha/m$ (Vovk, 2013). We propose a uniformly more powerful solution, based on specific combinations of conformal p-values that exploit the Simes inequality (Simes, 1986). Intuitively, the pooled evidence of fairly 'easy' examples of the batch can help provide narrower batch prediction sets. We also introduced adaptive versions of the novel procedure that are particularly effective when the batch prediction set is expected to be large. The theoretical guarantees are provided when all examples are independent and identically distributed (iid), as well as more generally when iid is assumed only conditionally within each class. In particular, our results are also valid under a label distribution shift since the distribution of the labels need not be the same in the calibration sample and in the new 'batch'. The usefulness of the method is illustrated on 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

*Université Paris Cité and Sorbonne Université, CNRS, Laboratoire de Probabilités, Statistique et Modélisation. Email: ugazin@lpsm.paris

†Department of Statistics and Operations Research, Tel-Aviv University. Email: ruheller@gmail.com

‡Sorbonne Université and Université Paris Cité, CNRS, Laboratoire de Probabilités, Statistique et Modélisation. Email: etienne.roquain@upmc.fr

§Department of Economics, Ca Foscari University of Venice. Email: aldo.solari@unive.it

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

label $k \in [K]$ and any *individual* covariate $x \in \mathcal{X}$. The considered task is to produce a collection $\mathcal{C}_\alpha^m$ of vectors of $[K]^m$ such that one of the two following guarantees holds:

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

$$\mathbb{P}((Y_{n+i})_{i\in[m]} \in \mathcal{C}_\alpha^m \mid (Y_j)_{j\in[n+m]}) \geq 1-\alpha\,. \tag{2}$$

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 guarantee in (2) is considered for the *conditional model* where the label vector $(Y_i)_{i\in[n+m]}$ is fixed and the probability is taken wrt the conditional distribution of $(X_i)_{i\in[n+m]}$ given $(Y_i)_{i\in[n+m]}$. This means that $(X_i)_{i\in[n+m]}$ given $(Y_i)_{i\in[n+m]}$ has independent components, and the i-th component has marginal distribution X_i given Y_i for $i \in [n+m]$.

While unconditional guarantees of the type (1) are the most used targets for inference in the conformal literature (Angelopoulos and Bates, 2021), we emphasize that (2) is a much stronger guarantee (Vovk et al., 2005; Sadinle et al., 2019; Romano et al., 2020), often referred to as Mondrian conformal prediction: in our framework, since the true label is 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.

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

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

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

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

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

$$[\ell_\alpha^{(k)}, u_\alpha^{(k)}] := [\min \mathcal{N}_k(\mathcal{C}_\alpha^m), \max \mathcal{N}_k(\mathcal{C}_\alpha^m)], \tag{4}$$

where $\mathcal{N}_k(\mathcal{C}_\alpha^m) := \{m_k(y) : y \in \mathcal{C}_\alpha^m\}$, 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 $\mathcal{C}_\alpha^m$ ensuring (1) or (2) provides a solution, see also Figure 1 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 $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$ as in (4) provides a solution, see the supplementary file for illustrations.

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 distribution-free joint inference on multiple test points. But they did not develop methodology targeting the inferential guarantees (1),(2).

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 α each of the $y \in [K]^m$ possible sets of labels for $(y_{n+i})_{i\in[m]}$. Vovk (2013) suggested in the full/transductive conformal setting using Bonferroni for each partitioning hypothesis. The advantage is that only $m \times K$ conformal p-values, K for each example, need to be computed. So there is no need to go over all K^m possible vectors of labels since $m \times K$ 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 key contributions are as follows:

- We cast 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 for $(y_{n+i})_{i\in[m]}$. 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 introduce a novel *adaptive* Simes batch prediction set as follows. For each candidate vector of batch labels $y \in [K]^m$, an estimate $\hat{m}_0(y)$ of the number of true labels in y is used instead of m in the threshold of the Simes test, when testing that the true vector of labels is y.

- We extend the method that uses (adaptive) Simes to accommodate any p-value combining function, not necessarily of the Simes's type. In particular, this allows to consider a valid version of the well-known Fisher combination.

- We provide a computational shortcut algorithm to compute the bounds (4) that maintains the $1-\alpha$ coverage guarantee. For recovering the bounds (4), the shortcut is exact for $K = 2$ and approximate for $K > 2$.

- We demonstrate the usefulness of our recommendations on Image data problems/ USPS digits problems and show that each of the novel methods increases accuracy, by producing narrower batch prediction sets in specific regimes.

The new introduced methods are all valid both in the iid and conditional model, and 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 1. For each combining function, Bonferroni or Simes, the proposed batch prediction set can be expressed as the batch label vector with p-values larger than α (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 2. 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).

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); 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

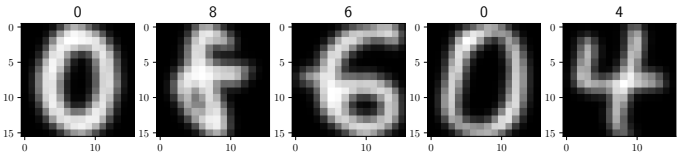

Figure 1: Illustration of task (i) : how to provide a prediction set for this observed batch, that is, a zip code composed of $m = 5$ digit images, for $K = 10$ possible digits? This illustration come from the USPS dataset (provided by the US Postal Service for the paper LeCun et al. (1989)) previously studied by Vovk (2013).

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**

Table 1: Batch prediction sets at level 0.05 for Bonferroni's and Simes' methods computed on the particular batch of Figure 1. The two last columns are the p-values of each method for the selected batches, see (7) and (9). The batch prediction set corresponds to batch p-values displayed in bold.

