

Uncertainty-aware Finite-sample Quantification

Clemens Damke^{1,2} (✉) and Eyke Hüllermeier^{1,2,3}

¹ LMU Munich, Germany {clemens.damke, eyke}@ifi.lmu.de

² Munich Center for Machine Learning (MCML)

³ German Centre for Artificial Intelligence (DFKI, DSA)

Abstract A *quantification learning* (QL) method estimates the label distribution of unlabeled data. Existing methods typically target the population label distribution, even though many applications and common evaluation protocols ask for the label distribution realized in a particular finite sample. This distinction matters for small target samples, where the finite-sample label distribution can differ substantially from the population label distribution. We formalize this estimand as finite-sample quantification and propose FS-UP-ACC, a Bayesian hard-classifier-based quantifier that assigns posterior mass directly to feasible finite target counts while propagating calibration uncertainty. The method retains the correction logic of *Adjusted Classify & Count* (ACC), admits a closed-form binary update, and avoids the continuous posterior over population label distributions defined by existing Bayesian black-box-estimator-based quantifiers. Synthetic and real-data experiments show that FS-UP-ACC improves point estimates and uncertainty intervals, especially for small target samples.

Keywords: Quantification learning · Bayesian estimation

1 Introduction

Quantification learning (QL) is the task of estimating the label distribution in an unlabeled sample $\mathcal{X}_U$, given a set of labeled training data $\mathcal{D}_L$. The term “quantification” was coined by Forman [6, 7, 8], who introduced it as a *machine learning* problem distinct from standard *classification* or *regression*. Estimating the label distribution in an unlabeled sample of feature vectors can be interpreted as a dataset-level prediction problem, as opposed to the instance-level predictions in classification. Practical applications of QL include, for instance, estimating disease prevalence in a population, sentiment distributions in collections of reviews, or topic distributions in collections of documents. Esuli et al. [4] provide a comprehensive overview of the field.

A naive approach to QL is to reduce it to classification by applying a classifier to the unlabeled data and counting the predicted labels, an approach known as *Classify & Count* (CC). Given a perfect classifier, this approach is plausible, because it would yield the true label distribution. However, in practice, classifiers are rarely perfect and their errors can lead to biased estimates of the label distribution. The bias of CC is ultimately induced by the fundamentally distinct

objectives of classification and quantification — while (binary) classification aims to minimize the total number of misclassified instances, i.e., $(\text{FP} + \text{FN})$, quantification aims to minimize the distance between the true and estimated label distributions, i.e., $|\text{FP} - \text{FN}|$. As a consequence, even poor classifiers can yield good or even perfect quantification performance if false positives and false negatives are balanced, while a good (but not perfect) classifier may fail to provide accurate quantification estimates.

To make the QL problem tractable, some assumptions about the relationship between the source measure P and the target measure Q are necessary. Without such assumptions, the QL problem is ill-posed, as the training data $\mathcal{D}_L$ would be uninformative about the label distribution of the target data $\mathcal{X}_U$. A common assumption in QL is the so-called *prior probability shift* (PPS) assumption, which states that the class-conditional distributions $P(X | Y)$ and $Q(X | Y)$ are identical, while the class priors $P(Y)$ and $Q(Y)$ may differ.

A second common assumption in the QL literature is that the target quantity of interest is the population label distribution $Q(Y)$, as opposed to the label distribution realized in the finite target sample. This is a natural estimand whenever the unlabeled sample is viewed as a finite observation of a larger target population: for example, when estimating disease prevalence in a region, the long-run sentiment distribution among reviews submitted to a platform, or the distribution of topics in articles published by a news outlet. This population-level question has been studied extensively in quantification learning and label-shift estimation. Under the PPS assumption, many Fisher-consistent estimators of the target-population label distribution $Q(Y)$ given $P(X, Y)$ and $Q(X)$ have been proposed [14, 19, 22], including the *Adjusted Classify & Count* (ACC) estimator [2, 6, 8, 14], *Probabilistic Adjusted Classify & Count* (PACC) [1], the *Expectation-Maximization* method [18] and, more recently, *Bayesian Classify & Count* (BCC) [24]. These methods differ in how they estimate $Q(Y)$ and in how they represent uncertainty, but their inferential target remains the population label distribution.

Apart from estimating $Q(Y)$, there is a less well-studied but equally natural second quantification problem. In many applications, the unlabeled sample itself is the object of interest: for example, when estimating the sentiment distribution in a given set of product reviews, the topic distribution in a specific document collection, or the prevalence of a condition in a particular tested cohort. Then the relevant quantity is not the population label distribution $Q(Y)$, but the label distribution realized in that particular finite target sample. Writing $Y_1^U, \dots, Y_m^U$ for the latent labels of the unlabeled sample, this finite-sample target is the empirical label distribution

$$S_y = \frac{1}{m} \sum_{j=1}^m \mathbf{1}[Y_j^U = y] \quad \forall y \in \mathcal{Y}.$$

Even under i.i.d. sampling from Q , S_y need not equal the population probability $q_Y(y) = Q(Y = y)$ for finite samples due to sampling variation and quantization effects.

The finite-sample problem has received comparatively little direct attention, despite being implicit in common experimental protocols. For instance, the widely used *artificial prevalence protocol* (APP) evaluates quantifiers by repeatedly drawing finite target samples and comparing estimates against the realized label distribution in each sample, not against the population prevalence from which the samples were drawn. This creates a mismatch between the estimand targeted by much of the theory and the quantity used in empirical evaluation, especially when target samples are small. In this paper, we focus on this second estimand and refer to the corresponding task as *finite-sample quantification*.

