Classification under Nuisance Parameters and Generalized Label Shift in Likelihood-Free Inference

Luca Masserano [*1,2] **Alex Shen** [*1] **Michele Doro** [3] **Tommaso Dorigo** [4,5,6] **Rafael Izbicki** [7] **Ann B. Lee** [1,2]

Abstract

An open scientific challenge is how to classify events with reliable measures of uncertainty, when we have a mechanistic model of the data-generating process but the distribution over both labels and latent nuisance parameters is different between train and target data. We refer to this type of distributional shift as *generalized label shift* (GLS). Direct classification using observed data $\mathbf{X}$ as covariates leads to biased predictions and invalid uncertainty estimates of labels Y. We overcome these biases by proposing a new method for robust uncertainty quantification that casts classification as a hypothesis testing problem under nuisance parameters. The key idea is to estimate the classifier's receiver operating characteristic (ROC) across the entire nuisance parameter space, which allows us to devise cutoffs that are invariant under GLS. Our method effectively endows a pre-trained classifier with domain adaptation capabilities and returns valid prediction sets while maintaining high power. We demonstrate its performance on two challenging scientific problems in biology and astroparticle physics with data from realistic mechanistic models.

1. Introduction

Problem Set-up Likelihood-free inference refers to settings where the likelihood function $\mathcal{L}(\mathbf{x};\boldsymbol{\theta})$ — associated with a "theory" or model of the data-generating process — is intractable, but one is able to simulate relatively large data sets $\mathcal{T} = \{(\boldsymbol{\theta}_1, \mathbf{X}_1), \dots, (\boldsymbol{\theta}_B, \mathbf{X}_B)\} \sim p_{\text{train}}(\boldsymbol{\theta})\mathcal{L}(\mathbf{x};\boldsymbol{\theta})$. These mechanistic models (or simulators) implicitly define the "causal" model $\boldsymbol{\theta} \to \mathbf{X}$ that encodes our knowledge of how internal parameters determine observable data, and are widely used in several domains of science.

While the likelihood $\mathcal{L}(\mathbf{x};\boldsymbol{\theta})$ stays the same under the assumed theory, the prior over parameters $p_{\text{train}}(\boldsymbol{\theta})$ is *chosen by design* and can be different from the true target distribution $p_{\text{target}}(\boldsymbol{\theta})$, thereby causing a potentially harmful bias when inferring $\boldsymbol{\theta}$ given a new observation $\mathbf{x}_{\text{target}}$. If the unknown parameter of interest is a categorical variable $Y \in \mathcal{Y} = \{0, 1, \dots, K\}$ and the causal mechanistic model remains the same — that is, $p_{\text{train}}(\mathbf{X} \mid Y) = p_{\text{target}}(\mathbf{X} \mid Y)$ — the difference in the joint distribution of $(\boldsymbol{\theta}, \mathbf{X})$ between train and target data is referred to as prior probability shift or label shift (Quinonero-Candela et al., 2008; Vaz et al., 2019; Polo et al., 2023; Storkey et al., 2009; Fawcett & Flach, 2005; Moreno-Torres et al., 2012). We refer to this setting as *standard label shift* (SLS).

In this paper, we consider a more general setup that reflects a richer mechanistic model: $\boldsymbol{\theta} = (Y, \boldsymbol{\nu}) \to \mathbf{X}$, where $\boldsymbol{\nu} \in \mathcal{N}$ are continuous or discrete nuisance parameters that are not of direct interest but critically influence the data-generating process. These nuisance parameters are available at the training stage, but are *not* observed at the inference stage when estimating Y from $\mathbf{x}_{\text{target}}$. We refer to a shift that simultaneously affects Y and $\boldsymbol{\nu}$ as *generalized label shift* (GLS), and assume that $p_{\text{train}}(\mathbf{X} \mid Y, \boldsymbol{\nu}) = p_{\text{target}}(\mathbf{X} \mid Y, \boldsymbol{\nu})$. Within this setting, our goal is not just to do binary classification per se (that is, providing a 0 versus 1 response), but rather to do trustworthy uncertainty quantification for the classification output, even under GLS.

Scientific Motivation Nuisance parameters can be seen as a way of accounting for model misspecifications. Statistical models are indeed rarely accurate in capturing the complexity of physical phenomena. To account for "known unknowns", such as calibration errors in the measuring device or inaccuracies and approximations in the theory, scientists usually resort to enlarging the mechanistic model with additional parameters that are not of direct relevance, but

[*]Equal contribution [1]Department of Statistics and Data Science, Carnegie Mellon University, Pittsburgh, USA [2]Machine Learning Department, Carnegie Mellon University, Pittsburgh, USA [3]Department of Physics and Astronomy, Università di Padova, Padova, Italy [4]Istituto Nazionale di Fisica Nucleare, Sezione di Padova, Italy [5]Lulea Techniska Universitet, Lulea, Sweden [6]Universal Scientific Education and Research Network, Italy [7]Department of Statistics, Universidade Federal de São Carlos, São Paulo, Brazil. Correspondence to: Luca Masserano <lmassera@andrew.cmu.edu>, Ann B. Lee <annlee@andrew.cmu.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

yet have to be considered during inference in order to make reliable statements about the parameters of interest. These additional parameters are commonly referred to as nuisance parameters (Kitching et al., 2009; Dorigo & de Castro, 2020; Pouget et al., 2013; HEP ML Community): they are necessary to achieve more faithful models of reality, but make correct inference much more challenging.

Statistical Challenges We introduce a simplified example (see Section 5.1 for details) to illustrate the challenges of classification under the presence of nuisance parameters. Suppose $Y = 1$ represents a class with cases of interest (e.g., the presence of a medical condition) and $Y = 0$ a class with cases of no interest. We have good knowledge of the probability density function (PDF) of $Y = 1$, $f_1(x)$, but the shape of the distribution of $Y = 0$ is largely unknown. To accommodate different scenarios, we resort to a nuisance-parameterized PDF $f_0(x; \nu)$. Our goal is to discriminate between negative $Y = 0$ and positive $Y = 1$ cases based on potentially high-dimensional data $\mathbf{x} \in \mathcal{X}$ and to provide valid measures of uncertainties on the true label Y under the presence of a nuisance parameter ν. However, directly classifying $\mathbf{x}_{\text{target}}$ based on $\mathbb{P}_{\text{train}}(Y = 1 \mid \mathbf{X})$ and a cutoff C derived from $\mathcal{T} = \{(Y_i, \mathbf{X}_i)\}_{i=1}^{B}$ would lead to invalid uncertainty quantification. Indeed, under GLS (or even SLS), standard prediction sets (defined as in, e.g., Equation 10) do not guarantee marginal validity:

$$\mathbb{P}_{\text{target}}(Y \in R_\alpha(\mathbf{X})) \geq 1 - \alpha,$$

where Y and $\mathbf{X}$ are random and $\alpha \in [0, 1]$ is a pre-specified miscoverage level. Various solutions have been proposed for the SLS setting (see references in Section 2), whereas GLS is still a largely unexplored area in the machine learning literature. The key open challenge is to design general-purpose inference algorithms that can guarantee *valid* measures of uncertainty for all Y and $\boldsymbol{\nu}$ while providing high constraining power on Y (that is, smaller prediction sets).

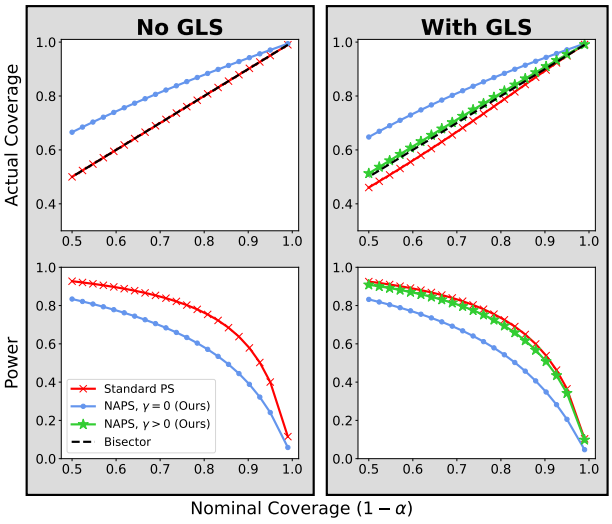

Figure 1. **Synthetic Example.** *Left (no GLS):* Standard prediction sets $R_\alpha(\mathbf{x})$ (red) guarantee marginal coverage at the nominal level. Nuisance-aware prediction sets (NAPS $\gamma = 0$; blue) are also marginally valid, but the "universality" of conditional validity across the entire nuisance parameter space comes at the price of more conservative prediction sets and lower power. *Right (with GLS):* Standard prediction sets are no longer valid and undercover for all α levels (red curve is below the black bisector), while NAPS are still valid. Furthermore, we can increase power while maintaining validity (NAPS $\gamma > 0$; green) by constructing $(1 - \gamma)$ confidence sets of the nuisance parameter ν and deriving less conservative cutoffs given an observation. Here $\gamma = \alpha \times 0.01$.

Returning to our simplified experiment, Figure 1 (top left) illustrates how standard prediction sets $R_\alpha(\mathbf{x})$ are marginally valid when the train and target distributions are the same, while under GLS prediction sets are no longer valid even marginally (top right). Our nuisance-aware prediction sets (NAPS, $\gamma = 0$ in Figure 1), on the other hand, are valid in both settings. In addition, we can increase the constraining power (NAPS, $\gamma > 0$) once we observe data without the need to re-train the classifier, effectively endowing our method with domain adaptation capabilities.

Approach and Contributions We categorize our main contributions as follows:

i) **TPR and FPR across $\mathcal{N}$.** By casting classification under GLS as a hypothesis testing problem with nuisance parameters, we propose a method to estimate the TPR and FPR curves across the nuisance parameter space via monotone regression. This allows us to compute the entire receiver-operating-characteristic (ROC) of the classifier for all $\boldsymbol{\nu} \in \mathcal{N}$ (Section 3.2 and Algorithm 2).

ii) **Nuisance-aware prediction sets (NAPS).** Rather than providing a point prediction based on an estimate of $\mathbb{P}_{\text{train}}(Y = 1 \mid \mathbf{X})$, we derive selection criteria that are valid under GLS and construct a *set-valued classifier* $\mathbf{H} : \mathbf{x} \mapsto \{\emptyset, 0, 1, \{0, 1\}\}$ which guarantees that the true label is included in the set with probability at least $(1 - \alpha)$, regardless of the true class y and of the value of the nuisance parameters $\boldsymbol{\nu}$. That is, the prediction sets $\mathbf{H}_\alpha(\mathbf{X})$ guarantee conditional validity under GLS (Theorem 2):

$$\mathbb{P}_{\text{target}}(Y \in \mathbf{H}_\alpha(\mathbf{X}) \mid y, \boldsymbol{\nu}) \geq 1 - \alpha, \ \forall y \in \mathcal{Y}, \ \boldsymbol{\nu} \in \mathcal{N}. \quad (1)$$

Standard point classifiers (e.g., the Bayes classifier; Appendix E) and prediction sets based on $\mathbb{P}_{\text{train}}(Y = 1 \mid \mathbf{X})$ are not conditionally valid across the nuisance parameter space, and hence are also not valid marginally under GLS. On the other hand, our algorithm returns valid NAPS for all levels $\alpha \in (0, 1)$ simultaneously given any new observation $\mathbf{x}_{\text{target}}$ without having to retrain the classifier. This

also yields marginal validity under GLS (Theorem 2). Our results do *not* rely on asymptotic theory with the number of observations $n \to \infty$. We only assume to have a sufficient number of simulations B to train and calibrate the classifier.

iii) **NAPS with higher power.** We show how one can further increase power while maintaining validity by constraining nuisance parameters given an observed $\mathbf{x}_{\text{target}}$ through $(1-\gamma)$ confidence sets of the nuisance parameters $\boldsymbol{\nu}$, where γ is a small pre-defined error level. This effectively allows to derive data-dependent cutoffs that decrease the average size of prediction sets given a specific observation.

We demonstrate our method using data from two high-fidelity scientific simulators: `scDesign3` (Song et al., 2023) which generates realistic single-cell RNA-sequencing data, and `CORSIKA` (Heck et al., 1998) which models the interactions of primary cosmic rays with the Earth's atmosphere. A flexible implementation of NAPS is available at https://github.com/lee-group-cmu/lf2i.

2. Related Work

To the best of our knowledge, this is the first work that estimates ROC curves across the entire parameter space $\Theta = \mathcal{Y} \times \mathcal{N}$. To construct *frequentist confidence sets*, we base our results directly on the class probability $\mathbb{P}_{\text{train}}(Y = 1 \mid \mathbf{X})$, rather than using a surrogate likelihood or likelihood ratio (see for example references in Cranmer et al. 2020). The idea of *improving power* of NAPS with $\gamma > 0$ is similar to Berger & Boos (1994), and close in spirit to likelihood profiling, with the key difference that profiling does not guarantee validity (even for a large number of simulations B and under no GLS), and also requires an approximation of the likelihood and the maximum likelihood estimate of $\boldsymbol{\nu}$. The ROC *calibration* framework of Section 3.2 is related to Zhao et al. (2021) and Dey et al. (2022), which use monotone regression to estimate the CDF of probability integral transforms for calibrating posterior probabilities, but not for constructing valid prediction sets under GLS. When the prior distribution over y in the target data is known, $\mathbb{P}_{\text{train}}(Y = 1 \mid \mathbf{X})$ can be easily recalibrated to match $\mathbb{P}_{\text{target}}(Y = 1 \mid \mathbf{X})$ under SLS (Saerens et al., 2002; Lipton et al., 2018). However, this is not possible under GLS since $\boldsymbol{\nu}$ is unknown at inference time. Moreover, our approach does not assume such a known prior. The construction of *set-valued classifiers* of Section 3.4 is inspired by Sadinle et al. (2019); Dalmasso et al. (2021); Masserano et al. (2023). There are also connections to *conformal prediction*: Conformal methods are widely used because they ensure prediction sets with marginal coverage when data are exchangeable (Papadopoulos et al., 2002; Vovk et al., 2005; Lei et al., 2018). However, conformal methods need adjustments under distributional shift when data are no longer exchangeable. Such adjustments need to be tailored for the type of shift at hand (Tibshirani et al., 2019). For instance, label shift can be addressed through label-conditional conformal prediction (Vovk et al., 2014; 2016; Sadinle et al., 2019), which guarantees coverage conditional on the label y (Podkopaev & Ramdas, 2021, Section 2.2) under SLS, but not under the presence of nuisance parameters and GLS. Finally, our work directly addresses the existing gap in methods for constructing *reliable simulator-based inference* algorithms with valid uncertainty quantification guarantees (Hermans et al., 2021). Our work is also inspired by the vast literature in high-energy physics on hypothesis testing and *nuisance-parameterized* machine-learning methods (Feldman & Cousins, 1998; Cousins, 2006; Sen et al., 2009; Chuang & Lai, 1998; Louppe et al., 2017; Cowan et al., 2011), which also includes the so-called "mining gold" idea of leveraging hidden information on latent variables in an all-knowing simulator (Brehmer et al., 2020).

3. Methodology

For simplicity, we will restrict our discussion to $Y \in \{0, 1\}$.

3.1. Classification as Hypothesis Testing

We reformulate the binary classification problem as a composite-versus-composite hypothesis test:

$$H_{0,y} : \boldsymbol{\theta} \in \Theta_0 \;\text{ versus }\; H_{1,y} : \boldsymbol{\theta} \in \Theta_1, \tag{2}$$

where $\Theta_0 = \{y\} \times \mathcal{N}$, $\Theta_1 = \{y\}^c \times \mathcal{N}$. We define

$$\tau_y(\mathbf{x}) = \frac{\mathbb{P}_{\text{train}}(Y = y \mid \mathbf{x})\, \mathbb{P}_{\text{train}}(Y \neq y)}{\mathbb{P}_{\text{train}}(Y \neq y \mid \mathbf{x})\, \mathbb{P}_{\text{train}}(Y = y)} \tag{3}$$

as our test statistic, which is equivalent to the Bayes factor for the test in Equation 2; see Appendix A for a derivation. Alternatively, one can define the test statistic as the probabilistic classifier $\mathbb{P}_{\text{train}}(Y = y|\mathbf{x})$ itself. Both quantities (which are related via a monotonic transformation) can be estimated directly from a *pre-trained* classifier based on $\mathcal{T}_B$. That is, there is no need for an extra step to, e.g., learn the likelihood function $\mathcal{L}(\mathbf{x}; Y, \boldsymbol{\nu})$ or the associated likelihood ratio statistic from simulated data as done in Cranmer et al. (2020), Rizvi et al. (2023), and references therein.