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

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

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

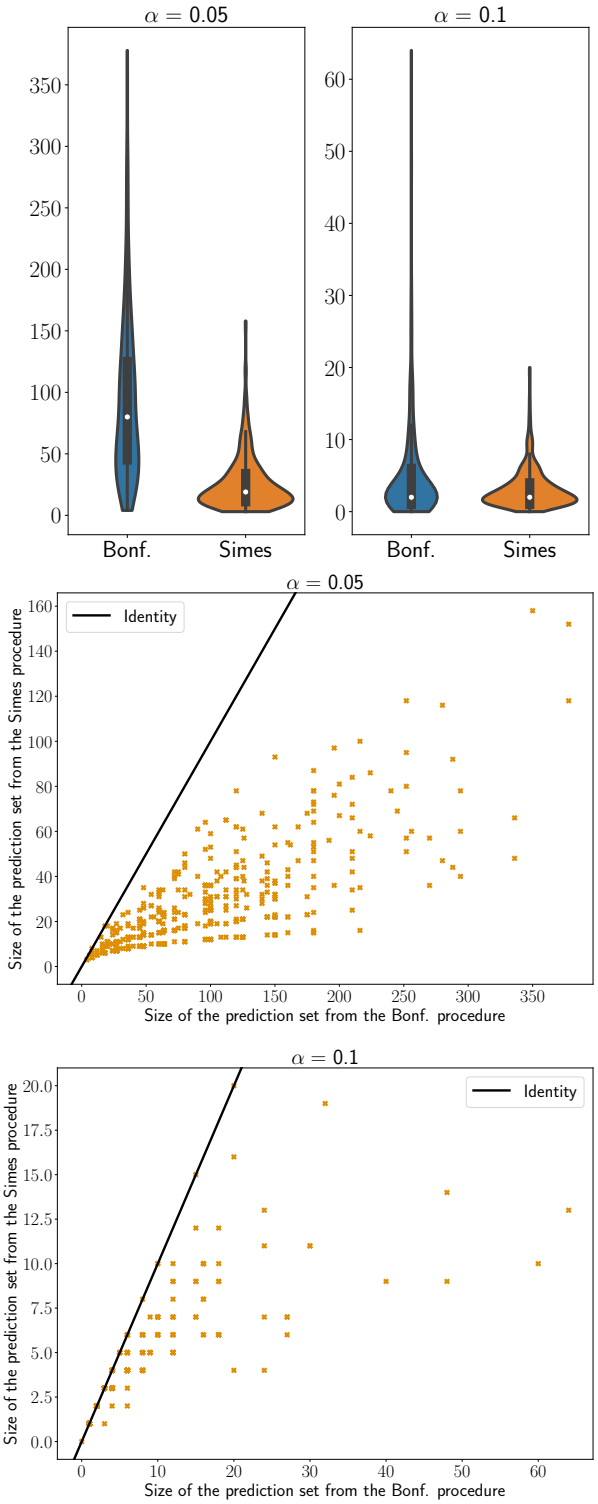

Figure 2: Violin plots (top row) and scatter plots (middle/bottom rows) for the size of the batch prediction sets of Bonferroni's and Simes' methods ($m = 3$, $K = 10$, 500 replications) for two values of α.

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 $\mathbb{P}(p_i^{(Y_{n+i})} \leq u | (Y_j)_{j \in [n+m]}) \leq u$ for class-calibrated p-values.*

Proposition 2.1 ensures that each individual label set $\mathcal{C}_{i,\alpha} := \{y_{n+i} \in [K] \,:\, p_i^{(y_{n+i})} > \alpha\}$ 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:

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

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

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

Hence, this prediction set is rectangular:

$$\mathcal{C}_{\alpha,\text{Bonf}}^m = \times_{i=1}^m \{k \in [K] \,:\, p_i^{(k)} > \alpha/m\},$$

and 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 (2) and (1) hold by using the class-calibrated and full-calibrated p-values, respectively.

2.3 Simes batch prediction set

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

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

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

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

Hence, 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 $\mathcal{C}_{\alpha,\text{Simes}}^m$ satisfies* (1) *and* (2) *by using the full-calibrated and class-calibrated p-value, respectively.*

To prove Theorem 2.2, we check that the Simes inequality (Simes, 1986) holds for the class/full-calibrated p-values, which is proved in Section C.1 by showing 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 α 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 Section C.2.

2.4 Adaptive version

For all possible label vector $y = (y_{n+i})_{i\in[m]} \in [K]^m$, let

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

the number of coordinates of y that are different from 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 consist first in estimating $m_0(y)$ by

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

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 Section A.

The adaptive Simes batch prediction set is

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

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

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

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 Section C.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 Section 4.

In the supplement, we provide another type of estimator, corresponding to the so-called the '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*).

3 Extensions and shortcut

3.1 Empirical batch prediction set

In this section, we present a general method for guaranteeing (1) and (2) using any combining function for the conformal p-values that test that the batch labels are $y \in [K]^m$. Let $F((p_i^{(y_{n+i})})_{i\in[m]})$ be any p-value vector combining function and consider a batch prediction set of the form

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

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). Here, we detail how to find a valid empirical choice of t for any F.

For this, at each b of B iterations, generate $n + m$ uniform random variables. Designate randomly n of these as calibration scores $(S_{j,b})_{j\in[n]}$ and the remaining m as test scores $(S_{i+n,b})_{i\in[m]}$. Then compute the appropriate conformal p-value vector $(\hat{p}_{i,b}^{(y_{n+i})}, i \in [m])$ according to

$$\hat{p}_{i,b}^{(k)} = \frac{1}{|\mathcal{D}^{(k)}_{\text{cal}}| + 1}\Big(1 + \sum_{j\in\mathcal{D}^{(k)}_{\text{cal}}} \mathbf{1}\{S_{j,b} \geq S_{n+i,b}\}\Big),\ k \in [K].$$

Next, combine $(\hat{p}_{i,b}^{(y_{n+i})}, i \in [m])$ into the test statistic $\xi_b = F((\hat{p}_{i,b}^{(y_{n+i})}, i \in [m]))$. Now consider the empirical threshold $t = \xi_{(\lfloor (B+1)\alpha \rfloor)}$ where $-\infty =: \xi_{(0)} < \xi_{(1)} \leq \cdots \leq \xi_{(B)}$ are the ordered test statistics.

Theorem 3.1. *Consider the batch prediction set $\mathcal{C}^m_{t,F}$ (14) with any combining function $F((p_i^{(y_{n+i})})_{i\in[m]})$ and the empirical threshold $t = \xi_{(\lfloor (B+1)\alpha \rfloor)}$ defined above. Then its coverage 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 provided in Section C.4. In short, the empirical batch prediction set hence trades computational cost for accuracy and generality.

In the conditional model, the threshold $\xi_{(\lfloor (B+1)\alpha \rfloor)}$ depends on $(m_k(y))_{k\in[K]}$, so $(n + m) \times B$ uniforms 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. However, these computations can be done once for all, before observing the data for the batch.

We underline that the method presented here is very flexible: combined with adaptive Simes combination F_{Simes}, any estimator $\hat{m}_0$ can be used, see detailed suggestions in Section 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 $\hat{m}_0(y)$ the smallest of several estimators of $\hat{m}_0(y)$. More generally, any p-value combination can be used, for instance the Fisher combination

$$F_{\text{Fisher}}((p_i^{(y_{n+i})})_{i\in[m]}) = T\Big(-2\sum_{i\in[m]} \log(p_i^{(y_{n+i})})\Big), \tag{15}$$

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.

3.2 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. The 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 Section E. 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 B.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 Section B.1 in the supplementary file).

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 $\ell_\alpha^{(k)} \leq m_k \leq u_\alpha^{(k)}$ for each $k \in \{1, \ldots, K\}$, and $m_1 + \cdots + m_K = m$, with each valid assignment counted by the multinomial coefficient $\binom{m}{m_1, m_2, \ldots, m_K}$, see Section E in the supplementary file 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 α/m for the i-th example of the batch.

4 Experiments

In this section, we study the performances of different methods: Bonferroni (6), Simes (8), Storey Simes (adaptive Simes (12) with the Storey estimator (16) where $\lambda = 1/2$), median Simes (adaptive Simes (12) with the 'median' estimator, see (18) in the supplementary file), and Fisher (empirical method with Fisher combining function (15)). 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. One example of this data generation is given in Figure 4 in the supplementary file (with the corresponding Table 3).

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. Among the two adaptive Simes variants, we see that narrower batch prediction sets are obtained using Storey's estimator for low SNR and using the median estimator otherwise. 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.

In Appendix B.1, Table 4, we show the non-coverage probability for each method, as well as the results using the true (unknown in practice) m_0. As expected, using the true $m_0(y)$ 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 Section 3.1, 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.

SNR	Bonf	Simes	Storey-Simes	Median-Simes	Fisher
1.00	418.3	392.1	332.5	353.2	**278.2**
1.50	214.7	185.7	142.1	152.5	**107.9**
2.00	76.45	60.27	46.80	47.21	**37.02**
2.50	22.27	17.26	14.66	**14.08**	14.61
3.00	6.82	5.57	5.28	**5.03**	7.63
3.50	2.55	2.29	2.32	**2.26**	5.20
4.00	1.41	**1.35**	1.38	1.38	4.40
4.50	1.07	**1.06**	1.09	1.09	3.98

Table 2: The average batch prediction set size at each SNR 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 and Fisher (see details in the text). The non-coverage probability for all methods is ≤ 0.1. In bold, the combining method that produces the narrowest batch prediction set. Based on 10000 simulations.

In Appendix B.1, Table 5, 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.

4.2 Real data sets

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

By using 500 replications, we display violin plots of the size of the batch prediction sets for the different methods in Figure 3 (for each violin, the white dot inside the inter-quartile box is the median).

For the USPS data set, the results strongly depend on the level α considered. For $\alpha = 0.01$, the batch prediction sets are all large and Fisher method is the best. For $\alpha = 0.05$ and $\alpha = 0.1$, the best batch prediction sets are obtained with the median Simes and the Simes method, respectively. On the CIFAR data set, the sizes of the prediction sets are all large (meaning that the prediction task is more difficult on this data set) and the Fisher combination is better than the other methods, followed by Storey Simes method. These findings corroborate those of the previous section. Other qualitatively similar results are obtained in the supplementary file.

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. 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.

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 $O(K \times m^2)$. It is exact for $K = 2$, and appears tight for $K > 2$ in our numerical experiments. Specifically for Simes type combination tests, computationally efficient algorithm 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

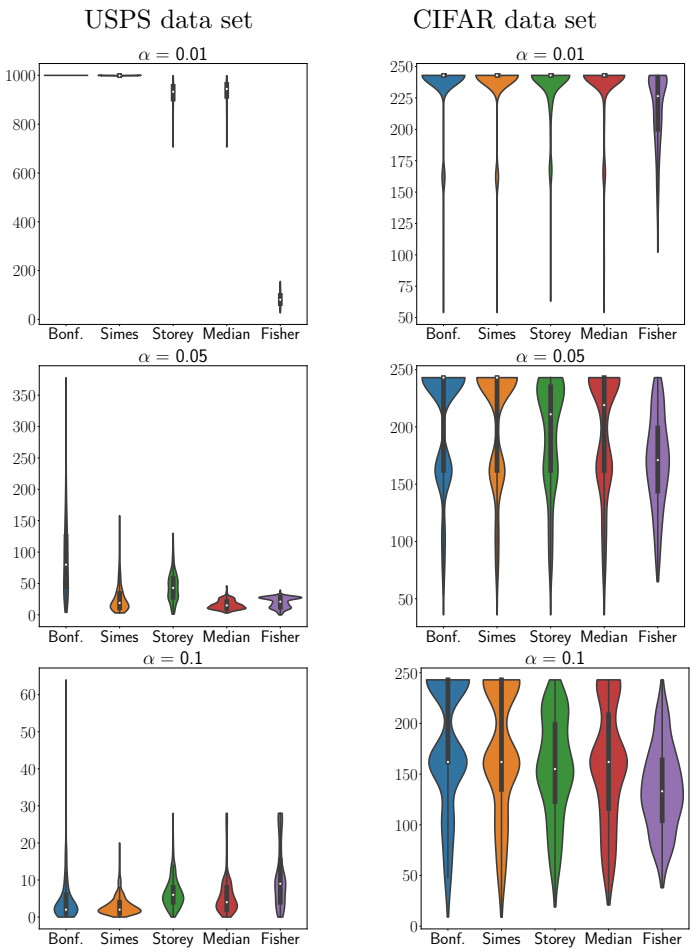

Figure 3: Violin plots for the batch prediction set for $\alpha = 1\%$, 5% and 10% (in rows) and data sets USPS and CIFAR (in columns), see details in the text.