We denote the estimate of τ_y by $\widehat{\tau}_y$ and reject the null $H_{0,y}$ for small values of $\widehat{\tau}_y$. For example, if the null represents $y = 0$, then a "positive" case ($y = 1$) in binary classification would correspond to small values of $\widehat{\tau}_0$, or equivalently, large values of the probabilistic classifier $\widehat{\mathbb{P}}_{\text{train}}(Y = 1 \mid \mathbf{x}) = 1 - \widehat{\mathbb{P}}_{\text{train}}(Y = 0 \mid \mathbf{x})$. In this work, we define cutoffs for $\widehat{\tau}_y$ so that prediction sets are approximately valid under nuisance parameters and GLS.

3.2. The Rejection Probability Across the Entire Parameter Space

To choose the optimal cutoff to reject $H_{0,y}$ and construct valid prediction sets, we need to know how the classifier performs for different values of the nuisance parameters $\boldsymbol{\nu}$. The first step is to compute the following quantity:

Definition 1 (Rejection probability)**.** *Let λ be any test statistic, e.g, the estimated Bayes factor, $\lambda = \widehat{\tau}_y$. The rejection probability of λ is defined as*

$$W_\lambda(C; y, \boldsymbol{\nu}) := \mathbb{P}_{\textit{target}}\left(\lambda(\mathbf{X}) \leq C | y, \boldsymbol{\nu}\right), \tag{4}$$

where $y \in \{0, 1\}$, $\boldsymbol{\nu} \in \mathcal{N}$, and $C \in \mathbb{R}$.

For fixed $\boldsymbol{\nu}$ and null $H_{0,0} : Y = 0$, the receiver operating characteristic (ROC) relates the true positive rate

$$\texttt{TPR}(C; \boldsymbol{\nu}) := W_{\widehat{\tau}_0}(C; 1, \boldsymbol{\nu})$$

to the false positive rate

$$\texttt{FPR}(C; \boldsymbol{\nu}) := W_{\widehat{\tau}_0}(C; 0, \boldsymbol{\nu}),$$

while varying the cutoff C. Figure 3 shows examples of some ROC curves at different values of $\boldsymbol{\nu}$ when the null represents the negative class $y = 0$, for the setting of Section 5.3.

A key insight behind our method is that the rejection probability (Equation 4) is invariant under GLS even if estimated from p_{train}; in other words, it is always the same for train and target data (Lemma 1). As a result, our ROC curves reliably measure the performance of the classifier under nuisance parameters. In practice, we can estimate $W_\lambda(C; y, \boldsymbol{\nu})$ for all y and $\boldsymbol{\nu}$ simultaneously using regression with a monotonic constraint in C. The whole procedure is amortized with respect to the target data, meaning that both the base classifier and the rejection probability are estimated only once, after which they can be evaluated on an arbitrary number of observations.

3.3. Selecting the Optimal Cutoff under GLS

Once we know the classifier's rejection probability function, we can apply it in various ways. All our choices are robust against GLS.

Controlling FPR or TPR Based on $W_\lambda(C; y, \boldsymbol{\nu})$, we can find the cutoff C for a new test point that either controls type-I error (FPR), or guarantees a minimum recall (TPR), or maximizes some other metric of choice that depends on both FPR and TPR. For example, FPR control at some pre-specified level $\alpha \in [0, 1]$ and $\boldsymbol{\nu}_0 \in \mathcal{N}$ implies $C_\alpha = \texttt{FPR}^{-1}(\alpha; \boldsymbol{\nu}_0)$, and TPR control at some minimum recall α implies $\widetilde{C}_\alpha = \texttt{TPR}^{-1}(\alpha; \boldsymbol{\nu}_0)$. To control FPR or TPR *uniformly* over $\boldsymbol{\nu}$, one can instead choose $C_\alpha = \inf_{\boldsymbol{\nu} \in \mathcal{N}} \texttt{FPR}^{-1}(\alpha; \boldsymbol{\nu})$, and $\widetilde{C}_\alpha = \sup_{\boldsymbol{\nu} \in \mathcal{N}} \texttt{TPR}^{-1}(\alpha; \boldsymbol{\nu})$, respectively. Although robust under GLS, such cutoffs can be overly conservative.

Controlling FPR or TPR, but with more power An alternative approach, which is still valid for any $\boldsymbol{\nu}$ and can increase power, is to restrict the search over nuisance parameters to a smaller region of $\mathcal{N}$. For this approach, we first construct a confidence set $S(\mathbf{x}; \gamma)$ for $\boldsymbol{\nu}$ and fixed $y \in \{0, 1\}$ at a pre-specified $(1 - \gamma)$ level (Definition 3). This allows to choose a data-dependent cutoff such that

$$C^*_\alpha(\mathbf{x}) = \inf_{\boldsymbol{\nu} \in S(\mathbf{x}; \gamma)} \{\texttt{FPR}^{-1}(\beta; \boldsymbol{\nu})\},$$

where $\beta = \alpha - \gamma$, where the minimization is over the restricted set $S(\mathbf{x}; \gamma) \subseteq \mathcal{N}$. In practice, $S(\mathbf{x}; \gamma)$ can be either obtained from auxiliary measurements that are available at inference time, or from a separate pre-trained model that returns valid confidence sets on $\boldsymbol{\nu}$ from data $\mathbf{x}$. Lemma 1 demonstrates that this cutoff guarantees a maximum type-I error equal to α (FPR control) for any $\boldsymbol{\nu} \in \mathcal{N}$. Similarly, for TPR control, choosing $\widetilde{C}^*_\alpha(\mathbf{x}) = \sup_{\boldsymbol{\nu} \in S(\mathbf{x}; \gamma)} \texttt{TPR}^{-1}(\beta; \boldsymbol{\nu})$ with $\beta = \alpha + \gamma$ guarantees a minimum recall of at least α. The special case of $\gamma = 0$ (and $\beta = \alpha$) corresponds to $S(\mathbf{x}; \gamma) = \mathcal{N}$; that is, no constraints on the nuisance parameters. Finally, note that hybrid cut-offs $\texttt{FPR}^{-1}(\beta; \widehat{\boldsymbol{\nu}})$ and $\texttt{TPR}^{-1}(\beta; \widehat{\boldsymbol{\nu}})$ based on a *point prediction* $\widehat{\boldsymbol{\nu}}(\mathbf{x})$ of the nuisance parameters (such as the posterior mean) would not lead to valid uncertainty quantification under GLS (see Figure 17 in Appendix).

3.4. Constructing Robust Set-Valued Classifiers

Rather than just returning a single label $0/1$ for each observation $\mathbf{x}$ like the standard Bayes classifier (Appendix E), our method yields prediction sets from a set-valued classifier.

Definition 2 (Nuisance-aware prediction set)**.** *A nuisance-aware prediction set (NAPS) is the set returned from a set-valued classifier $\mathbf{H} : \mathbf{x} \mapsto \{\emptyset, 0, 1, \{0, 1\}\}$ with*

$$\mathbf{H}(\mathbf{x}; \alpha) = \left\{y \in \{0, 1\} \mid \widehat{\tau}_y(\mathbf{x}) > C^*_{\alpha, y}(\mathbf{x})\right\}, \tag{5}$$

where

$$C^*_{\alpha, y}(\mathbf{x}) = \inf_{\boldsymbol{\nu} \in S_y(\mathbf{x}; \gamma)} \{W^{-1}_{\widehat{\tau}_y}(\beta; y, \boldsymbol{\nu})\}, \tag{6}$$

is the rejection cutoff, $\beta = \alpha - \gamma$ and $S_y(\mathbf{x}; \gamma)$ is a $(1 - \gamma)$ confidence set for $\boldsymbol{\nu}$ defined by Equation 7.

This classifier guarantees user-defined levels of coverage $1 - \alpha$ (the probability that the true label is included in the set), no matter what the true class y and the nuisance parameters $\boldsymbol{\nu}$ are (Theorem 2). The resulting prediction sets contain all labels that were not rejected by the corresponding hypothesis test. Ambiguous sets can arise in two cases: *i)*

Algorithm 1 `Nuisance-aware prediction sets`

Input: training set $\mathcal{T} = \{(Y_i, \mathbf{X}_i)\}_{i=1}^{B}$; calibration set $\mathcal{T}' = \{(Y_i', \boldsymbol{\nu}_i', \mathbf{X}_i')\}_{i=1}^{B'}$; observation $\mathbf{x}$; test statistic $\lambda = \tau_y$; miscoverage levels $\alpha \in [0,1]$ and $\gamma \in [0, \alpha]$.
Output: Prediction set $H_\alpha(\mathbf{x})$ such that Equation 1 holds.

// Training
Estimate $\mathbb{P}_{\text{train}}(Y = y \mid \mathbf{X})$ via a probabilistic classifier
// Calibration
Estimate $W_{\tau_y}(C; y, \boldsymbol{\nu}) := \mathbb{P}_{\text{target}}(\tau_y(\mathbf{X}) \leq C \mid y, \boldsymbol{\nu})$ as detailed in Algorithm 2 by
  i. Computing the test statistic $\widehat{\tau}_y(\mathbf{x})$ as in Equation 3 for all $\mathbf{X} \in \mathcal{T}'$;
  ii. Constructing the augmented calibration set $\mathcal{T}''$;
  iii. Estimating the rejection probability function $W_{\widehat{\tau}_y}(C; y, \boldsymbol{\nu})$ from $\mathcal{T}''$ via monotone regression.
// Inference
for $y \in \{0,1\}$ **do**
  Compute $\widehat{\tau}_y(\mathbf{x})$ as in Equation 3
  if $\gamma = 0$ **then**
    $C^*_{\alpha,y}(\mathbf{x}) \leftarrow \inf_{\boldsymbol{\nu} \in \mathcal{N}} \{\widehat{W}^{-1}_{\widehat{\tau}_y}(\alpha; y, \boldsymbol{\nu})\}$
  else
    Constrain nuisance parameters by constructing a level-γ confidence set $S_y(\mathbf{x}; \gamma)$ for $\boldsymbol{\nu}$
    $C^*_{\alpha,y}(\mathbf{x}) \leftarrow \inf_{\boldsymbol{\nu} \in S_y(\mathbf{x};\gamma)} \{\widehat{W}^{-1}_{\widehat{\tau}_y}(\alpha - \gamma; y, \boldsymbol{\nu})\}$
  end if
end for
$\mathbf{H}(\mathbf{x}; \alpha) \leftarrow \{y \in \{0,1\} \mid \widehat{\tau}_y(\mathbf{x}) > C^*_{\alpha,y}(\mathbf{x})\}$ prediction set $\mathbf{H}(\mathbf{x}; \alpha)$ for Y

When both null hypotheses are rejected, we obtain an empty set. However, empty sets only arise at very low confidence levels (high values of α), which is typically not considered an interesting regime; *ii)* When both null hypotheses are accepted, we obtain a prediction set that includes both 0 and 1. This latter type of ambiguity reflects the uncertainty of the classifier, which typically grows at higher confidence levels (low values of α). A low-quality classifier will often report an "I-don't-know answer" for ambiguous instances if forced to guarantee a certain confidence level, rather than returning a $0/1$ answer that has a high chance of being incorrect.

While $\gamma = 0$ can be the default choice for NAPS, choosing a small $\gamma > 0$ often leads to higher power (see Section 5). Finally, note that while our set-valued classifier targets conditional coverage under GLS according to Equation 1, as a by-product we also achieve prediction sets with marginal coverage under GLS (see Theorem 2).

Algorithm 1 includes a step-by-step description of the entire procedure for constructing nuisance-aware prediction sets.

4. Theoretical Results

Proofs for this section can be found in Appendix B.

4.1. Validity and Robustness to GLS

Lemma 1 (Invariance of the Rejection Probability to GLS)**.** *Under GLS, the rejection probability (Definition 1) of any test statistic λ is invariant to GLS, that is*

$$W_\lambda(C; y, \boldsymbol{\nu}) = \mathbb{P}_{\textit{target}}(\lambda(\mathbf{X}) \leq C \mid y, \boldsymbol{\nu}) = \mathbb{P}_{\textit{train}}(\lambda(\mathbf{X}) \leq C \mid y, \boldsymbol{\nu}).$$

4.1.1. Nuisance-Aware Cutoffs

Definition 3 (Confidence set for nuisance parameters)**.** *The random set $S_y(\mathbf{x}; \gamma)$ is a valid $(1-\gamma)$ level confidence set for $\boldsymbol{\nu}$ at fixed $y \in \{0,1\}$, if*

$$\mathbb{P}_{\textit{target}}(\boldsymbol{\nu} \in S_y(\mathbf{X}; \gamma) \mid y, \boldsymbol{\nu}) \geq 1 - \gamma, \quad \forall \boldsymbol{\nu} \in \mathcal{N}, \tag{7}$$

for some pre-specified value $\gamma \in [0,1]$.

The following theorem shows that nuisance-aware cutoffs control FPR and TPR at the specified level.

Theorem 1 (Nuisance-aware cutoffs for FPR/TPR control)**.** *Choose a threshold $\alpha \in [0,1]$ and $\gamma \in [0, \alpha]$. Let $S_y(\mathbf{x}; \gamma)$ be a valid $(1-\gamma)$ confidence set for $\boldsymbol{\nu}$ at fixed $y \in \{0,1\}$ according to Definition 3. Let $\lambda(\mathbf{X})$ be any test statistic that measures how plausible it is that $\mathbf{X}$ was generated from $H_{0,y}$. Define the nuisance-aware rejection cutoff to be*

$$C^*_{\alpha,y}(\mathbf{x}) = \inf_{\boldsymbol{\nu} \in S_y(\mathbf{x};\gamma)} \{W^{-1}_\lambda(\beta; y, \boldsymbol{\nu})\}, \tag{8}$$

where $\beta = \alpha - \gamma$, and W is the rejection probability in Definition 1. Then, for all $\boldsymbol{\nu} \in \mathcal{N}$, we have FPR control:

$$\mathbb{P}_{\textit{target}}\left(\lambda(\mathbf{X}) \leq C^*_{\alpha,y}(\mathbf{X}) \mid y, \boldsymbol{\nu}\right) \leq \alpha \tag{9}$$

(maximum type-I error probability for $H_{0,y}$).

Similarly, if

$$\widetilde{C}^*_{\alpha,y}(\mathbf{x}) = \sup_{\boldsymbol{\nu} \in S_{1-y}(\mathbf{x};\gamma)} \{W^{-1}_\lambda(\beta; 1-y, \boldsymbol{\nu})\},$$

with $\beta = \alpha + \gamma$, then for all $\boldsymbol{\nu} \in \mathcal{N}$, we have TPR control:

$$\mathbb{P}_{\textit{target}}\left(\lambda(\mathbf{X}) \leq \widetilde{C}^*_{\alpha,y}(\mathbf{X}) \mid 1-y, \boldsymbol{\nu}\right) \geq \alpha$$

(minimum recall for $H_{0,y}$).

4.1.2. Properties of the Nuisance-Aware Prediction Set

The nuisance-aware prediction set (Definition 2) is both *conditionally* and *marginally* valid with respect to both y and $\boldsymbol{\nu}$ under GLS.

Theorem 2. *Let $\mathbf{H}(\mathbf{x}; \alpha)$ be the nuisance-aware prediction set of Definition 2. Under GLS, for every $y \in \{0, 1\}$ and $\nu \in \mathcal{N}$*

$$\mathbb{P}_{target}(Y \in \mathbf{H}(\mathbf{X}; \alpha) \mid y, \boldsymbol{\nu}) \geq 1 - \alpha.$$

Moreover,

$$\mathbb{P}_{target}(Y \in \mathbf{H}(\mathbf{X}; \alpha)) \geq 1 - \alpha.$$

5. Experiments

5.1. Synthetic Example

Consider a simplified setting where we are certain about the data-generating process of $Y = 1$ cases of interest, but not about that of $Y = 0$ cases. We assume

$$p(x_i \mid Y_i = 1) = \frac{e^{x_i}}{e - 1}$$
$$p(x_i \mid Y_i = 0, \nu_i) = \frac{\nu_i e^{-\nu_i x_i}}{1 - e^{-\nu_i}},$$

where $\nu \in [1, 10]$ is a nuisance parameter, which enlarges the model for $Y = 0$ to reflect our uncertainty of how cases of no direct interest might manifest themselves.

Before Data Collection Before having specific knowledge about target data and experimental conditions, we decide to draw ν from a uniform reference distribution $p_{\text{train}}(\nu) = \mathcal{U}[1, 10]$ (here $\mathbb{P}_{\text{train}}(Y = 1) = \mathbb{P}_{\text{target}}(Y = 1) = 0.5$ is fixed). We then pre-train a classifier[1] and compute the class posterior $\mathbb{P}_{\text{train}}(Y = 1|x)$, and construct $(1 - \alpha)$ prediction sets

$$R_\alpha(x) := \{y : \mathbb{P}_{\text{train}}(Y = y \mid x) > C^*_\alpha\} \tag{10}$$

with cutoffs

$$C^*_\alpha \text{ s.t. } \mathbb{P}_{\text{train}}\left(\mathbb{P}_{\text{train}}(Y = y \mid X) \leq C^*_\alpha\right) = \alpha,$$

for a pre-specified miscoverage level α. These are the oracle prediction sets that minimize ambiguity (i.e., average size) subject to having the correct total coverage according the Theorem 1 from Sadinle et al. (2019). We will henceforth refer to them as "standard prediction sets" to distinguish them from the oracle class-conditional prediction sets from Sadinle et al. (2019) and NAPS.