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.

In this work, we suggest testing that the batch label vector is $y \in [K]^m$ using conformal p-value combination tests. More generally, it is possible to combine score functions $S_k(x_{n+i})$, $i \in [m], k \in [K]$, to obtain an overall batch score function $G((x_{n+i})_{i\in[m]}, (y_{n+i})_{i\in[m]})$, but then obtaining the appropriate threshold for inclusion in the batch prediction set may be more challenging. In addition, in that case, a permutation-based null distribution may depend on the score values in the batch of test points, and thus may not be distribution free. This is in contrast with the distribution free test statistics we suggest in this paper. We leave for future work the investigation of test statistics based on batch score functions.

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

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.

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.

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.

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.

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.

The appendix 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 Section 2.4. 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:

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

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

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

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

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).[2]

B Additional numerical experiments

B.1 Gaussian multivariate setting

We provide more results for the data generation described in Section 4.1. Figure 4 shows the data available in one data generation. Table 3 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 4 adds the *oracle adaptive Simes procedure*, that uses $\hat{m}_0(y) = m_0(y)$ as estimator (perfect estimation), 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.

[2]Recall that a valid coverage for the quantile Simes procedure can be ensured by using the empirical method of Section 3.1 (not used in our numerical experiments).

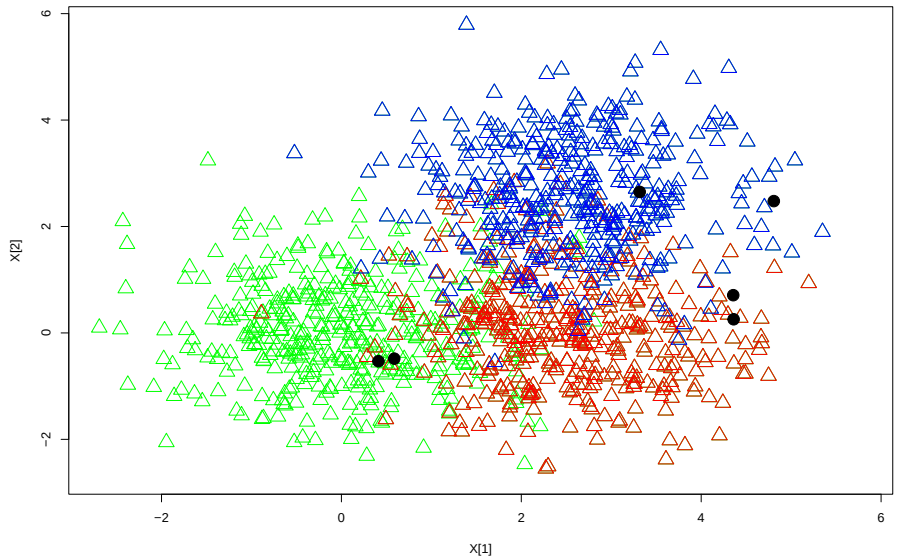

Figure 4: Illustration of one data generation with SNR = 2.5. The batch of six test samples are in black. There are 400 calibration examples from each class (class one in green, class two in red, and class three in blue). At $\alpha = 0.1$, the size of the prediction set using Bonferroni and Simes is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

Table 5 provides the average sum of lower and upper bounds for the three classes by the different methods. The goal in the comparisons in this table are two fold. First, to assess how conservative the shortcut suggested in section E 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. 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 ≥ 2.5, and better for the upper bound if SNR ≤ 2.

B.2 USPS and CIFAR data sets

To obtain a visualization different from the one of Section 4.2, Figure 5 displays the averaged size of batch prediction sets in function of α in the same setting as Figure 3. The conclusions are analogue.

B.3 Survey animal populations for CIFAR data set

In this section, we illustrate the task (ii) 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 6. 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.

	$Y_1 = 1$	$Y_2 = 1$	$Y_3 = 2$	$Y_4 = 2$	$Y_5 = 3$	$Y_6 = 3$	Bonf	Simes
1	1	1	2	3	2	3	0.12	0.07
2	2	1	2	3	2	3	0.12	0.07
3	1	2	2	3	2	3	0.12	0.07
4	2	2	2	3	2	3	0.12	0.07
5	1	1	3	3	2	3	0.12	0.07
6	2	1	3	3	2	3	0.12	0.07
7	1	2	3	3	2	3	0.12	0.07
8	2	2	3	3	2	3	0.12	0.07
9	1	1	2	2	2	3	0.12	0.12
10	2	1	2	2	2	3	0.12	0.12
11	1	2	2	2	2	3	0.12	0.12
12	2	2	2	2	2	3	0.12	0.12
13	1	1	3	2	2	3	0.12	0.12
14	2	1	3	2	2	3	0.12	0.12
15	1	2	3	2	2	3	0.12	0.12
16	2	2	3	2	2	3	0.12	0.12
17	2	2	3	3	3	3	0.15	0.12
18	1	2	3	3	3	3	0.15	0.12
19	1	1	2	3	3	3	0.15	0.15
20	2	1	2	3	3	3	0.15	0.15
21	1	2	2	3	3	3	0.15	0.15
22	2	2	2	3	3	3	0.15	0.15
23	1	1	3	3	3	3	0.15	0.15
24	2	1	3	3	3	3	0.15	0.15
25	2	2	3	2	3	3	0.33	0.16
26	1	2	3	2	3	3	0.33	0.19
27	2	2	2	2	3	3	0.33	0.24
28	2	1	3	2	3	3	0.37	0.24
29	1	2	2	2	3	3	0.33	0.33
30	1	1	3	2	3	3	0.37	0.37
31	2	1	2	2	3	3	0.48	0.48
32	1	1	2	2	3	3	1	0.65

Table 3: The batch prediction set using Bonferroni at $\alpha = 0.1$, as well as the Bonferroni and Simes p-values for each y.

C Proofs

In this section, we prove Theorems 2.2, 2.3 and 2.4. Each time, the result follows 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 class-calibrated p-values. The main technical tool for the proof is Lemma D.2 (for comparison, we also recall Lemma D.1 that was obtained for the iid case). 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 $(p_j^{(Y_{n+j})})_{j\in[m]\setminus\{i\}:Y_{n+j}=Y_{n+i}}$ exactly in the same way as for the iid case, and are independent of the p-values of the other classes $(p_j^{(Y_{n+j})})_{j\in[m]\setminus\{i\}:Y_{n+j}\neq Y_{n+i}}$.