Setting 1: No GLS When train and target data have the same distributions, the prediction sets $R_\alpha(X)$ have guaranteed marginal coverage

$$\mathbb{P}_{\text{train}}(Y \in R_\alpha(X)) = 1 - \alpha$$

at the nominal $(1 - \alpha)$ level by construction (red curve overlapping black bisector in Figure 1, top left), although they might still undercover in specific regions of the nuisance parameter space (see Figure 17 in Appendix I). NAPS with $\gamma = 0$ are instead both marginally valid (blue curve, top left) and conditionally valid (Theorem 2). The latter "universality" can cause overly conservative prediction sets and a loss of power (defined as the probability of rejecting $H_{0,y} : Y = y$ when $Y \neq y$); see bottom left panel.

Setting 2: With GLS Suppose now that we apply the pre-trained classifier to a target distribution with a *different* distribution over the nuisance parameters, namely $p_{\text{target}}(\nu) = N(4, 0.1) \neq p_{\text{train}}(\nu)$. The top right panel of Figure 1 shows that the prediction sets $R_\alpha(X)$ are no longer valid even marginally (red curve below bisector), whereas NAPS are still valid. Moreover, we can achieve higher power by constraining the optimization to a high-confidence set of the nuisance parameter (compare green with blue NAPS curves). In summary: our proposed method can leverage the original $\mathbb{P}_{\text{train}}(Y = 1 \mid \mathbf{x})$ classifier to provide prediction sets that are both valid and precise for any distribution $p(y, \nu)$ as long as $x|y, \nu$ stays the same. Additional results for other prediction set methods and NAPS with $\gamma > 0$ are available in Appendix I.

5.2. Single-Cell RNA Sequencing

RNA sequencing, or RNA-Seq, is a vital technique in genetics and genomics research that has revolutionized our understanding of gene expression. Many RNA-seq experiments involve extracting RNA from target cells and examining counts of specific genes. While the natural variation in gene counts between different types of cells is interesting to researchers, the observed gene counts depend also on the precise steps of the sequencing process. For example, the exact chemicals, equipment, room temperature and lab technician can greatly influence the final measurements, in addition to the cell type. In practice, these so-called "batch effects" are often unmeasured confounders whose exact value is unknown at the inference stage. Thus, analysis of experimental gene counts must take them into account in order to conduct reliable scientific analysis. In what follows, we define a "batch protocol" to be a particular set of these conditions common to a batch of cells.

We use data from the recently proposed `scDesign3` simulator (Song et al., 2023), with reference data taken from the PBMC Systematic Comparative Analysis (Ding et al., 2019). We consider two cell types (`CD4`$^+$ T-cells and `Cytotoxic` T-cells) and a subset of 100 random genes. The reference data contains counts from two separate experiments, which will serve as the basis of our simulated batch protocols. We use the two original experimental conditions as well as two artificial perturbations derived from them to generate

[1] In this simplified example we can actually compute everything semi-analytically.

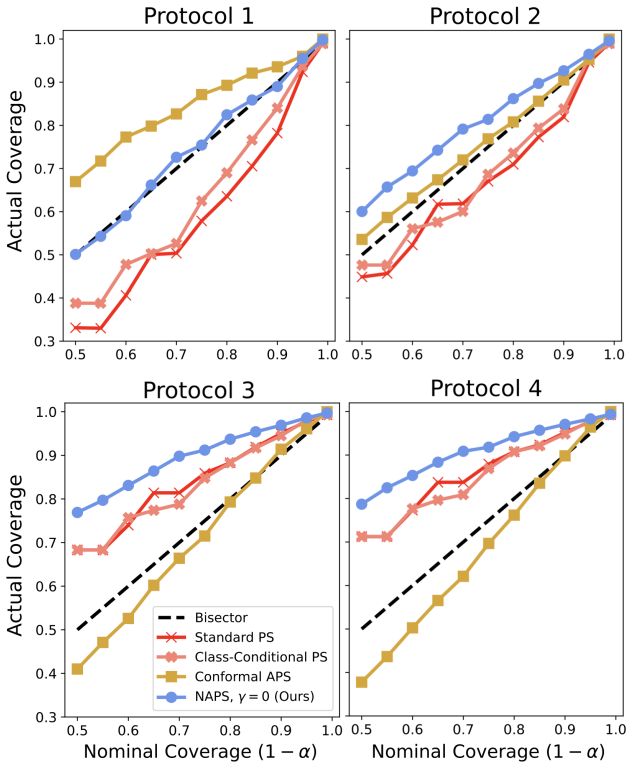

Figure 2. **Coverage under different batch protocols ν for the RNA-Seq example.** Each marker represents the proportion of samples in the test set whose true label was included in the constructed prediction sets. Nuisance-aware prediction sets (NAPS $\gamma = 0$; blue) are valid regardless of the protocol, which is unknown at inference time. All other methods for prediction sets with marginal coverage (red), class-conditional coverage (pink), and conformal adaptive prediction sets (gold) undercover for at least two batch protocols.

four possible batch protocols. Following our terminology, this corresponds to a discrete nuisance parameter with four groups. We consider the setup of a classifier trained on data from all four possible protocols and tested on different $\mathbf{x}_{\text{target}}$ whose true protocol value is unknown (in addition to the cell type). In total, we have available $80{,}000$ samples which we divide into train (60%), calibration (35%) and test (5%) sets. Our goal is to infer the cell's type from the observed gene count under the presence of the unknown nuisance parameter.

We compare our method with three baselines: (i) standard prediction sets for which cutoffs are computed from $\mathbb{P}(Y|\mathbf{X})$ (Sadinle et al., 2019, Theorem 1); (ii) class-conditional prediction sets with cutoffs derived separately from each $\mathbb{P}(Y = i|\mathbf{X})$, $i \in \{0, 1\}$ (Sadinle et al., 2019); and (iii) conformal adaptive prediction sets (APS; Romano et al. (2020)). Figure 2 shows that nuisance-aware prediction sets (NAPS) are valid regardless of the protocol, which is unknown at inference time. On the other hand, all of the other prediction sets from the analyzed baselines undercover for at least two protocols. Nuisance-aware cutoffs need to control type-I error for every single value of the nuisance parameter, including the hardest case. Here, Protocol 1 (top left) appears to be the most difficult to classify correctly. Finally, we note that while conformal APS approximately achieves coverage for $(1-\alpha) \approx 1$, this comes at the expense of uninformative prediction sets that contain both labels for all $\mathbf{x}_{\text{target}}$. NAPS, on the other hand, is able to maintain high power (see Figure 10 in Appendix G). Additional results and details on the base classifier, the model used to estimate the rejection probability function, and the baselines adopted for comparison can be found in Appendix G.

5.3. Atmospheric Cosmic-Ray Showers

High-energy cosmic rays, both charged and neutral, are extremely informative probes of astrophysical sources in our galaxy and beyond. Gamma rays (which constitute the vast majority of neutral cosmics) reach the Earth atmosphere from specific directions that coincide with the location of the originating source in the sky. On the other hand, charged cosmic rays (hadrons) arrive from non-informative directions as they get deflected by galactic magnetic fields while travelling. An important step in analyzing gamma-ray sources is to separate gamma-induced showers (G) from the very large background ($> 1000 : 1$) of hadron-induced showers (H) using ground-based detector arrays that collect particles $\mathbf{x}$ from secondary showers (Dorigo et al. (2023); see top left of Figure 4 for an illustration). G/H separation is a challenging

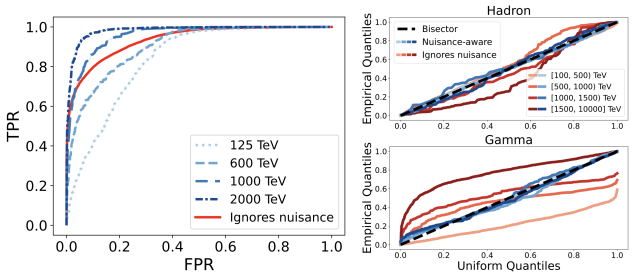

Figure 3. **Dependence of the ROC on the energy of the cosmic-ray shower.** *Left:* Receiver operating characteristic evaluated according to our method at different energy values (shades of blue). By estimating the entire ROC, we can control FPR or TPR at specified confidence levels for all $\boldsymbol{\nu} \in \mathcal{N}$, which is not possible with the "marginal" ROC curve (red). *Right:* Diagnostic P-P plot evaluated at four bins over energy for nuisance-aware ROC (shades of blue) and ROC that ignores nuisances (shades of red). To check if $\mathbb{P}_{\text{target}}(\lambda(\mathbf{X}) \leq C|y, \boldsymbol{\nu})$ is well estimated, we plot PIT values against a $\mathcal{U}(0, 1)$ distribution (dashed bisector; see Appendix D for details). This is clearly not the case if one ignores nuisance parameters.

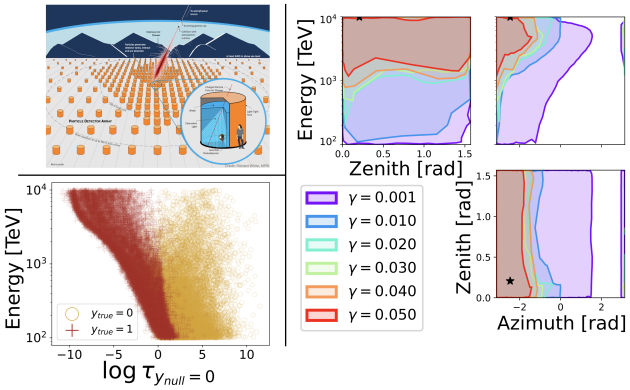

Figure 4. **Constraining the cosmic ray shower parameters.** *Top left*: Illustration of the Southern Wide-field Gamma-ray Observatory (SWGO; Abreu et al. (2019); image credit: Richard White) array of detectors with an incoming gamma ray (red). *Bottom Left*: Test statistic under $y_0 = 0$ (hadron) as a function of energy. At high energies, the class-conditional test statistics are well separated, implying that it is easier to distinguish gamma showers (red) from hadron showers (gold). *Right*: Confidence set for $\boldsymbol{\nu}$ at different $(1-\gamma)$ confidence levels obtained via the framework of Masserano et al. (2023). The true value of $\boldsymbol{\nu}$ is the black star.

rare-event detection problem, where the true distribution of both the shower type Y and the shower parameters $\boldsymbol{\nu}$ might be misspecified in simulated data. Our goal is to infer the cosmic ray identity Y from ground measurements $\mathbf{X}$ while accounting for additional shower parameters: energy E, azimuth angle A and zenith angle Z. Together, these form a nuisance parameter vector $\boldsymbol{\nu} = (E, A, Z)$. We construct a data set of 99,850 samples simulated from `CORSIKA` (Heck et al., 1998) divided into train (45%), calibration (45%) and test (10%) sets. Figure 3 (left) shows several ROC curves as a function of different energy values, demonstrating a clear dependency of the classification problem on this shower parameter.

Figure 5 summarizes our results as a function of the confidence level $(1-\alpha)$ for different classification metrics. These are computed within true and within predicted gamma rays for two different bins whose border is the median energy level. Nuisance-aware prediction sets (NAPS with $\gamma = 0$) achieve high precision and low false discovery rates but slightly under-perform relative to the standard Bayes classifier (Appendix E) for lower energy values (left column in Figure 5), specifically at low confidence levels. This behaviour originates from the complexity of the data: at lower energies it is indeed much harder to distinguish gamma rays from hadrons (see bottom left panel of Figure 4).

By constructing $(1-\gamma)$ confidence sets for $\boldsymbol{\nu}$ (see the right panel of Figure 4 for an example), we are able to outperform the standard Bayes classifier at all confidence levels (NAPS with $\gamma > 0$). This result is explained by the bottom panel in Figure 5: NAPS predicts a single label only when it is relatively certain about it, and otherwise outputs an ambiguous prediction set that contains both labels. Nonetheless, for this example, NAPS with $\gamma = 0$ is able to achieve a higher number of true positives and lower number of false

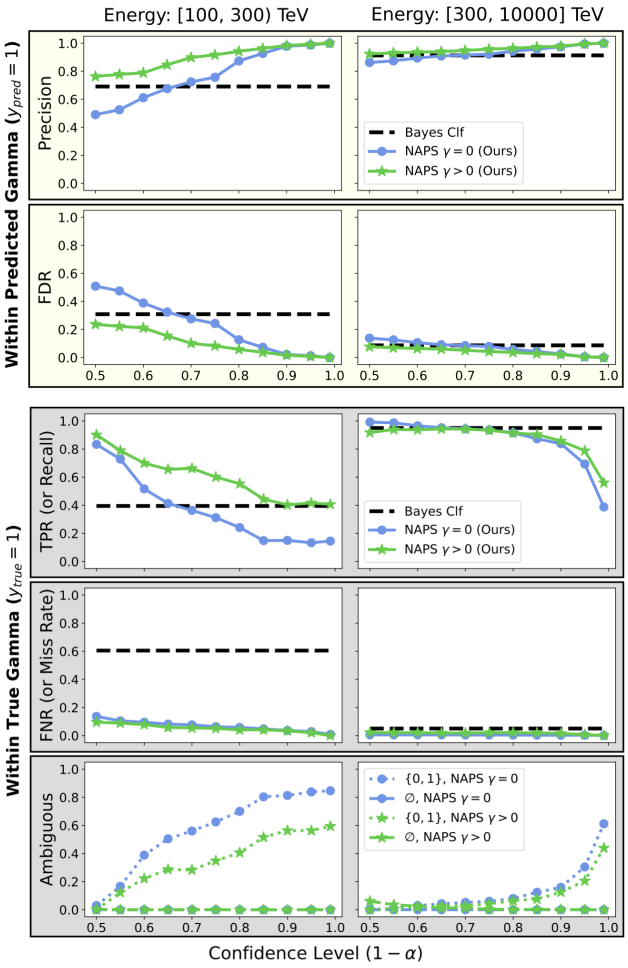

Figure 5. **Classification metrics within true and within predicted Gamma rays** ($y = 1$)**.** Results are binned according to whether the shower energy is below (left) or above (right) the median value. *Top panel:* Nuisance-aware prediction sets (NAPS $\gamma = 0$; blue) achieve high precision and low false discovery rates (FDR), especially at high confidence levels. In addition, by constraining the nuisance parameters $\boldsymbol{\nu} = (E, A, Z)$, we can increase performance (NAPS $\gamma > 0$; green) with uniformly better results relative to the standard Bayes classifier (black dashed line). *Bottom panel:* Our set-valued classifier makes explicit its level of uncertainty on the label y by returning ambiguous prediction sets (bottom row) for hard-to-classify $\mathbf{x}_{\text{target}}$. Even so, NAPS with $\gamma > 0$ is able to achieve a higher number of true positives and lower number of false negatives relative to the Bayes classifier. Here $\gamma = \alpha \times 0.3$.

negatives relative to the Bayes classifier. Additional results and details on the models used can be found in Appendix F.

6. Conclusion and Discussion

The introduction of nuisance parameters complicates the effectiveness and reliability of machine learning models in tasks such as classification. This paper introduces a new method for handling prior probability shift of both label and nuisance parameters in likelihood-free inference when a high-fidelity mechanistic model is available. We demonstrate a new technique for estimating the ROC across the entire parameter space for binary classification problems. We also show how to construct set-valued classifiers that have a guaranteed user-specified probability $(1-\alpha)$ of including the true label (parameter of interest), for all levels $\alpha \in [0,1]$ simultaneously, without having to retrain the model for every α. These set-valued classifiers are valid, no matter what the true label and unknown nuisance parameters are. Finally, we demonstrate how to increase power while maintaining validity by constraining nuisance parameters.

Extensions and Limitations. Our approach can be extended to standard classification problems where the training data does not come from a simulator, as long as (i) the nuisance parameters $\boldsymbol{\nu}$ in the data-generating process have been identified and are available at training time, and (ii) we can reliably estimate the rejection probability function across the entire parameter space as in Section 3.2. We recommend checking the latter with diagnostic P-P plots (see Appendix D, and Figure 3 (right) for an example).

NAPS directly extends to multiclass as one-vs-one problems, since we can estimate one-vs-one ROC curves for each $\boldsymbol{\nu} \in \mathcal{N}$. The computational cost for K classes would increase by a factor of $\binom{K}{2}$. However, an extension to multiclass as one-vs-rest problems is non-trivial, because estimating ROC curves requires knowledge of the distribution of labels Y on the target set for every nuisance parameter ν. Without such knowledge, the ROC curves would not be invariant to GLS.