	Expected size of batch prediction set						Probability of non-coverage					
SNR	Bonf	Simes	Storey Simes	Median Simes	Oracle Simes	Fisher	Bonf	Simes	Storey Simes	Median Simes	Oracle Simes	Fisher
1.00	418.30	392.09	332.53	353.23	*162.77*	**278.23**	0.08	0.09	0.09	0.09	0.09	0.09
1.50	214.74	185.66	142.08	152.50	*69.95*	**107.91**	0.09	0.09	0.09	0.09	0.09	0.10
2.00	76.45	60.27	46.80	47.21	*24.88*	**37.02**	0.09	0.09	0.10	0.10	0.09	0.09
2.50	22.27	17.26	14.66	**14.08**	*8.88*	14.61	0.08	0.09	0.10	0.09	0.09	0.09
3.00	6.82	5.57	5.28	**5.03**	*3.64*	7.63	0.09	0.09	0.10	0.09	0.09	0.10
3.50	2.55	2.29	2.32	**2.26**	*1.83*	5.20	0.09	0.09	0.10	0.09	0.09	0.09
4.00	1.41	**1.35**	1.38	1.38	*1.21*	4.40	0.09	0.09	0.10	0.09	0.09	0.09
4.50	1.07	**1.06**	1.09	1.09	*1.03*	3.98	0.09	0.09	0.10	0.10	0.09	0.10

Table 4: The average batch prediction set size at each SNR (columns 2–7) and probability of non-coverage (columns 8–13) 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, and Fisher. In bold, the (practical) combining method that produces the narrowest prediction region (oracle adaptive Simes is in italic). Based on 10000 simulations.

SNR	Bonf	Simes	Shortcut Simes	Oracle Simes	Storey Simes	Shortcut Storey 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 5: Lower bound $\ell_\alpha^{(k)}$ (rows 1–8) and upper bound $u_\alpha^{(k)}$ (rows 9–16) of $m_k(Y)$ (3) (class $k = 1$) at each SNR for different batch conformal prediction inferences at level $\alpha = 0.1$. Estimation with an average over 10000 replications. The bound the most informative has highest lower bounds / lowest upper bounds (in bold). Oracle Simes is in italic.

	Bonferroni	Simes	Storey	Median	Fisher
Bird	0 ; 9	0 ; 8	0 ; 7	0 ; 7	0 ; 7
Cat	0 ; 9	0 ; 9	0 ; 9	0 ; 8	0 ; 8
Dog	1 ; 10	1 ; 10	1 ; 10	1 ; 10	1 ; 10
Size	19683	14580	9540	7035	8349

Table 6: Bounds for the particular batch of Figure 6 from the CIFAR data set.

USPS data set CIFAR data set

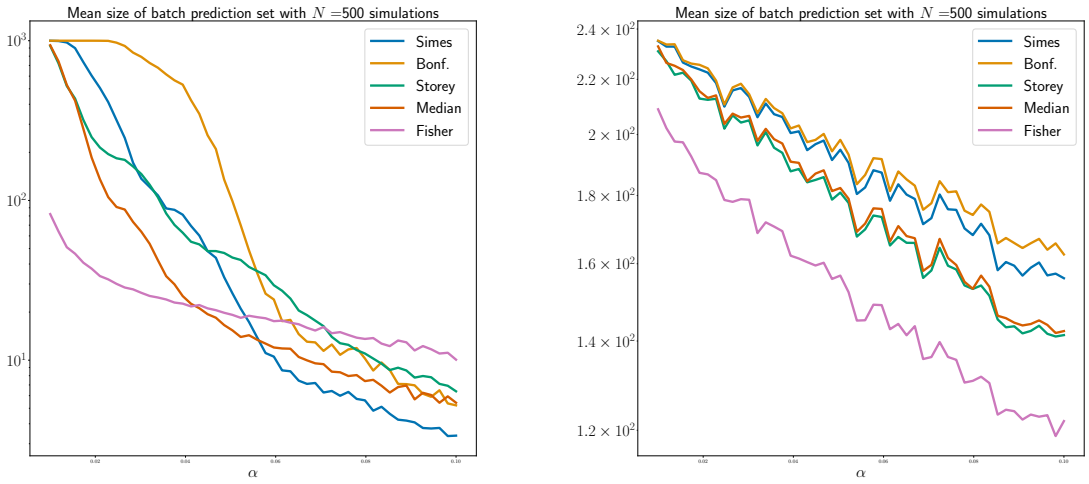

Figure 5: Averaged size of the batch prediction sets in function of α for different procedures. Same setting as for Figure 3.

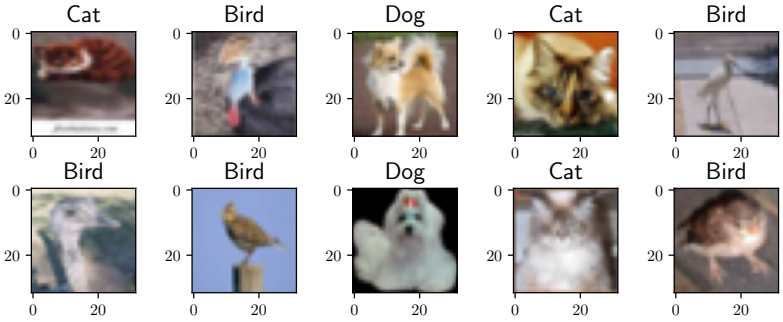

Figure 6: One batch of the CIFAR dataset (Krizhevsky, 2009).

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.

C.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 holds: for any nondecreasing[3] 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 $i \in [m]$.

[3]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 $y \in D$.

Proposition C.1. *The family of class-calibrated p-values is PRDS on* $[m]$.