NAPS achieves validity under GLS. However, in the absence of a shift, this results in reduced power compared to standard prediction sets (Equation 10). Although we can recover some of this power by constraining nuisance parameters (i.e. setting $\gamma > 0$), the cutoffs need to be computed for *each* test point, which can be computationally expensive, especially for high-dimensional $\boldsymbol{\nu}$. Furthermore, setting $\gamma > 0$ is not guaranteed to increase power relative to $\gamma = 0$: Since rejection probability inversion is performed at level $\alpha - \gamma$, power might *decrease* when optimizing the NAPS cutoff over the $(1-\gamma)$ confidence set for $\boldsymbol{\nu}$ (see Equation 8). This can occur if the $(1-\gamma)$ confidence sets are too large, or when the distribution of $\boldsymbol{\nu}$ is skewed toward certain regions (Figure 18). For further discussion, refer to Appendix I.4.

Finally, we note that NAPS may sometimes result in empty prediction sets, though this is uncommon when $(1-\alpha)$ is large. Future adaptations could incorporate strategies from Sadinle et al. (2019) to mitigate this issue.

Acknowledgements

We thank Larry Wasserman and Mikael Kuusela for their feedback on earlier versions of the work. We are also grateful to Federico Nardi and Will Townes for their help with the cosmic rays and RNA sequencing examples, respectively. ABL and AS are partially supported by NSF DMS-2053804. RI is partially supported by FAPESP-2023/07068-1 and CNPq-305065/2023-8.

Impact Statement

In the physical and biological sciences, nuisance parameters are often needed to account for limitations in the modeling of the underlying processes. However, their inclusion reduces the effectiveness of machine learning and statistical procedures. Nuisance parameters (sometimes also referred to as systematic uncertainties) are one of the main factors limiting the precision and discovery reach of scientific analyses. Our work addresses this issue and could have a broader impact on reliable scientific discovery.

References

Abreu, P., Albert, A., Alfaro, R., Alvarez, C., Arceo, R., Assis, P., Barao, F., Bazo, J., Beacom, J., Bellido, J., et al. The southern wide-field gamma-ray observatory (swgo): A next-generation ground-based survey instrument for vhe gamma-ray astronomy. *arXiv preprint arXiv:1907.07737*, 2019.

Berger, R. L. and Boos, D. D. P values maximized over a confidence set for the nuisance parameter. *Journal of the American Statistical Association*, 89(427):1012–1016, 1994.

Brehmer, J., Louppe, G., Pavez, J., and Cranmer, K. Mining gold from implicit models to improve likelihood-free inference. *Proceedings of the National Academy of Sciences*, 117(10):5242–5249, 2020.

Brent, R. P. *Algorithms for minimization without derivatives*. Courier Corporation, 2013.

Chuang, C.-S. and Lai, T. L. Resampling methods for confidence intervals in group sequential trials. *Biometrika*, 85(2):317–332, 06 1998. ISSN 0006-3444. doi: 10.1093/biomet/85.2.317. URL https://doi.org/10.1093/biomet/85.2.317.

Cook, S. R., Gelman, A., and Rubin, D. B. Validation of

software for bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 15 (3):675–692, 2006.

Cousins, R. D. Treatment of nuisance parameters in high energy physics, and possible justifications and improvements in the statistics literature. In *Statistical Problems In Particle Physics, Astrophysics And Cosmology*, pp. 75–85. World Scientific, 2006.

Cowan, G., Cranmer, K., Gross, E., and Vitells, O. Asymptotic formulae for likelihood-based tests of new physics. *The European Physical Journal C*, 71:1–19, 2011.

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

Dalmasso, N., Masserano, L., Zhao, D., Izbicki, R., and Lee, A. B. Likelihood-free frequentist inference: Confidence sets with correct conditional coverage. *arXiv preprint arXiv:2107.03920*, 2021.

Dey, B., Zhao, D., Newman, J. A., Andrews, B. H., Izbicki, R., and Lee, A. B. Calibrated predictive distributions via diagnostics for conditional coverage. *arXiv preprint arXiv:2205.14568*, 2022.

Ding, J., Adiconis, X., Simmons, S. K., Kowalczyk, M. S., Hession, C. C., Marjanovic, N. D., Hughes, T. K., Wadsworth, M. H., Burks, T., Nguyen, L. T., et al. Systematic comparative analysis of single cell rna-sequencing methods. *BioRxiv*, pp. 632216, 2019.

Dorigo, T. and de Castro, P. Dealing with nuisance parameters using machine learning in high energy physics: a review. *arXiv preprint arXiv:2007.09121*, 2020.

Dorigo, T., Aehle, M., Donini, J., Doro, M., Gauger, N. R., Izbicki, R., Lee, A., Masserano, L., Nardi, F., Shen, A., et al. End-to-end optimization of the layout of a gamma ray observatory. *arXiv preprint arXiv:2310.01857*, 2023.

D'Isanto, A. and Polsterer, K. L. Photometric redshift estimation via deep learning-generalized and pre-classification-less, image based, fully probabilistic redshifts. *Astronomy & Astrophysics*, 609:A111, 2018.

Fawcett, T. and Flach, P. A. A response to webb and ting's on the application of roc analysis to predict classification performance under varying class distributions. *Machine Learning*, 58:33–38, 2005.

Feldman, G. J. and Cousins, R. D. Unified approach to the classical statistical analysis of small signals. *Physical Review D*, 57(7):3873–3889, Apr 1998. ISSN 1089-4918. doi: 10.1103/physrevd.57.3873.

Freeman, P. E., Izbicki, R., and Lee, A. B. A unified framework for constructing, tuning and assessing photometric redshift density estimates in a selection bias setting. *Monthly Notices of the Royal Astronomical Society*, 468 (4):4556–4565, 2017.

Heck, D., Knapp, J., Capdevielle, J., Schatz, G., Thouw, T., et al. Corsika: A monte carlo code to simulate extensive air showers. *Report fzka*, 6019(11), 1998.

HEP ML Community. A Living Review of Machine Learning for Particle Physics. URL https://iml-wg.github.io/HEPML-LivingReview/.

Hermans, J., Delaunoy, A., Rozet, F., Wehenkel, A., and Louppe, G. Averting a crisis in simulation-based inference. *stat*, 1050:14, 2021.

Izbicki, R., Lee, A. B., and Freeman, P. E. Photo-z estimation: An example of nonparametric conditional density estimation under selection bias. *The Annals of Applied Statistics*, 2017.

Kitching, T., Amara, A., Abdalla, F., Joachimi, B., and Refregier, A. Cosmological systematics beyond nuisance parameters: form-filling functions. *Monthly Notices of the Royal Astronomical Society*, 399(4):2107–2128, 2009.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

Lipton, Z., Wang, Y.-X., and Smola, A. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pp. 3122–3130. PMLR, 2018.

Louppe, G., Kagan, M., and Cranmer, K. Learning to pivot with adversarial networks. *Advances in neural information processing systems*, 30, 2017.

Masserano, L., Dorigo, T., Izbicki, R., Kuusela, M., and Lee, A. Simulator-based inference with WALDO: Confidence regions by leveraging prediction algorithms and posterior estimators for inverse problems. In *International Conference on Artificial Intelligence and Statistics*, pp. 2960–2974. PMLR, 2023.

Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., and Herrera, F. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012.

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

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Podkopaev, A. and Ramdas, A. Distribution-free uncertainty quantification for classification under label shift. In *Uncertainty in Artificial Intelligence*, pp. 844–853. PMLR, 2021.

Polo, F. M., Izbicki, R., Lacerda Jr, E. G., Ibieta-Jimenez, J. P., and Vicente, R. A unified framework for dataset shift diagnostics. *Information Sciences*, pp. 119612, 2023.

Pouget, A., Beck, J. M., Ma, W. J., and Latham, P. E. Probabilistic brains: knowns and unknowns. *Nature neuroscience*, 16(9):1170–1178, 2013.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. *Dataset shift in machine learning*. Mit Press, 2008.

Rizvi, S., Pettee, M., and Nachman, B. Learning likelihood ratios with neural network classifiers. *arXiv preprint arXiv:2305.10500*, 2023.

Romano, Y., Sesia, M., and Candes, E. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591, 2020.

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

Saerens, M., Latinne, P., and Decaestecker, C. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002.

Sen, B., Walker, M., and Woodroofe, M. On the unified method with nuisance parameters. *Statistica Sinica*, 19 (1):301–314, 2009. ISSN 10170405, 19968507.

Song, D., Wang, Q., Yan, G., Liu, T., Sun, T., and Li, J. J. scdesign3 generates realistic in silico data for multimodal single-cell and spatial omics. *Nature Biotechnology*, pp. 1–6, 2023.

Storkey, A. et al. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, 30(3-28):6, 2009.

Storn, R. and Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11:341–359, 1997.

Tibshirani, R. J., Foygel Barber, R., Candes, E., and Ramdas, A. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Vaz, A. F., Izbicki, R., and Stern, R. B. Quantification under prior probability shift: The ratio estimator and its extensions. *The Journal of Machine Learning Research*, 20(1):2921–2953, 2019.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Vovk, V., Petej, I., and Fedorova, V. From conformal to probabilistic prediction. In *Artificial Intelligence Applications and Innovations: AIAI 2014 Workshops: CoPA, MHDW, IIVC, and MT4BD, Rhodes, Greece, September 19-21, 2014. Proceedings 10*, pp. 221–230. Springer, 2014.

Vovk, V., Fedorova, V., Nouretdinov, I., and Gammerman, A. Criteria of efficiency for conformal prediction. In *Conformal and Probabilistic Prediction with Applications: 5th International Symposium, COPA 2016, Madrid, Spain, April 20-22, 2016, Proceedings 5*, pp. 23–39. Springer, 2016.

Vovk, V. et al. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

Zhao, D., Dalmasso, N., Izbicki, R., and Lee, A. B. Diagnostics for conditional density models and bayesian inference algorithms. In *Uncertainty in Artificial Intelligence*, pp. 1830–1840. PMLR, 2021.

A. The Bayes Factor as a Frequentist Test Statistic

In this work, we treat the Bayes factor as a frequentist test statistic, similar to the Bayes Frequentist Factor (`BFF`) method in (Dalmasso et al., 2021). Consider the composite-versus-composite hypothesis test:

$$H_{0,y} : \boldsymbol{\theta} \in \Theta_0 \;\text{ versus }\; H_{1,y} : \boldsymbol{\theta} \in \Theta_1 \tag{11}$$

where $\Theta_0 = \{y\} \times \mathcal{N}$, $\Theta_1 = \{y\}^c \times \mathcal{N}$, and $y \in \{0, 1\}$. The Bayes factor of the test is defined as

$$\tau_y(\mathbf{x}) := \frac{\mathbb{P}'(\mathbf{x}|H_{0,y})}{\mathbb{P}'(\mathbf{x}|H_{1,y})} = \frac{\int_{\mathcal{N}} \mathcal{L}(\mathbf{x}; y, \boldsymbol{\nu})\; p'(\boldsymbol{\nu}|y)\; d\boldsymbol{\nu}}{\int_{\mathcal{N}} \mathcal{L}(\mathbf{x}; 1-y, \boldsymbol{\nu})\; p'(\boldsymbol{\nu}|1-y)\; d\boldsymbol{\nu}}$$

By Bayes theorem,

$$\begin{aligned}\tau_y(\mathbf{x}) &= \frac{\int_{\mathcal{N}} \frac{p'(y,\boldsymbol{\nu}|\mathbf{x})}{p'(y,\boldsymbol{\nu})} p'(\boldsymbol{\nu}|y)\; d\boldsymbol{\nu}}{\int_{\mathcal{N}} \frac{p'(1-y,\boldsymbol{\nu}|\mathbf{x})}{p'(1-y,\boldsymbol{\nu})} p'(\boldsymbol{\nu}|1-y)\; d\boldsymbol{\nu}} = \frac{\int_{\mathcal{N}} \frac{p'(y,\boldsymbol{\nu}|\mathbf{x})}{\mathbb{P}'(Y=y)}\; d\boldsymbol{\nu}}{\int_{\mathcal{N}} \frac{p'(1-y,\boldsymbol{\nu}|\mathbf{x})}{\mathbb{P}'(Y=1-y)}\; d\boldsymbol{\nu}} \\ &= \frac{\mathbb{P}'(Y=y|\mathbf{x})\; \mathbb{P}'(Y=1-y)}{\mathbb{P}'(Y=1-y|\mathbf{x})\; \mathbb{P}'(Y=y)}.\end{aligned} \tag{12}$$

However, unlike `BFF`, we are not estimating the likelihood or odds from simulated data, but instead directly evaluate a pretrained classifier $\mathbb{P}'(Y = y|\mathbf{x})$.

B. Proofs

For simplicity in notation, we will henceforth omit the "train" and "target" subscripts in $\mathbb{P}$. The symbol $\mathbb{P}'$ will represent the training distribution, while $\mathbb{P}$ will denote the target distribution.

Proof of Lemma 1. This follows from the fact that $W_\lambda(C; y, \boldsymbol{\nu})$ only depends on the conditional randomness of $\mathbf{X}|y, \nu$, which, under GLS, is the same on both train and target data. □

Proof of Theorem 1. Notice that

$$\begin{aligned}\mathbb{P}(\lambda(\mathbf{X}) \le C^*_{\alpha,y}(\mathbf{X})|y, \boldsymbol{\nu}) &= \mathbb{P}(\lambda(\mathbf{X}) \le C^*_{\alpha,y}(\mathbf{X}), \boldsymbol{\nu} \in S_y(\mathbf{X}; \gamma)|y, \boldsymbol{\nu}) + \mathbb{P}(\lambda(\mathbf{X}) \le C^*_{\alpha,y}(\mathbf{X}), \boldsymbol{\nu} \notin S_y(\mathbf{X}; \gamma)|y, \boldsymbol{\nu}) \\ &\le \mathbb{P}(\lambda(\mathbf{X}) \le W_\lambda^{-1}(\beta; y, \boldsymbol{\nu})|y, \boldsymbol{\nu}) + \mathbb{P}(\boldsymbol{\nu} \notin S_y(\mathbf{X}; \gamma)|y, \boldsymbol{\nu}) \\ &\le \beta + \gamma = \alpha,\end{aligned}$$

which proves the first part of the result. Similarly,

$$\begin{aligned}\mathbb{P}(\lambda(\mathbf{X}) \ge \widetilde{C}^*_{\alpha,y}(\mathbf{X})|1-y, \boldsymbol{\nu}) &= \mathbb{P}(\lambda(\mathbf{X}) \ge \widetilde{C}^*_{\alpha,y}, \boldsymbol{\nu} \in S_{1-y}(\mathbf{X}; \gamma)|1-y, \boldsymbol{\nu}) \mathbb{P}(\lambda(\mathbf{X}) \ge \widetilde{C}^*_{\alpha,y}, \boldsymbol{\nu} \notin S_{1-y}(\mathbf{X}; \gamma)|1-y, \boldsymbol{\nu}) \\ &\le \mathbb{P}(\lambda(\mathbf{X}) \ge W_\lambda^{-1}(\beta; 1-y, \boldsymbol{\nu})|1-y, \boldsymbol{\nu}) + \mathbb{P}(\boldsymbol{\nu} \notin S_{1-y}(\mathbf{X}; \gamma)|1-y, \boldsymbol{\nu}) \\ &\le 1 - \beta + \gamma = 1 - \alpha,\end{aligned}$$

and therefore

$$\mathbb{P}(\lambda(\mathbf{X}) \le \widetilde{C}^*_{\alpha,y}(\mathbf{X})|1-y, \boldsymbol{\nu}) \ge \alpha,$$

which concludes the proof. □

Proof of Theorem 2. By construction

$$\begin{aligned}\mathbb{P}(Y \in \mathbf{H}(\mathbf{X}; \alpha)|y, \boldsymbol{\nu}) &= \mathbb{P}\left(\widehat{\tau}_y(\mathbf{X}) > C^*_{\alpha,y}(\mathbf{X})|y, \boldsymbol{\nu}\right) \\ &= 1 - \mathbb{P}\left(\widehat{\tau}_y(\mathbf{X}) \le C^*_{\alpha,y}(\mathbf{X})|y, \boldsymbol{\nu}\right) \\ &\ge 1 - \alpha,\end{aligned}$$

where the last inequality follows from Lemma 1. This proves the first statement of the theorem. To prove the second statement, notice that

$$\begin{aligned}\mathbb{P}(Y \in \mathbf{H}(\mathbf{X};\alpha)) &= \int \mathbb{P}(Y \in \mathbf{H}(\mathbf{X};\alpha)|y,\boldsymbol{\nu}) d\mu(y,\boldsymbol{\nu}) \\ &\geq \int (1-\alpha) d\mu(y,\boldsymbol{\nu}) \\ &= 1-\alpha,\end{aligned}$$

where $\mu(y,\boldsymbol{\nu})$ denotes the measure on $(Y,\boldsymbol{\nu})$ on the target set. □

C. Estimating the Rejection Probability Function

We learn $W_\lambda(C;y,\boldsymbol{\nu})$ using a monotone regression that enforces the rejection probability to be a non-decreasing function of C. For each point $i = 1,\ldots,B'$ in the calibration set $\mathcal{T}' = \{(Y_1,\boldsymbol{\nu}_1,\mathbf{X}_1),\ldots,(Y_{B'},\boldsymbol{\nu}_{B'},\mathbf{X}_{B'})\}$ drawn from $p_{\text{train}}(\boldsymbol{\theta})\mathcal{L}(\mathbf{x};\boldsymbol{\theta})$ where $\boldsymbol{\theta} = (Y,\boldsymbol{\nu})$, we sample a set of K cutoffs according to the empirical distribution of the test statistic λ. Then, we regress the random variable

$$Z_{i,j} := \mathbb{I}\left(\lambda(\mathbf{X}_i) \leq C_j\right) \tag{13}$$

on Y_i, $\boldsymbol{\nu}_i$ and $C_{i,j}$ $(= C_j)$ using the "augmented" calibration set $\mathcal{T}'' = \{(Y_i,\boldsymbol{\nu}_i,C_{i,j},Z_{i,j})\}_{i,j}$, for $i = 1,\ldots,B'$ and $j = 1,\ldots,K$, where K is the augmentation factor. See Algorithm 2 for details.