Proof. By Lemma D.2, by writing $(p_i)_{i\in[m]} = (p_i, (p_j)_{j\in[m]\setminus\{i\}:Y_{n+j}=Y_{n+i}}, (p_j)_{j\in[m]:Y_{n+j}\neq Y_{n+i}})$ (with some abuse of notation in the ordering of the vector), we have

$$\begin{aligned}
&\mathbb{P}((p_i)_{i\in[m]} \in D \mid p_i = u, W_i, (p_j)_{j\in[m]:Y_{n+j}\neq Y_{n+i}}, (Y_j)_{j\in[n+m]}) \\
&= \mathbb{P}((p_i, (p_j)_{j\in[m]\setminus\{i\}:Y_{n+j}=Y_{n+i}}, (p_j)_{j\in[m]:Y_{n+j}\neq Y_{n+i}}) \in D \mid p_i = u, W_i, (p_j)_{j\in[m]:Y_{n+j}\neq Y_{n+i}}, (Y_j)_{j\in[n+m]}) \\
&= \mathbb{P}((p_i, \Psi_i(p_i, W_i), (p_j)_{j\in[m]:Y_{n+j}\neq Y_{n+i}}) \in D \mid p_i = u, W_i, (p_j)_{j\in[m]:Y_{n+j}\neq Y_{n+i}}, (Y_j)_{j\in[n+m]}) \\
&= \mathbf{1}\{(u, \Psi_i(u, W_i), (p_j)_{j\in[m]:Y_{n+j}\neq Y_{n+i}}) \in D\},
\end{aligned}$$

because p_i, W_i and $(p_j)_{j\in[m]:Y_{n+j}\neq Y_{n+i}}$ are independent conditionally on $(Y_j)_{j\in[n+m]}$. Since D is a nondecreasing set and $\Psi_i(u, W_i)$ is nondecreasing in u, we have that $\mathbf{1}\{(u, \Psi_i(u, W_i)) \in D\}$ is nondecreasing in u, which proves the result by an integration. □

C.2 Proof of Theorem 2.3

Let us denote for any $y = (y_{n+i})_{i\in[m]} \in [K]^m$,

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

(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 $\mathbf{p}(y) = (p_i^{(y_{n+i})})_{i\in[m]}$. Observe that, $y \notin \mathcal{C}^m_{\alpha,\text{Simes}}$ if and only if $\widehat{\ell}(\mathbf{p}(y)) \geq 1$, which is true if and only if $\sum_{i\in[m]} \mathbf{1}\{p_i^{(y_{n+i})} \leq (\alpha/m)(1 \vee \widehat{\ell}(\mathbf{p}(y)))\} = 1 \vee \widehat{\ell}(\mathbf{p}(y))$.

Now, denoting $\mathbf{p} = (p_i)_{i\in[m]}$ the family of class-calibrated p-values, we have

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

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

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

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

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

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

C.3 Proof of Theorem 2.4

For short, we sometimes write in this proof λ_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 κ instead of $\kappa((Y_{n+i})_{i\in[m]})$ and m_k instead of $m_k((Y_{n+i})_{i\in[m]})$.

Let $G(\mathbf{p}) = \hat{m}_0((Y_{n+i})_{i\in[m]}) = \kappa(1 + \sum_{i=1}^m \mathbf{1}\{p_i \geq \lambda_i\})$ the estimator of m_0 at the true point $(Y_{n+i})_{i\in[m]}$ given in (16). Similarly to (21), we have

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

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 D.2 and the notation therein, and we observe that $(p_j)_{j\in[m]\setminus\{i\}}$ 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) which are nonincreasing in p_i. Now let

$$c^*(W_i) = \max \mathcal{N}(W_i)$$
$$\mathcal{N}(W_i) = \{a/(n_i+1) \,:\, a \in [n_i+1], a/(n_i+1) \le (\alpha/G(a/(n_i+1), (p_j)_{j\in[m]\setminus\{i\}}))1 \vee \widehat{\ell}(a/(n_i+1), (p_j)_{j\in[m]\setminus\{i\}})\},$$

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

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

where the first inequality comes from the definition of $\mathcal{N}(W_i)$ and $c^*(W_i)$ and from the fact that $\widehat{\ell}(c^*(W_i), (p_j)_{j\in[m]\setminus\{i\}})$ is W_i-measurable; the second inequality comes from Lemma D.2 (ii); and the third one comes from the fact that $c^*(W_i)$ is in $\mathcal{N}(W_i)$ and $1/G(\mathbf{p})$ is nonincreasing in p_i. Given the notation of Lemma D.2 (v), this leads to

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

where $\mathbf{p}' = (p'_j)_{j\in[m]}$ is such that $p'_i = (n_i+1)^{-1}$, $(p'_j)_{j\in[m]:Y_{n+j}=Y_{n+i}} \sim \mathcal{D}_i$ and for each $k \ne Y_{n+i}$, $(p'_j)_{j\in[m]:Y_{n+j}=k} \sim \mathcal{D}_k$ where the distribution of $\mathcal{D}_i$ and $\mathcal{D}_k$ are defined in Lemma D.2. Also note that $(p'_j)_{j\in[m]:Y_{n+j}=Y_{n+i}}$ and all $(p'_j)_{j\in[m]:Y_{n+j}=k}$, $k \ne Y_{n+i}$, are independent vectors, so that the distribution of $\mathbf{p}'$ is well specified. Now observe that

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

by using Lemma D.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 $\lfloor(n_k+1)\lambda\rfloor \le 1$) 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 $Y_{n+j} = k \ne Y_{n+i}$,

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

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

Now, by Lemma D.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 ν be the random variable

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

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

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

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

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

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

$$\kappa^{-1}(\mathbb{E}((\nu_i')^{-1}))^{(m_i-1)/(m-1)}\prod_{k\neq Y_{n+i}}(\mathbb{E}((\nu_k)^{-1})))^{m_k/(m-1)}$$
$$=\kappa^{-1}\Big(\frac{n_i+1}{n_i+1-\lfloor(n_i+1)\lambda\rfloor}\Big)^{(m_i-1)/(m-1)}\prod_{k\neq Y_{n+i}}\Big(\frac{n_k}{n_k+1-\lfloor(n_k+1)\lambda\rfloor}\Big)^{m_k/(m-1)}$$
$$\leq\kappa^{-1}\Big(\frac{1}{1-\lambda_i}\Big)^{(m_i-1)/(m-1)}\prod_{k\neq Y_{n+i}}\Big(\frac{1}{1-\lambda_k}\Big)^{m_k/(m-1)}\leq 1,$$

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 D.4 and by the definition (17) of κ. Combining the latter with (22) gives the result.

C.4 Proof of Theorem 3.1

First, by letting $\xi_{(B+1)}:=\infty$, we have $\xi_{(\lfloor(B+1)\alpha)\rfloor)}=(-\xi)_{(\lceil(B+1)(1-\alpha)\rceil)}$ and thus

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

where the probability is taken conditionally on $(Y_{n+i})_{i\in[m]}$ in the conditional model. Now, since the scores $S_{Y_i}(X_i)$, $i\in[n+m]$, have no ties and p-values $(p_i^{(Y_{n+i})})_{i\in[m]}$ involve only ranks between those scores, the distribution of $(p_i^{(Y_{n+i})})_{i\in[m]}$ is same as if the scores were iid uniform on $[0,1]$. As a result, letting $\xi=F((p_i^{(Y_{n+i})})_{i\in[m]})$, the variables ξ, ξ_1, …, ξ_B are iid and thus exchangeable. By Romano and Wolf (2005), this entails that

$$\mathbb{P}\Big((B+1)^{-1}\Big(1+\sum_{b=1}^{B}\mathbf{1}\{-\xi_b\geq-\xi\}\Big)\leq\alpha\Big)\leq\alpha.$$

Since $(B+1)^{-1}\Big(1+\sum_{b=1}^{B}\mathbf{1}\{-\xi_b\geq-\xi\}\Big)\leq\alpha$ if and only if $-\xi>(-\xi)_{(\lceil(B+1)(1-\alpha)\rceil)}$, this gives the result.

D Technical results

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

Lemma D.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 $i \in [m]$,*

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

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 $j \in [m]$, $p'_j \leq p_j$ when $p_j \leq p_i$ and $p'_j = p_j$ when $p_j > p_i$.

The next lemma adapts Lemma D.1 to the class conditional model (with class-calibrated p-values).

Lemma D.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 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*

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

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

(i) $(p_j)_{j\in[m]\setminus\{i\}:Y_{n+j}=Y_{n+i}}$ *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 $(p_j)_{j\in[m]\setminus\{i\}:Y_{n+j}=Y_{n+i}}$ 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 $\mathcal{D}_i((Y_j)_{j\in[n+m]})$ which is defined as follows: $(p'_j)_{j\in[m]\setminus\{i\}:Y_{n+j}=Y_{n+i}} \sim \mathcal{D}_i((Y_j)_{j\in[n+m]})$ if, conditionally on the ordered statistics $U_{(1)} > \cdots > U_{(n_i+1)}$ of an iid sample of uniform random variables $(U_1, \dots, U_{n_i+1})$ (independent of everything else), the variables $(p'_j)_{j\in[m]\setminus\{i\}:Y_{n+j}=Y_{n+i}}$ 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 $\mathcal{D}_k((Y_j)_{j\in[n+m]})$ which is defined as follows: $(p'_j)_{j\in[m]:Y_{n+j}=k} \sim \mathcal{D}_k((Y_j)_{j\in[n+m]})$ if, conditionally on the ordered statistics $U^{(k)}_{(1)} > \cdots > U^{(k)}_{(n_k)}$ of an iid sample of uniform random variables $(U^{(k)}_1, \ldots, U^{(k)}_{n_k})$ (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 $j \neq i$. Then, $(p'_j)_{j\in[m]:Y_{n+j}=Y_{n+i}}$ is W_i-measurable and almost surely, for all $j \in [m]$, $p'_j \leq p_j$ 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 $Y_{n+j} = Y_{n+i}$,

$$\begin{aligned} p_j &= \frac{1}{|\mathcal{D}^{(Y_{n+j})}_{\text{cal}}| + 1}\Big(1 + \sum_{\ell\in\mathcal{D}^{(Y_{n+j})}_{\text{cal}}} \mathbf{1}\{S_\ell \geq S_{n+j}\}\Big) \\ &= \frac{1}{|\mathcal{D}^{(Y_{n+i})}_{\text{cal}}| + 1}\Big(1 + \sum_{s\in A_i} \mathbf{1}\{s \geq S_{n+j}\} - \mathbf{1}\{S_{n+i} \geq S_{n+j}\}\Big), \end{aligned} \tag{23}$$

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 $\{S_j, j \in \mathcal{D}^{(Y_{n+i})}_{\text{cal}}\} \cup \{S_{n+i}\}$ 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 $U(0,1)$. Now, by (i), we have for all $j \in [m]\backslash\{i\}$ with $Y_{n+j} = Y_{n+i}$,

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

which thus 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

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

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 $Y_{n+j} = k$,

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

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 (23). 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 (23). The result is proved. □

Lemma D.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*

$$\forall j \in [m], \begin{cases} p'_j \le p_j & \text{if } p_j \le p_i; \\ p'_j = p_j & \text{if } p_j > p_i. \end{cases} \tag{24}$$

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

Lemma D.4. *For $V_{(1)} > \cdots > V_{(\ell)}$ the order statistics of ℓ 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 D.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 D.6 (Lemma 1 of Benjamini et al. (2006)). *If T is a Binomial variable with parameter $m-1 \ge 0$ and $\nu \in (0,1]$, we have*

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

E Computational shortcut

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)[4]. 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.

E.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 E.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 1 provides the pseudocode for the shortcut to compute the bounds $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$ derived from the (adaptive) Simes conformal prediction set.

Proposition E.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 1 returns the bounds $[\tilde{\ell}_\alpha^{(k)}, \tilde{u}_\alpha^{(k)}]$ such that $\tilde{\ell}_\alpha^{(k)} \le \ell_\alpha^{(k)}$ and $\tilde{u}_\alpha^{(k)} \ge u_\alpha^{(k)}$ 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 $k \in [K]$.*

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

Algorithm 1: 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 $q_{(1)} \leq \ldots \leq q_{(m)}$;
    $h_{v,k} \leftarrow \min\Big(\frac{\hat{m}_0(\mathbf{q})}{\ell} q_{(\ell)}, \ell \in [m]\Big)$
  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: $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$, $k \in [K]$

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 $a_1, \ldots, a_{m-i}$ and $b_1, \ldots, b_i$ 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 $p_{(1)} \leq \ldots \leq p_{(m)}$. Simes' test is defined as $F_{\text{Simes}}(\mathbf{p}) = \min\Big(\frac{m}{\ell} p_{(\ell)}, \ell \in [m]\Big)$. This test is monotonic, meaning that if $\mathbf{p} \leq \mathbf{q}$ componentwise (i.e. $p_{(i)} \leq q_{(i)}$ for all $i \in [m]$), then $F_{\text{Simes}}(\mathbf{p}) \leq F_{\text{Simes}}(\mathbf{q})$.