Algorithm 2 `Learning the Rejection Probability Function`

Input: test statistic λ; calibration data $\mathcal{T}' = \{(Y_1,\boldsymbol{\nu}_1,\mathbf{X}_1),\ldots,(Y_{B'},\boldsymbol{\nu}_{B'},\mathbf{X}_{B'})\}$; sampled cutoffs $G = \{C_1,\ldots,C_K\}$
Output: Estimate of the rejection probability $W_\lambda(C;y,\boldsymbol{\nu})$ for all $C \in G$, $y \in \{0,1\}$ and $\boldsymbol{\nu} \in \mathcal{N}$

// Learn rejection probability from augmented calibration data $\mathcal{T}''$
Set $\mathcal{T}'' \leftarrow \emptyset$
for i in $\{1,...,B'\}$ **do**
  for j in $\{1,...,K\}$ **do**
    Compute $Y_{i,j} \leftarrow \mathbb{I}\left(\lambda(\mathbf{X}_i) \leq C_j\right)$
    Let $\mathcal{T}'' \leftarrow T'' \cup \{(Y_i,\boldsymbol{\nu}_i,C_j,Z_{i,j})\}$
  end for
end for
Estimate $W_\lambda(C;y,\boldsymbol{\nu}) := \mathbb{P}_{y,\boldsymbol{\nu}}\left(\lambda(\mathbf{X}) \leq C\right)$ from $\mathcal{T}''$ via a regression of Z on Y, $\boldsymbol{\nu}$ and C, which is monotonic in C.
return Estimated rejection probabilities $\widehat{W}_\lambda(C;y,\boldsymbol{\nu})$, for $C \in G$, $y \in \{0,1\}$ and $\boldsymbol{\nu} \in \mathcal{N}$

D. Diagnostics of Estimated ROC Curves

Here we describe how to evaluate goodness-of-fit of an estimate of the rejection probability function. This is inspired by methods that use the Probability Integral Transform (PIT) to assess conditional density estimators (Cook et al., 2006; Freeman et al., 2017; Izbicki et al., 2017; D'Isanto & Polsterer, 2018).

If $W_\lambda(C;y,\boldsymbol{\nu}) = \mathbb{P}_{\text{target}}\left(\lambda(\mathbf{X}) \leq C|y,\boldsymbol{\nu}\right) = F_{\lambda(\mathbf{X})|y,\boldsymbol{\nu}}(C)$ is well estimated, then the random variable $W_\lambda(\lambda(\mathbf{X}');y,\boldsymbol{\nu}) \sim U(0,1)$, where $\mathbf{X}'$ is drawn from the simulator using $(y,\boldsymbol{\nu})$ as parameters. This suggests we assess the performance of our estimator of W, $\widehat{W}$, via a P-P plot comparing $\widehat{W}(\lambda(\mathbf{X}_1);Y_1,\boldsymbol{\nu}_1),\ldots,\widehat{W}(\lambda(\mathbf{X}_B);Y_B,\boldsymbol{\nu}_B)$ to a Uniform(0,1) distribution, where $(\lambda(\mathbf{X}_1);Y_1,\boldsymbol{\nu}_1),\ldots,(\lambda(\mathbf{X}_B);Y_B,\boldsymbol{\nu}_B)$ denote an evaluation sample drawn from the simulator. The distribution of these statistics can however be uniform even if $\widehat{W}$ is not a good estimate (Zhao et al., 2021, Theorem 1). Here, we avoid this problem by dividing the parameter space Θ into bins and constructing separate distribution plots for samples within each bin.

E. The Standard Bayes Classifier

Lemma 2 (Bayes classifier)**.** *Let $h : \mathcal{X} \to \{0,1\}$ be a classification rule. Define the weighted loss*

$$W = c_1 I_{\{1\}}(Y)\, I_{\{0\}}(h(\mathbf{X})) + c_0 I_{\{0\}}(Y)\, I_{\{1\}}(h(\mathbf{X})), \tag{14}$$

where c_k is the cost of mis-classifying a $Y = k$ observation, for $k = 0, 1$. The Bayes (that is, optimal) classifier that minimizes the error rate $\mathbb{E}_{target}(W)$ averaged over both $\mathbf{X}$ and Y is given by

$$h^*(x) = \begin{cases} 1 & \text{if } \mathbb{P}_{target}(Y = 1|\mathbf{x}) > \alpha^*, \\ 0 & \text{if } \mathbb{P}_{target}(Y = 1|\mathbf{x}) < \alpha^*, \\ \text{arbitrary} & \text{if } \mathbb{P}_{target}(Y = 1|\mathbf{x}) = \alpha^*, \end{cases} \tag{15}$$

where $\alpha^* := \frac{c_0}{c_0+c_1}$.

Remark 1 (Balanced accuracy)**.** *If there is no shift between the train and target sets, a common choice for the loss (14) is $c_1 = 1/\mathbb{P}_{train}(Y = 1)$ and $c_0 = 1/\mathbb{P}_{train}(Y = 0)$. This yields the balanced error rate*

$$\mathbb{E}_{train}(W) = \mathbb{P}_{train}(h(\mathbf{X}) = 0|Y = 1) + \mathbb{P}_{train}(h(\mathbf{X}) = 1|Y = 0)$$

and the cut-off $\alpha^* = \mathbb{P}_{train}(Y = 1)$ for the Bayes classifier (Equation 15).

Remark 2 (Bayes classifier under GLS)**.** *Under GLS, there is no monotonic relationship between $\mathbb{P}_{target}(Y = 1|\mathbf{x})$ and $\mathbb{P}_{train}(Y = 1|\mathbf{x})$. Thus, it is not possible to use $\mathbb{P}_{train}(Y = 1|\mathbf{x})$ to recover $\mathbb{P}_{target}(Y = 1|\mathbf{x})$ using standard label shift corrections (Saerens et al., 2002; Lipton et al., 2018).*

Remark 3 (Bayes classifier under the presence of nuisance parameters but no GLS)**.** *If there is no GLS, $\mathbb{P}_{train}(Y = 1|\mathbf{x}) = \mathbb{P}_{target}(Y = 1|\mathbf{x})$. However, without a nuisance-aware cutoff, the Bayes classifier is usually calibrated to control type-I error marginally over $\boldsymbol{\nu}$. NACS instead controls this error for all $\boldsymbol{\nu} \in \mathcal{N}$.*

F. Additional Results and Details on Cosmic Ray Experiment

F.1. Experimental Set-Up with Ground-Based Detector Arrays

The data used in this paper are generated via the CORSIKA cosmic ray simulator (Heck et al., 1998). CORSIKA is a Monte Carlo simulation program that models the interactions of primary cosmic rays with the Earth's atmosphere. Given values of the parameters μ, E, Z, A, which define the primary cosmic ray identity, energy, zenith and azimuth angle, respectively, CORSIKA outputs the identities, momenta, positions, and arrival times of all secondary particles generated in the atmospheric shower, that eventually reach the ground and that are mostly muons, electrons and photons at gamma-ray energies with minor abundance of heavier particles.

The measured data $\mathbf{x}$ in our analysis does not incorporate the full shower footprint, as this level of information cannot be captured in any realistic scenario. Instead, we simulate a simple 6×6 detector grid, where each detector covers a 2×2 m^2 area, with 48 m detector spacing. Information for a secondary particle of a particular shower footprint is incorporated into the analysis only if that secondary particle lands within the area of a detector. See Figure 6 (right) for a simplified representation of the detector grid.

We assume 100% detector efficiency and that all secondary particles types are detectable. We also assume that showers always originate at the center of the detector grid. Finally, we assume that both the zenith and azimuth angles Z and A are known due to the relative ease with which they can be estimate from observed footprint data. Thus, our only nuisance parameter for inference on μ is the energy E of the cosmic ray.

The data used to estimate the test statistic are drawn according to the following distribution (which may be different from that of actual astrophysical sources):

1. Gamma ray to Hadron ratio 1:1 (whereas actual observed ratios are in the range 1:1,000 – 1:100,000)
2. Energy between 100 TeV and 10 PeV, with probability density proportional to E^{-1} for gamma rays and E^{-2} for hadrons (with standard astrophysical sources closer to between -2:-4)
3. Zenith uniformly distributed between 0 and 65 degrees
4. Azimuth uniformly distributed between -180 and 180 degrees

To derive $\mathbf{x}_i$, we first define four secondary particle groups: photons (neutral); electrons and positrons; muons (charged); and all other secondary particle types. Then for each simulated detector, we record the count of particles in each group that

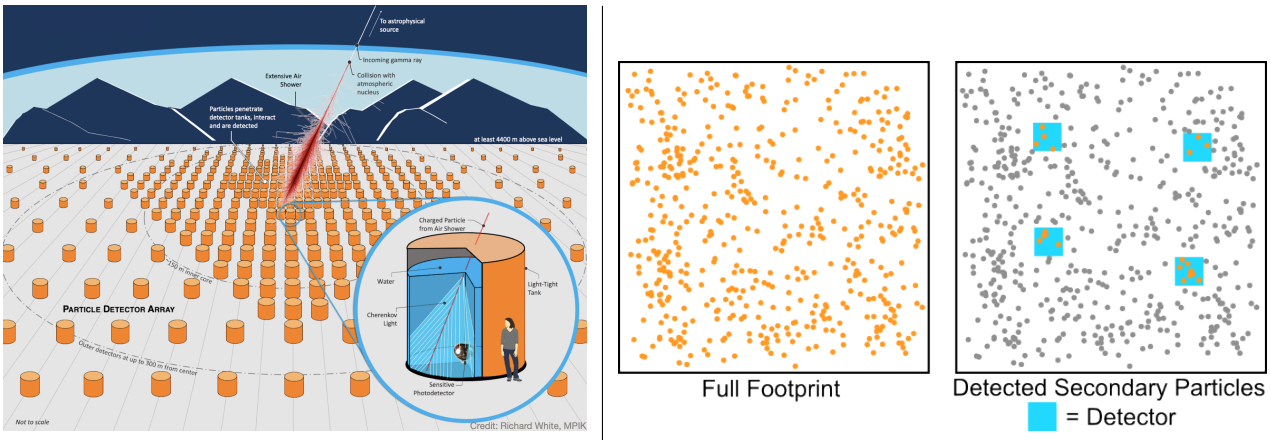

Figure 6. Left: Artistic representation of the SWGO array. The inlay shows the individual detector unit. *Right:* Although we have access to all secondary particles in our simulated cosmic ray showers, we only include the particles that hit our simulated detector setup (blue rectangles) in the analysis. This layout pictured here is an illustrative example.

hit the detector. This results in a vector of length $4 \cdot 36 = 144$ for each primary cosmic ray that represents the detector data. We construct $\mathbf{x}_i$ by concatenating the detector data with Z_i and A_i.

For the calibration and test sets, we use the same reference distribution.

F.2. Details on the algorithms used in Section 5.3

We used gradient boosting probabilistic classifiers as implemented in `CatBoost` (Prokhorenkova et al., 2018) to estimate both $\mathbb{P}(Y|\mathbf{X})$ and $W_\lambda(C; y, \boldsymbol{\nu})$. For the latter, `CatBoost` allows to easily enforce monotonicity constraints on the features, which we used on C. To compute cutoffs, we used the `brentq` routine (Brent, 2013) to calculate the inverse and the differential evolution global optimization algorithm (Storn & Price, 1997) to find the infimum. Both are implemented in `SciPy` (Virtanen et al., 2020). To obtain confidence sets for $\boldsymbol{\nu}$, we used the method developed by Masserano et al. (2023) with a masked autoregressive flow (Papamakarios et al., 2017) since it guarantees that the constructed region contains the true value of $\boldsymbol{\nu}$ at the desired confidence level for all $\boldsymbol{\nu} \in \mathcal{N}$.

F.3. Additional Results

Figures 7 and 8 mirror the results for 5, focusing on cosmic rays predicted to be hadrons and true hadron cosmic rays repectively. Identifying hadrons is of lesser scientific value than identifying gamma rays, so the results here are presented mainly for reference.

G. Additional Results and Details on the RNA Sequencing experiment

G.1. Data Simulation Procedure

The `scDesign3` simulator for RNA-Seq constructs a new simulated dataset through the following steps

1. The user chooses a model type (e.g. linear with Gaussian noise) and specification to model the relationship between cell gene counts and cell features.
2. `scDesign3` estimates model parameters on the reference data.
3. The user supplies a matrix of all features of all cells in for the new simulated data.
4. `scDesign3` outputs the gene counts for these cells by sampling from the estimated model.

In our paper, we use a negative binomial GLM with cell type and batch protocol indicator as the only features:

$$\log \mathbb{E}[X_{i,j} \mid Y_j, B_j] = \alpha_i + \beta_i Y_j + \gamma_{\mathbf{i}} \mathbf{B_j},$$

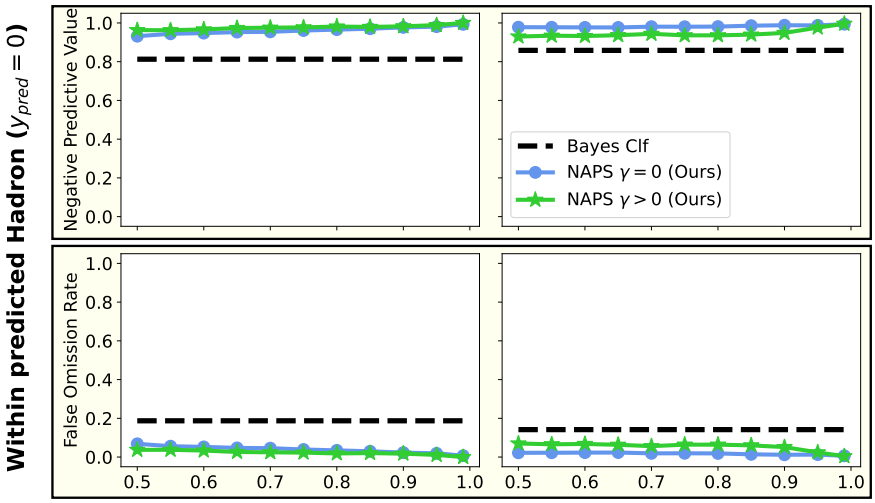

Figure 7. **Classification metrics within predicted Hadrons ($y_{\text{pred}} = 0$).** Results are binned according to whether the shower energy is below (left) or above (right) the median value. Nuisance-aware prediction sets (NAPS $\gamma = 0$; blue) achieve high precision and low false discovery rates (FDR), especially at high confidence levels. In addition, by constraining the nuisance parameters $\boldsymbol{\nu} = (E, A, Z)$, we see performance (NAPS $\gamma > 0$; green) increase in the lower energy bin but with a corresponding tradeoff in the higher energy bins. Both approaches yield better results relative to the oracle Bayes classifier (black dashed line).