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

Then, for some $\mathbf{q} = (q_i)_{i\in[m]}$ with $\mathbf{q} \geq \mathbf{p}(y)$ for all $y \in [K]^m$ such that $m_k(y) = v$, $F_{\text{Simes}}(\mathbf{q}) \leq \alpha$ 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 1 identifies a suitable vector $\mathbf{q} = \mathbf{q}_{v,k}$ such that $\mathbf{q} \geq \mathbf{p}(y)$ for all $y \in [K]^m$ where $m_k(y) = v$. Then we let

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

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

We now need to demonstrate that Algorithm 1 produces a vector $\mathbf{q}$ such that $\mathbf{q} \geq \mathbf{p}(y)$ 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 $p_{i_1}^{(k)}, \ldots, p_{i_v}^{(k)}$ and $m-v$ conformal p-values $p_{i_{v+1}}^{(j_1)}, \ldots, p_{i_m}^{(j_{m-v})}$, where $i_1, \ldots, i_m$ is a permutation of $[m]$ and $j_1, \ldots, j_{m-v} \in [K] \setminus \{k\}$. If we consider the vector $\mathbf{p}(\tilde{y})$, which is formed by $p_{i_1}^{(k)}, \ldots, p_{i_v}^{(k)}$ and the maximum values $\max(p_{i_{v+1}}^{(j)}, j \neq k), \ldots, \max(p_{i_{v+m}}^{(j)}, j \neq k)$, we can conclude that $\mathbf{p}(\tilde{y}) \geq \mathbf{p}(y)$. Since the vector $\mathbf{q}$ in Algorithm 1 is constructed using the largest v values from $(p_i^{(k)})_{i\in[m]}$ and the largest $m-v$ values from $(\max(p_i^{(j)}, j \neq k))_{i\in[m]}$, it follows that $\mathbf{q} \geq \mathbf{p}(\tilde{y}) \geq \mathbf{p}(y)$ for all $y \in [K]^m$ such that $m_k(y) = v$. This establishes the conservativeness of the shortcut for $K \geq 2$ 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 $i_1, \ldots, i_m$ such that the sequence $S_k(x_{n+i_{j_1}}) \leq \ldots \leq S_k(x_{n+i_{j_m}})$ is nondecreasing, while the sequence $S_{3-k}(x_{n+i_{j_1}}) \geq \ldots \geq S_{3-k}(x_{n+i_{j_m}})$ is nonincreasing. Consequently, the ranks of $S_k(x_{n+j_1}), \ldots, S_k(x_{n+j_m})$ within the set $(S_{y_j}(x_j))_{j \in \mathcal{D}_{\text{cal}}^{(k)}}$ will be nondecreasing, while the ranks of $S_{3-k}(x_{n+j_1}), \ldots, S_{3-k}(x_{n+j_m})$ within the set $(S_{y_j}(x_j))_{j \in \mathcal{D}_{\text{cal}}^{(3-k)}}$ will be nonincreasing. Since these ranks are proportional to the conformal p-values, it follows that $p_{i_1}^{(k)} \leq \ldots \leq p_{i_m}^{(k)}$ and $p_{i_1}^{(3-k)} \geq \ldots \geq p_{i_m}^{(3-k)}$.

Consider $y \in [K]^m$ such that $m_k(y) = v$. Let the vector $\mathbf{p}(y^*)$ consist of the v largest values from $(p_i^{(k)})_{i \in [m]}$, specifically $p_{i_{m-v+1}}^{(k)}, \ldots, p_{i_m}^{(k)}$. Consequently, the remaining $m - v$ values in $\mathbf{p}(y^*)$ are $p_{i_1}^{(3-k)}, \ldots, p_{i_{m-v}}^{(3-k)}$, i.e. the largest $m-v$ values from $(p_i^{(3-k)})_{i \in [m]}$. Thus, we have $\mathbf{p}(y^*) \geq \mathbf{p}(y)$ for all $y \in [K]^m$ such that $m_k(y) = v$. Furthermore, by construction, $\mathbf{q}$ in Algorithm 1 is equal to $\mathbf{p}(y^*)$. Therefore $F_{\text{Simes}}(\mathbf{q}) \leq \alpha$ if and only if $F_{\text{Simes}}(\mathbf{p}(y)) \leq \alpha$ 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 $\mathbf{p}(y) \leq \mathbf{q}$, then $\hat{m}_0(\mathbf{p}(y)) \leq \hat{m}_0(\mathbf{q})$ holds for any $y \in [K]^m$. This, combined with $F_{\text{A-Simes}}(\mathbf{p}(y)) \leq \alpha$ if and only if $F_{\text{Simes}}(\mathbf{p}(y)) \leq m\alpha / \hat{m}_0(\mathbf{p}(y))$ yields the desired result. □

E.2 Extension to other combining functions

Algorithm 2 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 3.1, 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 2 yields conservative yet valid bounds is analogous to the previous result and is therefore omitted.

Algorithm 2: 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: $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$, $k \in [K]$

E.3 Batch prediction set reconstruction from the bounds

As announced in Section 3.2, 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 $|\tilde{\mathcal{C}}_{\alpha,\text{Simes}}^m| = 600$, compared to $|\mathcal{C}_{\alpha,\text{Simes}}^m| = 6$ 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* $\mathcal{C}_{\alpha,Bonf}^m$. 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 $\mathcal{C}_{i,\alpha}^m$ 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 $[\ell_\alpha^{(k)}, u_\alpha^{(k)}]$ 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 $\mathcal{C}_{\alpha,\text{Bonf}}^m$ because it violates the constraint that the number of 1s must not exceed 8.