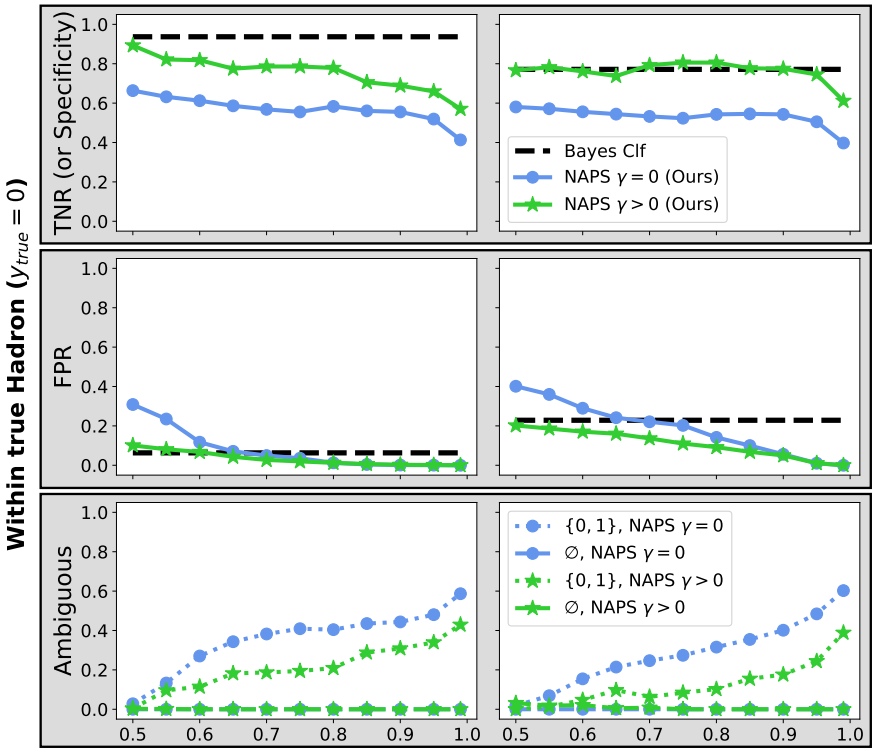

Figure 8. **Classification metrics within true Hadrons ($y = 0$).** Results are binned according to whether the shower energy is below (left) or above (right) the median value. Our set-valued classifier makes explicit its level of uncertainty on the label y by returning ambiguous prediction sets (bottom row) for hard-to-classify $\mathbf{x}_{\text{target}}$. Even so, NAPS with $\gamma > 0$ is able to achieve a comparable number true negatives in the higher energy bins and lower number of false positives in both energy bins relative to the Bayes classifier. Here $\gamma = \alpha \times 0.3$

where

1. $X_{i,j}$ are the observed counts for gene i for cell j
2. $Y_j \in \{0, 1\}$ is the cell type for cell j, `CD4`$^+$ T-cells (Y = 1) or `Cytotoxic` T-cells (Y = 0)
3. $\mathbf{B}_j$ is which of the 4 protocols was used to process cell j, with a separate model coefficient for each protocol excluding the baseline (represented by the vector $\gamma_i \in \mathbb{R}^3$)

We also restrict our analysis to 100 genes chosen randomly from the approximately 6000 genes in the reference dataset. Although each gene count receives its own set of model parameters, new gene counts are generated in a way that captures the correlation between gene counts in the reference data. See (Song et al., 2023) for more details.

The reference data used in our analysis contains two experimental protocols. One is used as a baseline to derive $\widehat{\alpha}_i$. The second is used to fit the first entry of each $\widehat{\gamma}_i$, denoted $\widehat{\gamma}_{i,1}$. The last two entries $\widehat{\gamma}_{i,2}$ and $\widehat{\gamma}_{i,3}$ are constructed in this way:

1. Each $\widehat{\gamma}_{i,2}$ is sampled with replacement from $\{\widehat{\gamma}_{i,1} : |\widehat{\gamma}_{i,1}| < \text{median}(\{|\widehat{\gamma}_{j,1}|, j \in [100]\})\}$
2. Each $\widehat{\gamma}_{i,3}$ is sampled with replacement from $\{\widehat{\gamma}_{i,1} : |\widehat{\gamma}_{i,1}| \geq \text{median}(\{|\widehat{\gamma}_{j,1}|, j \in [100]\})\}$

These last two batch protocols are meant to emulate a weak and stronger batch effect respectively than the different between the two original experimental protocols, while keeping realistic estimates for the effects on gene counts.

G.2. Details on the algorithms used in Section 5.2

We used gradient boosting probabilistic classifiers as implemented in `CatBoost` (Prokhorenkova et al., 2018) to estimate both $\mathbb{P}(Y|\mathbf{X})$ and $W_\lambda(C; y, \boldsymbol{\nu})$. For the latter, `CatBoost` allows to easily enforce monotonicity constraints on the features, which we used on C. To compute cutoffs, we used the `brentq` routine (Brent, 2013) to calculate the inverse and the differential evolution global optimization algorithm (Storn & Price, 1997) to find the infimum. Both are implemented in `SciPy` (Virtanen et al., 2020). The three baselines against which we compare NAPS were computed from the same base probabilistic classifier (also used for NAPS). After training it, we calibrated it on the same set used for NAPS via isotonic regression, but only for the baselines (our method has a separate calibration procedure as described in Section 3. Then we computed cutoffs as described in Sadinle et al. (2019); Romano et al. (2020).

G.3. Additional Results

Taking `CD4`$^+$ T-cells (Y = 1) to be the positive class, Figures 9, 10, 11, 12 show various performance metrics for four prediction set methodologies: standard prediction sets (Sadinle et al., 2019, Theorem 1), class-conditional prediction sets (Sadinle et al., 2019), conformal adaptive prediction sets (APS; Romano et al. (2020)), and NAPS with $\gamma = 0$. For many of the metrics like precision and NPV, each method achieves very good performance (perhaps due to the ease of the underlying inference problem). For TPR, we see that each method has differing strength for each of the protocols. We also notice that at very high levels of confidence, conformal APS starts outputting $\{0, 1\}$ for every observation, leading to a sharp drop in performance across all metrics.

H. Computational Analysis: Training and Inference Times

Table 1 reports training and inference times for NAPS under the Single-Cell RNA Sequencing (Section 5.2) and Atmospheric Cosmic-Ray Showers (Section 5.3) experiments. Dataset sizes are the proportions included in the training, calibration and inference sets out of the total number of simulations indicated in Sections 5.2 and Section 5.3. For calibration, we report the time needed to estimate ROC curves from the augmented calibration set, including "re-calibration" of the estimated rejection probabilities via isotonic regression. For NAPS with $\gamma > 0$ (only performed in Section 5.3), inference times are measured per-observation (on average) since cutoffs are data-dependent and need to be computed for each $\mathbf{x}$. For NAPS with $\gamma = 0$, we report the total time needed to compute cutoffs, as they can then be applied to any new observation $\mathbf{x}$ (i.e., they are amortized with respect to observations). Once this is done, constructing the prediction sets takes only a few milliseconds. All times are computed for inference at a single level α. Classifier training and the calibration procedure only need to be estimated once (here we report times that include five-fold cross-validation). All computations were performed on a MacBook Pro M1Pro with 16 GB of RAM.

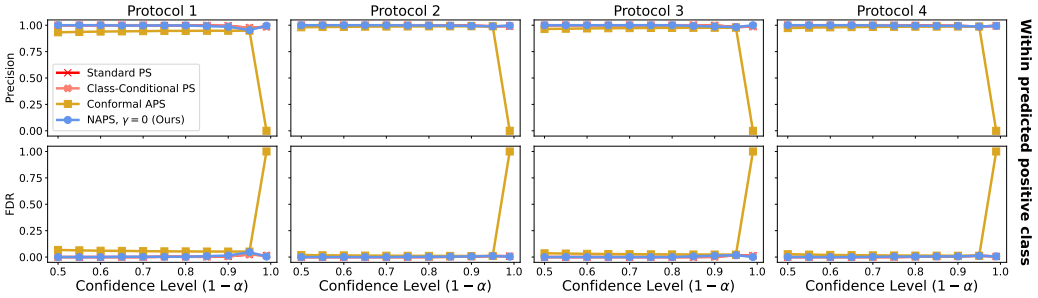

Figure 9. **Classification metrics within predicted positive class**: Precision (top) and FDR (bottom) for observations predicted to be $\texttt{CD4}^+$ T-cells (i.e. prediction set output is $\{1\}$), additionally separated by protocol (columns). Metrics are shown for nuisance-aware prediction sets (NAPS $\gamma = 0$; blue), standard prediction sets (red), class-conditional prediction sets (pink), and conformal adaptive prediction sets (APS) (gold). At high levels of confidence, conformal APS outputs $\{0, 1\}$ for all points in the test set; the corresponding metrics that require the prediction set to have one element have been set to their worst-case value.

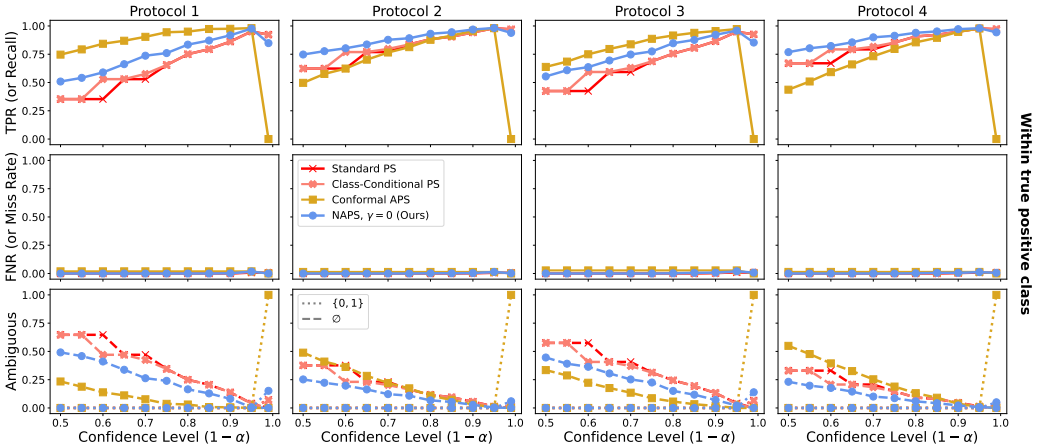

Figure 10. **Classification metrics within true positive class**: TPR (top), FNR (middle) and proportion of ambiguous sets (bottom) for true $\texttt{CD4}^+$ T-cells, additionally separated by protocol (columns). Metrics are shown for Nuisance-aware prediction sets (NAPS $\gamma = 0$; blue), standard prediction sets (red), class-conditional prediction sets (pink), and conformal adaptive prediction sets (APS) (gold). At high levels of confidence, conformal APS outputs $\{0, 1\}$ for all points in the test set; the corresponding metrics that require the prediction set to have one element have been set to their worst-case value.

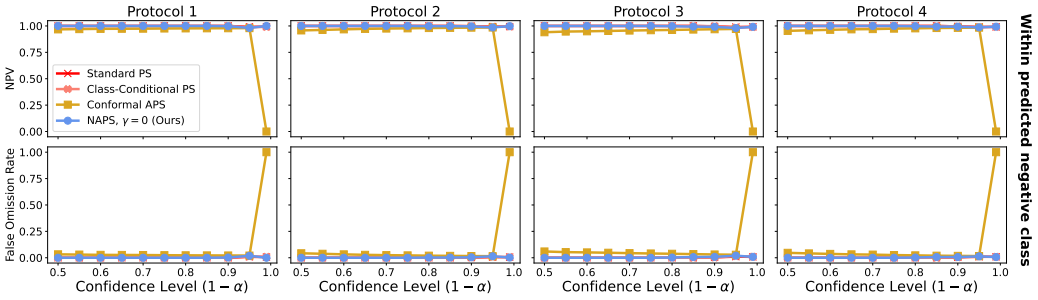

Figure 11. **Classification metrics within predicted negative class**: NPV (top) and False Omission Rate (bottom) for observations predicted to be `Cytotoxic` T-cells (i.e. prediction set output is $\{0\}$), additionally separated by protocol (columns). Metrics are shown for Nuisance-aware prediction sets (NAPS $\gamma = 0$; blue), standard prediction sets (red), class-conditional prediction sets (pink), and conformal adaptive prediction sets (APS) (gold). At high levels of confidence, conformal APS outputs $\{0, 1\}$ for all points in the test set; the corresponding metrics that require the prediction set to have one element have been set to their worst-case value.

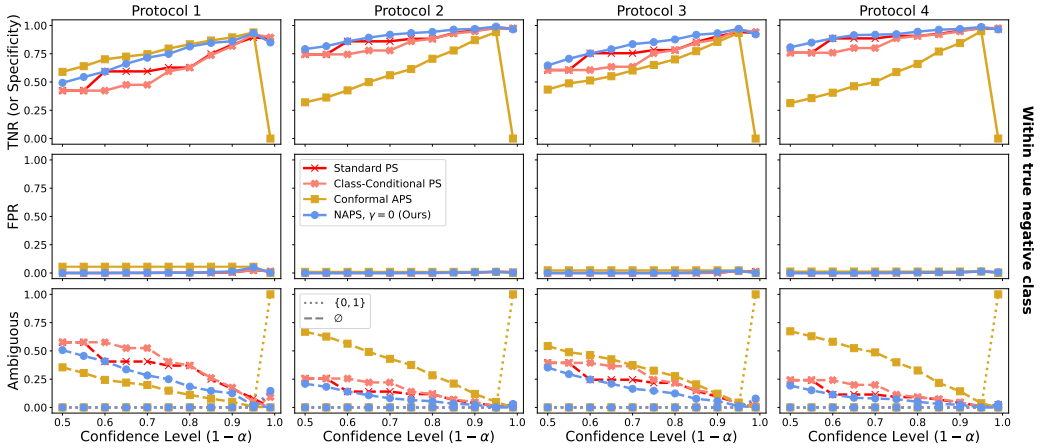

Figure 12. **Classification metrics within true negative class**: TNR (top), FPR (middle) and proportion of ambiguous sets (bottom) for true `Cytotoxic` T-cells, additionally separated by protocol (columns). Metrics are shown for Nuisance-aware prediction sets (NAPS $\gamma = 0$; blue), standard prediction sets (red), class-conditional prediction sets (pink), and conformal adaptive prediction sets (APS) (gold). At high levels of confidence, conformal APS outputs $\{0, 1\}$ for all points in the test set; the corresponding metrics that require the prediction set to have one element have been set to their worst-case value.

I. Synthetic Example: Deep Dive

I.1. Impact of the Nuisance Parameter

As mentioned in the main text, we consider a process that generates events (Y_i, X_i), where $Y_i \in \{0, 1\}$ determines the type or label of the event, and $X_i \in [0, 1]$ is the sole feature of the event. The distribution of events is defined as follows

1. $\mathbb{P}[Y_i = 0] = \mathbb{P}[Y_i = 1] = 1/2$
2. Conditional density for $Y = 1$: $p(\mathbf{x}_i \mid Y_i = 1) = \dfrac{e^{\mathbf{x}_i}}{e - 1}$
3. Conditional density for $Y = 0$: $p(\mathbf{x}_i \mid Y_i = 0, \nu_i) = \dfrac{\nu_i e^{-\nu_i \mathbf{x}_i}}{1 - e^{-\nu_i}}$

Where ν is an additional nuisance parameter that influences the density of X for $Y = 0$ events. ν_i is assumed to be drawn from some distribution independently for each $Y = 0$ event. We are interested in inferring Y given observed $\mathbf{X}$ and unobserved ν. Figure 13 shows how the presence of the nuisance parameter affects this inference task.

The top left of Figure 13 demonstrates how the shape of the density of X for $Y = 0$ events can vary dramatically depending on the value of ν. Assuming any prior of ν can yield a density of X that does not depend on ν, but it may not closely resemble the conditional densities of X given ν for all values of ν. The top right panel shows how this variation in the shape of the densities subsequently affects the behavior of the posterior probabilities of Y given X and ν. Again, we can derive a posterior that does not depend on ν, with the same caveat as before. We also observe that the posterior probabilities are always monotonic in x, therefore any classifier or prediction set that uses cutoffs on posterior probabilities can be equivalently defined using cutoffs on x directly. The bottom left figure shows how the ROC for the Bayes Classifier (i.e. directly using the posterior probabilities to classify events) can vary under fixed ν or a prior on ν. These ROC curves

Table 1. Training and inference times for NAPS for the experiments of Sections 5.2 and 5.3.

EXPERIMENT	DATASET SIZE	TRAINING	CALIBRATION	INFERENCE ($\gamma = 0$)	INFERENCE ($\gamma > 0$)
RNA-SEQ	$0.6, 0.35, 0.5$	6 MINUTES	30 MINUTES	1 SECOND	/
COSMIC RAYS	$0.45, 0.45, 0.1$	8 MINUTES	65 MINUTES	6 SECONDS	4 SECONDS PER-OBS

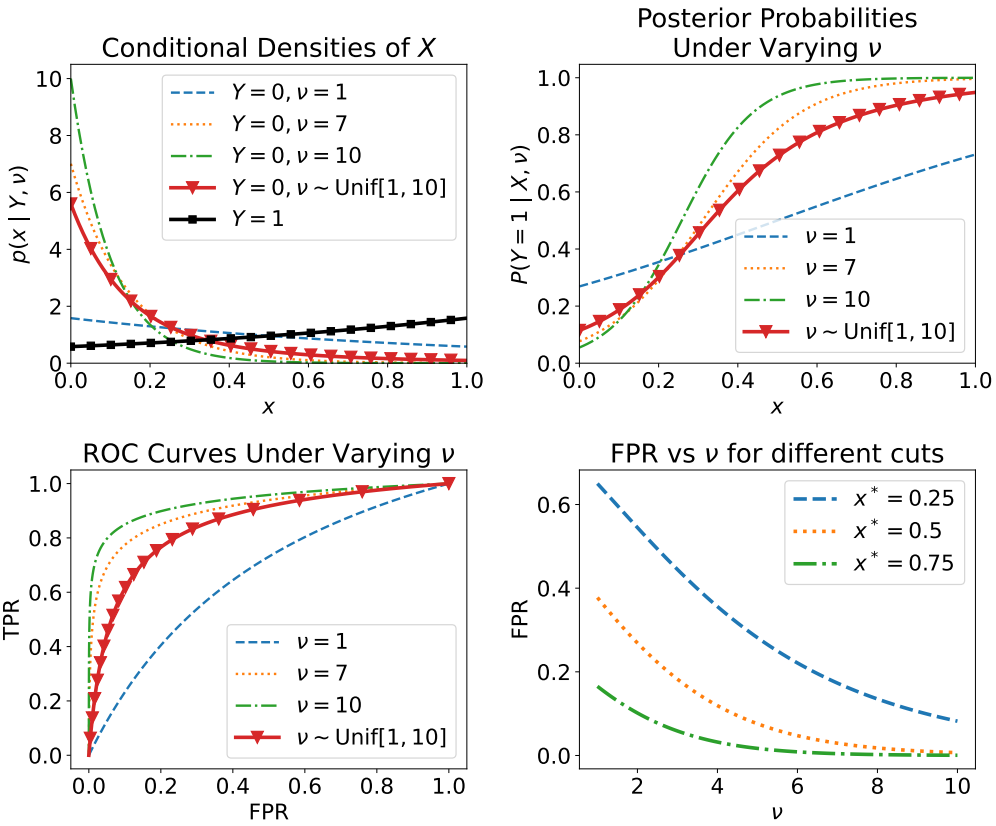

Figure 13. **Impacts of Nuisance Parameters on the Inference Task** *Top Left*: Conditional densities $p(x \mid Y, \nu)$ for various values of Y and ν according to the problem setup. The marginal density $p(x \mid Y = 0)$ shown in red is induced by a Unif$[1, 10]$ prior on ν. *Top Right*: Posterior probability $P(Y = 1 \mid X, \nu)$ as a function of X for different values of the nuisance parameter ν. The marginal posterior $P(Y = 1 \mid X)$ is shown in red for a Unif$[1, 10]$ prior on ν. *Bottom Left*: ROC curves for the Bayes Classifier holding ν fixed (blue, orange, and green curves) and for a Unif$[1, 10]$ prior on ν (red). $Y = 1$ is taken to be the positive class. *Bottom Right*: Under the classification rule that $\widehat{y}_i = 1$ if $x_i > x^*$, this figure shows how the FPR of that classifier will vary with ν. Each curve represents a different cut x^* for the classification rule.

demonstrate why ignoring nuisance parameters can yield biased or otherwise unreliable results. Every fixed value of ν as well as every prior on ν yields a completely different relationship between FPR and TPR. The bottom right figure shows that if our goal is valid FPR control for our inference task, we must take the nuisance parameter into account. Because the ultimate FPR for any cutoff depends on the value of ν for each observation, the selection of an cutoff that controls FPR must properly account for the influence of the nuisance parameter.

I.2. Additional Results

Figures 14, 15, and 16 show additional results from the synthetic examples for both standard prediction sets and class-specific prediction sets used in the cosmic ray application. All prediction sets are formed under the training prior $\nu \sim$ Unif$[1, 10]$, which is the same prior used to compute metrics under the "No GLS" setting. "With GLS" changes the target prior to $\nu \sim \mathcal{N}(4, 0.1)$ without modifying the training prior. Coverage for $Y = 1$ events, power for $Y = 0$ events (defined as $\mathbb{P}[1 \notin$ Prediction Set $\mid Y = 0]$), and precision for $\{0\}$ outputs do not vary significantly across methodologies due to the fact that $p(\mathbf{x}_i \mid Y_i = 1)$ does not depend on the distribution of ν_i. As seen in the text, our methods achieve validity regardless of the presence of GLS. We also achieve higher precision than standard or class-specific prediction sets, although we do sacrifice power compared to those methods. However, careful selection of γ in the NAPS framework can help increase power without losing validity.

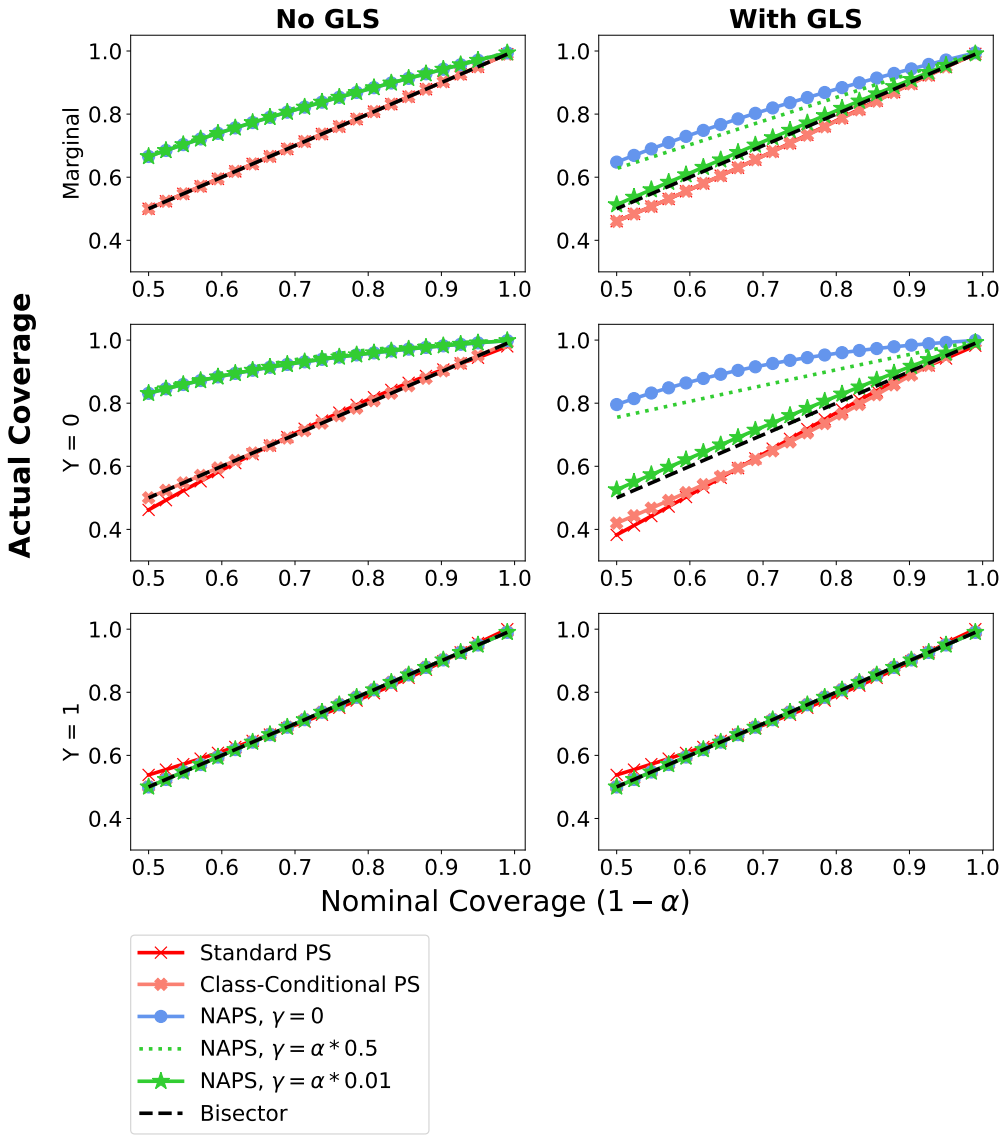

Figure 14. **Actual vs Nominal Coverage for Several Prediction Set Methods**: We compare the actual coverage of standard prediction sets (red), class-specific prediction sets (pink), and NAPS under different γ values under no GLS (left) and with GLS (right). We show marginal coverage (top), and conditional coverage for $Y = 0$ events (middle) and $Y = 1$ events (bottom)

I.3. ν-Conditional Coverage and validity under GLS

Figure 17 below explores coverage of different prediction set methods conditional on Y *and* ν, under the training prior $\nu \sim \text{Unif}[0, 1]$. We compare 4 methods:

1. Standard prediction sets that target marginal coverage only

2. Class-conditional prediction sets that target coverage conditional on Y

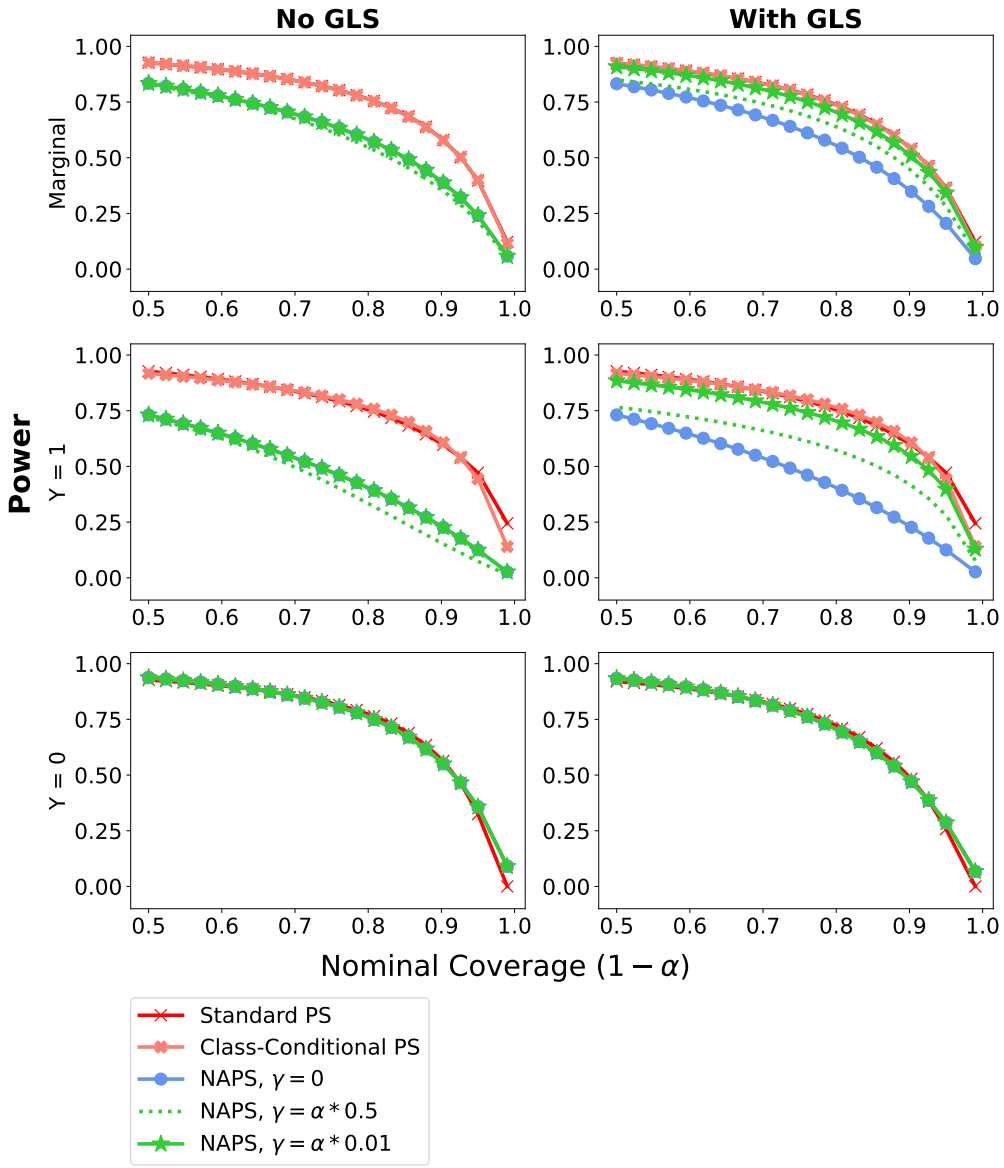

Figure 15. **Power vs Nominal Coverage for Several Prediction Set Methods**: We compare the power of standard prediction sets (red), class-specific prediction sets (pink), and NAPS under different γ values under no GLS (left) and with GLS (right). Power for $Y = 0$ events (bottom) is defined as $\mathbb{P}[1 \notin \text{Prediction Set} \mid Y = 0]$ and vice versa for $Y = 1$ (middle). Marginal power (top) is the sum of these two power metrics weighted by $\mathbb{P}[Y = 1]$.

3. Class-conditional prediction sets that additionally use the posterior mean $\widehat{\nu}(x) = \int_{\mathcal{N}} \nu \, p(\nu \mid x) d\nu$ as an point estimate of ν to evaluate the posterior. Specifically, $P[Y = 1 \mid X, \nu = \widehat{\nu}(X)]$ is used instead of $P[Y = 1 \mid X]$, where the latter integrates over the prior on ν

4. NAPS with $\gamma = 0$

Method 3 is added as a possible alternative to forming confidence sets on ν within the NAPS framework. The figure shows

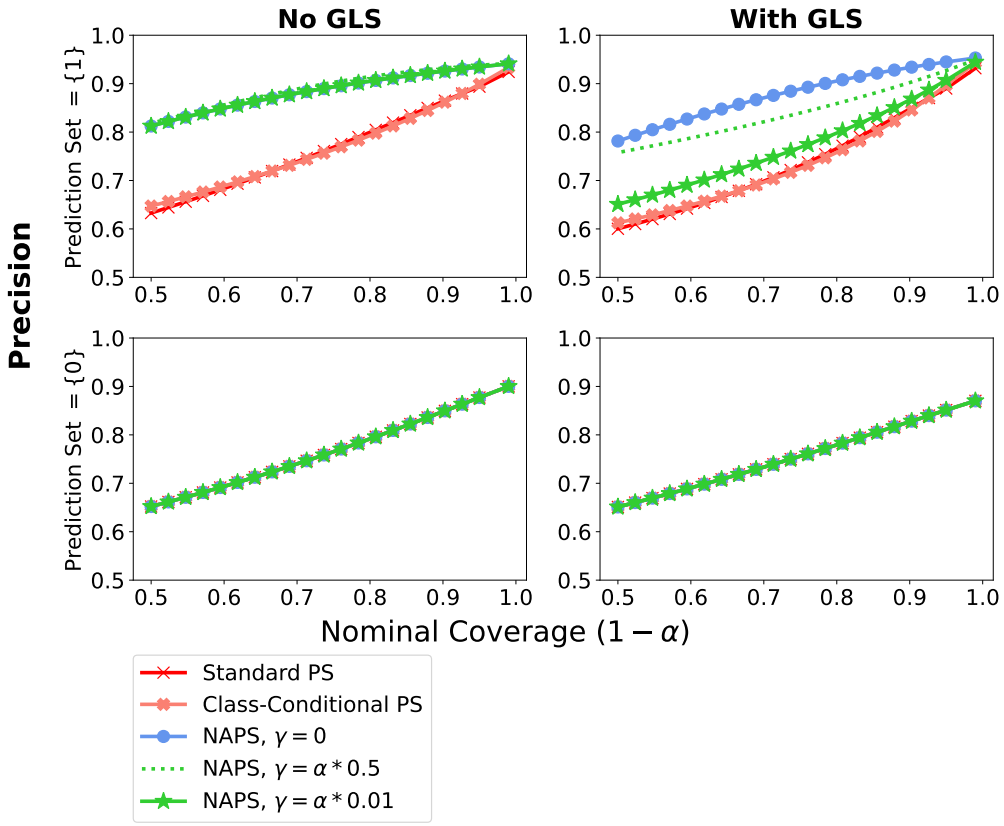

Figure 16. **Precision vs Nominal Coverage for Several Prediction Set Methods**: We compare the precision of standard prediction sets (red), class-specific prediction sets (pink), and NAPS under different γ values under no GLS (left) and with GLS (right). We define precision for prediction set $= \{0\}$ as $\mathbb{P}[Y = 0 \mid \text{prediction set} = \{0\}]$ and vice versa for prediction set $= \{1\}$ outputs. Events where prediction set $= \{0, 1\}$ or prediction set $= \emptyset$ are not considered here.

that, although standard and class-conditional prediction sets achieve marginal and class-conditional validity respectively, they do not maintain validity when conditioning on all values of ν. This is the fundamental reason that these methods do not achieve validity under GLS. Whereas, NAPS achieves validity conditional on both Y and ν, resulting in robustness to GLS. We note that method 3 achieves neither marginal nor class-conditional validity, indicating that even well-formed point estimates of ν are insufficient to reach nominal coverage levels.

I.4. When does $\gamma > 0$ for NAPS increase power?

The γ parameter for NAPS gives us the option to first form a confidence set for ν on a new observation x before optimizing the cutoffs for our test statistic (see Section 4). Because the test statistic is monotonic in the posterior probabilities, we can derive cutoffs on x directly based on the confidence set for ν. Specifically, we can simplify the procedure in Theorem 1 to the following

$$x_0(\nu; \alpha, \gamma) = x \quad \text{s.t.} \quad \mathbb{P}[X \geq x \mid Y = 0, \nu] = \alpha - \gamma$$

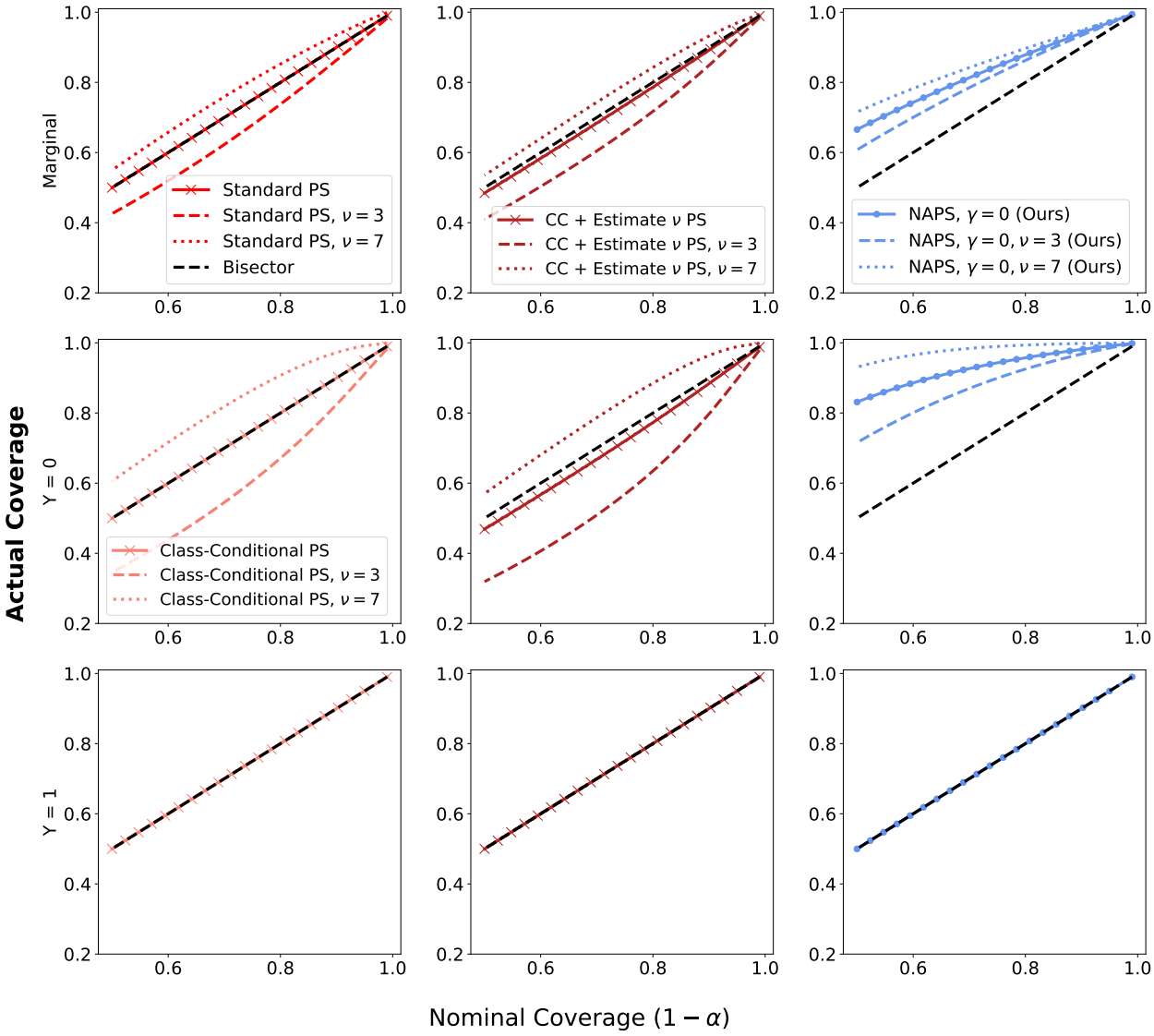

Figure 17. **Marginal, Class-conditional, and ν-conditional Coverage of Several Prediction Set Methods**: We examine marginal coverage under the training prior on ν (top), $Y = 0$ conditional coverage (middle) and $Y = 1$ conditional coverage (bottom) for standard prediction sets (red, top right only), class-conditional prediction sets (pink, middle left and bottom left), class-conditional prediction sets with estimated ν (dark red, middle column), and NAPS (blue, right column). In each figure, we also show coverage when additionally conditioning on certain values of ν (dotted and dashed lines)

$$x_0^*(\alpha) = \sup_{\nu \in S_0(x;\gamma)} x_0(\nu, \alpha)$$

$$x_1^*(\alpha) = x \quad \text{s.t.} \quad \mathbb{P}[X \leq x \mid Y = 1] = \alpha$$

Where $S_0(x;\gamma)$ is a $1-\gamma$ confidence set on ν given $Y = 0$. Then, our prediction set becomes

$$0 \in \mathbf{H}(x;\alpha) \text{ if } x < x_0^*(\alpha)$$

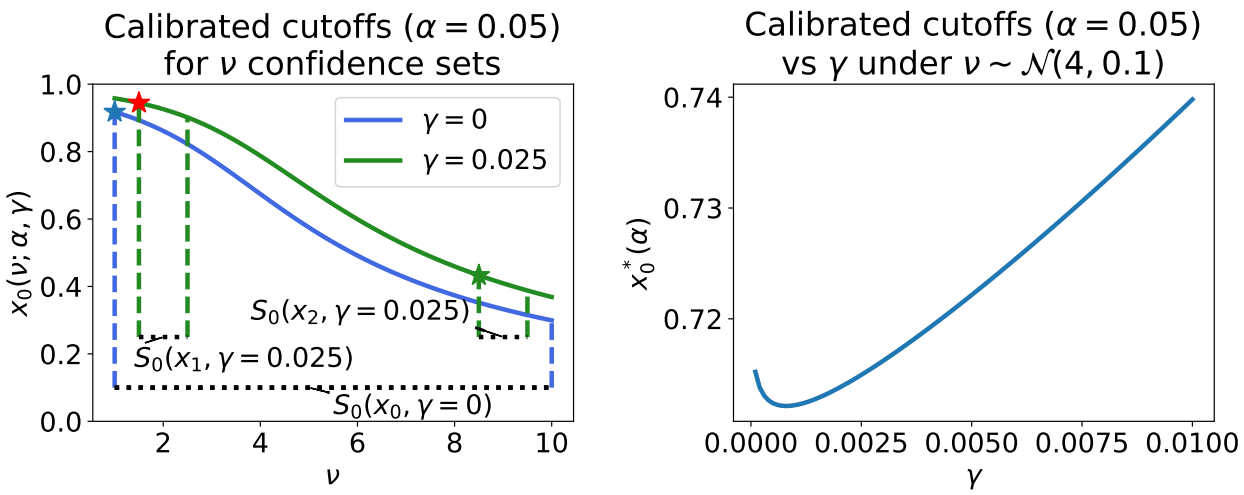

Figure 18. **Effect of γ on NAPS Power** *Left*: We show how the optimization of $x_0(\nu; \alpha, \gamma)$ depends on γ and $S_0(x; \gamma)$. The two curves show the relationship between $x_0(\nu; \alpha, \gamma)$ and ν under two values of γ. When $\gamma = 0$, we must optimize over the entire space of ν to derive $x_0^*(\alpha)$ (or equivalently, $S_0(x; \gamma = 0) = [1, 10]$ for all x. This leads to a $x_0^*(\alpha)$ value indicated by the blue star. When $\gamma = 0.0025$, we consider two hypothetical confidence sets $S_0(x_1; \gamma)$ and $S_0(x_2; \gamma)$ for ν, indicated by the two pairs of green dotted lines. In each case, we only optimize $x_0(\nu; \alpha, \gamma)$ over the values of ν in the confidence set; however, to maintain coverage at $1 - \alpha$, optimization is done over the green curve instead of the blue curve. Optimization over $S_0(x_1; \gamma)$ yields $x_0^*(\alpha)$ indicated by the red star, while optimization over $S_0(x_2; \gamma)$ yields $x_0^*(\alpha)$ indicated by the green star. *Right*: When $S_0(x; \gamma)$ is taken to be the $(\gamma/2, 1 - \gamma/2)$ quantiles of the truncated $\mathcal{N}(4, 0.1)$ distribution for all x, we can derive a relationship between $x_0^*(\alpha)$ and γ. In this case, the calibrated cutoff is minimized at $\gamma \approx 0.001$.

$$1 \in \mathbf{H}(x; \alpha) \text{ if } x > x_1^*(\alpha)$$

We note that $x_1^*(\alpha)$ does not depend on our choice of γ, so we focus on $x_0^*(\alpha)$. We also note that lower values of $x_0^*(\alpha)$ result in higher power of the final NAPS. Figure 18 below shows how the choice of γ can affect the power of the resulting NAPS.

The left panel demonstrates the tradeoff inherent in selection a value of γ. Fixing ν and α, $x_0(\nu; \alpha, \gamma)$ is increasing in γ (illustrated by the green curve being always higher than the blue curve), so the cutoff at every ν will always be higher (and power subsequently lower). However, constraining ν to $S_0(x; \gamma)$ may avoid optimizing over regions of ν where $x_0(\nu; \alpha, \gamma)$ is relatively high (i.e. small values of ν). In the synthetic example, the most power is gained when S_0 constrains ν to a region where ν is much larger than 1 (the value of ν that yields $x_0^*(\alpha)$ when $\gamma = 0$). This is illustrated by the fact that $S_0(x_2; \gamma = 0.0025)$ yields a $x_0^*(\alpha)$ value (green star) much lower than the value obtained when $\gamma = 0$ (blue star). However, setting $\gamma > 0$ can sometimes result in power *loss* if S_0 contains small values of ν. This is illustrated by the fact that $S_0(x_1; \gamma = 0.0025)$ yields an even higher $x_0^*(\alpha)$ value (red star) than the case when $\gamma = 0$. The right panel shows that, in our simple synthetic example, there is a relatively clear optimal value for γ which is non-zero.

In general, the distribution of the nuisance parameter(s) and the efficiency of the confidence sets on those NPs will determine which value of γ is optimal. If most data points have nuisance parameter values in "favorable" regions of the NP space, then it may be worth setting $\gamma > 0$ to form confidence sets. In other cases, letting $\gamma = 0$ may be the optimal choice.

I.5. Performance of NAPS under SLS

In the synthetic example, we assumed that the distribution of labels $\mathbb{P}[Y = 1]$ was the same for the training and target data. However, the distribution of ν is not the same, which leads to $p_{\text{train}}(x \mid Y) \neq p_{\text{target}}(x \mid Y)$, since

$$p(x \mid Y = y) = \int p(x \mid Y = y, \nu) \pi(\nu \mid Y = y) \, d\nu$$

and we explicitly allow for a change in $\pi(\nu \mid Y = y)$ under GLS. This setup is essentially the reverse of the Standard Label Shift (SLS) setup. Under SLS, we would assume that $\mathbb{P}_{\text{train}}[Y = 1] \neq \mathbb{P}_{\text{train}}[Y = 1]$, but that $p_{\text{train}}(x \mid Y) = p_{\text{target}}(x \mid Y)$, which is most directly achieved when the distribution of ν does not change between the training and target data.

We have shown that class-conditional prediction sets (designed to maintain coverage under SLS) do not maintain coverage under GLS due to the violation of the assumption that $p_{\text{train}}(x \mid Y) = p_{\text{target}}(x \mid Y)$. In this section, we explore how NAPS performs in the SLS setting relative to class-conditional prediction sets. We expect NAPS coverage guarantees to hold, with a decrease in power due to NAPS enforcing nominal coverage at every point in the nuisance parameter space. Figure 19 shows the results of our experiments under SLS.

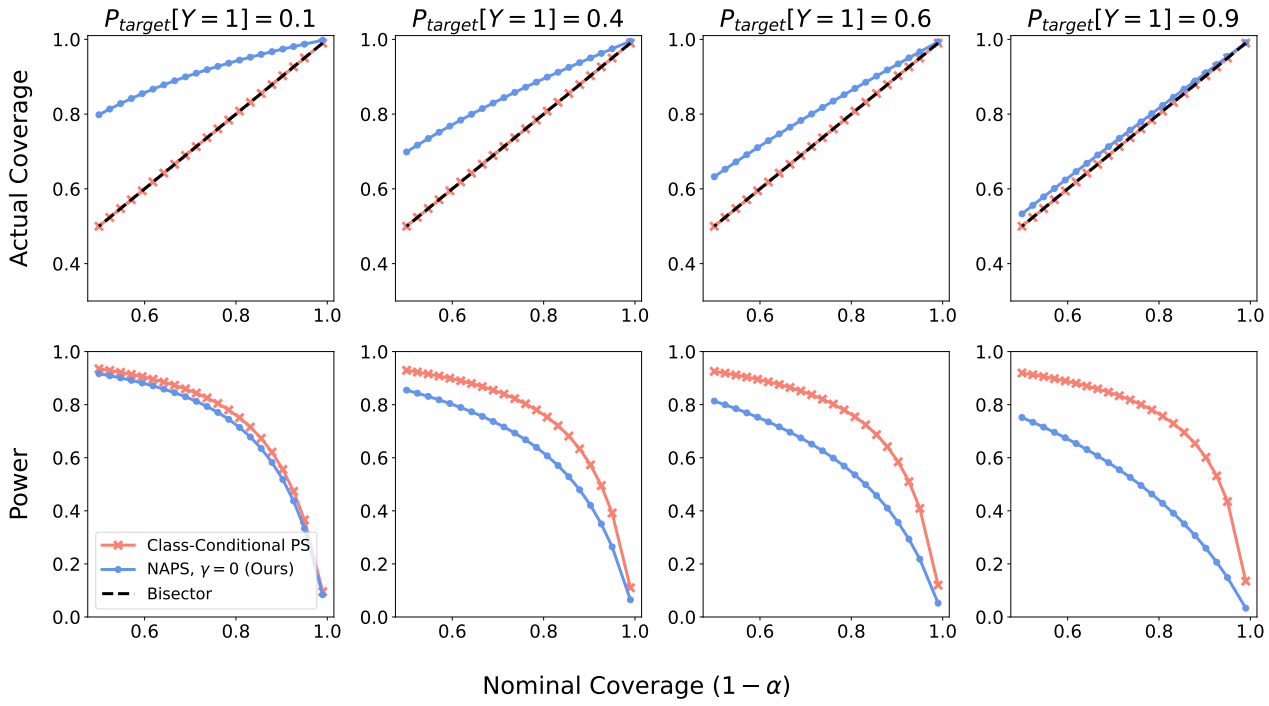

Figure 19. **Comparison of NAPS and Class-Conditional Prediction Sets under Standard Label Shift**: We plot the test set marginal coverage (top row) and marginal power (bottom row, defined as $\mathbb{P}_{\text{target}}[1 - Y \notin \text{prediction set}]$). We compare NAPS (blue) to Class-Conditional PS (pink). This comparison is done for several levels of SLS (columns), where we shift the distribution Y in the evaluation set from $\mathbb{P}_{\text{train}}[Y = 1] = 0.5$. The distribution of the nuisance parameter ν is *the same* for training versus target data; that is, we have an SLS setting.

In all SLS scenarios we tested, NAPS over-covers and achieves lower levels of power compared to class-conditional prediction sets, demonstrating the theoretical tradeoff described above. Looking at coverage, we see that as $\mathbb{P}_{\text{target}}[Y = 1]$ increases, the level of overcoverage for NAPS decreases. This is expected, since the nuisance parameter ν only affects the distribution of features for $Y = 0$ events and causes NAPS to exclude 0 from the prediction set less often. Unsurprisingly, class-conditional prediction sets exactly achieve nominal coverage under every SLS scenario.

Looking at power, we note that class-conditional prediction sets achieve similar (but not identical) power across all SLS scenarios. Power for NAPS appears to decrease as $\mathbb{P}_{\text{target}}[Y = 1]$ increases. This is a consequence of the same fact that ν only affects $Y = 0$ events; because NAPS will exclude 0 from its prediction sets less often, it will suffer a performance loss when there are relatively more $Y = 1$ events in the data. In this particular case, NAPS appears to perform best relative to class-conditional prediction sets when $\mathbb{P}_{\text{target}}[Y = 1]$ is low, but results may vary in other settings where the relationship between the nuisance parameter(s) and labels may be more complex. However, we do not expect NAPS to outperform class-conditional prediction sets (or any method developed for SLS) under SLS-only scenarios.