2 Quantification Learning

2.1 Problem Setting

Let $\mathcal{X}$ denote the feature space and let $\mathcal{Y} = \{1, \dots, C\}$ denote a finite label space. We consider two probability measures P and Q on $\mathcal{X} \times \mathcal{Y}$. The measure P describes the labeled source population, while Q describes the target population. Let X and Y denote the canonical random variables that project from $\mathcal{X} \times \mathcal{Y}$ to the feature and label spaces, respectively. We write $P(X)$, $P(Y)$, $Q(X)$, and $Q(Y)$ for the corresponding pushforward measures and denote their densities or *probability mass functions* (PMFs) by p_X , p_Y , q_X , and q_Y , respectively.

The observed labeled data are a realized sample $\mathcal{D}_L$, modeled as an outcome of the random vector

$$\mathbf{D}_L = ((X_1^L, Y_1^L), \dots, (X_n^L, Y_n^L)), \quad (X_i^L, Y_i^L) \stackrel{\text{iid}}{\sim} P.$$

Likewise, the target data are modeled by the complete random vector

$$\mathbf{D}_U = ((X_1^U, Y_1^U), \dots, (X_m^U, Y_m^U)), \quad (X_j^U, Y_j^U) \stackrel{\text{iid}}{\sim} Q.$$

Only the feature vector is observed,

$$\mathbf{X}_U = (X_1^U, \dots, X_m^U), \quad \mathbf{Y}_U = (Y_1^U, \dots, Y_m^U),$$

where $\mathbf{Y}_U$ is latent. The realized value of $\mathbf{X}_U$ is denoted by $\mathcal{X}_U = (x_1^U, \dots, x_m^U)$.

Population quantification targets the target-population label distribution $Q(Y)$, or equivalently its PMF $q_Y \in \Delta_C$, where Δ_C is the probability simplex over C classes. **Finite-sample quantification** instead targets the label distribution realized in the target sample. We denote this random vector by $\mathbf{S} = (S_y)_{y \in \mathcal{Y}}$, where

$$S_y = \frac{1}{m} \sum_{j=1}^m \mathbf{1}[Y_j^U = y] \quad \forall y \in \mathcal{Y}. \quad (1)$$

Thus, $\mathbf{S}$ takes values in the finite simplex lattice

$$\Delta_{m,C} = \left\{ s \in [0, 1]^C : m \cdot s \in (\mathbb{N}_0)^C, \sum_{y \in \mathcal{Y}} s_y = 1 \right\}.$$

Under i.i.d. sampling from Q , it holds that $\mathbb{E}[S_y] = q_Y(y)$, but it is generally not true that $S_y = q_Y(y)$ for finite m . The two estimands therefore differ even when the sampling model is correctly specified.

In the following, we use uppercase symbols for random quantities and lowercase symbols for their realizations. Thus, $\mathbf{S}$ denotes the random finite-sample label distribution induced by $\mathbf{D}_U$, while $\mathbf{s}$ denotes its realized value for a particular target sample. Likewise, $\hat{\mathbf{S}}$ denotes the random output of a **finite-sample quantifier** when applied to random samples $\mathbf{D}_L$ and $\mathbf{X}_U$, whereas $\hat{\mathbf{s}}$ denotes the estimate obtained for observed samples $\mathcal{D}_L$ and $\mathcal{X}_U$. A **population quantifier** instead returns an estimate $\hat{q}_Y$ of q_Y . **Aggregative quantifiers** additionally have access to a hypothesis $h : \mathcal{X} \rightarrow \mathcal{Y}$ trained on a separate labeled sample.

2.2 Population Label Distribution Estimation

Most quantification methods are formulated as estimators of the population label distribution $Q(Y)$. We recall the two population-level constructions that are most relevant for the method developed in Section 3. Let $h : \mathcal{X} \rightarrow \mathcal{Y}$ be a fixed classifier and write $\hat{Y} = h(X)$ for the induced predicted-label random variable. For the observed target feature vectors $\mathcal{X}_U$, the realized predicted-label counts are

$$m_y = \sum_{j=1}^m \mathbb{1}[h(x_j^U) = y] \quad \forall y \in \mathcal{Y}. \quad (2)$$

The corresponding (potentially biased) *Classify & Count* (CC) estimate of the target-population probability of class y is $\hat{q}_{Y,CC}(y) = m_y/m$.

Adjusted Classify & Count (ACC). ACC corrects CC by accounting for the classifier’s class-conditional error rates [6, 8]. It relies on the *prior probability shift* (PPS) assumption (also commonly referred to as *label shift*), which states that the source and target class-conditional feature distributions agree,

$$P(X \mid Y = y) = Q(X \mid Y = y) \quad \forall y \in \mathcal{Y}, \quad (3)$$

while the label marginals $P(Y)$ and $Q(Y)$ may differ [14, 19, 22]. For a fixed binary classifier h , Eq. (3) implies equality of the class-conditional probabilities

$$\begin{aligned} a &= P(\hat{Y} = 1 \mid Y = 1) = Q(\hat{Y} = 1 \mid Y = 1), \\ b &= P(\hat{Y} = 1 \mid Y = 0) = Q(\hat{Y} = 1 \mid Y = 0). \end{aligned} \quad (4)$$

Here, a is the true-positive rate and b is the false-positive rate. Consequently, the population predicted-positive probability under Q satisfies

$$Q(\hat{Y} = 1) = a \cdot q_Y(1) + b \cdot q_Y(0) = b + q_Y(1)(a - b).$$

Replacing $Q(\hat{Y} = 1)$ by the observed fraction m_1/m and a, b by calibration estimates $\hat{a}, \hat{b}$ yields the binary ACC estimator

$$\hat{q}_{Y, \text{ACC}}(1) = \frac{m_1/m - \hat{b}}{\hat{a} - \hat{b}}. \quad (5)$$

Thus, binary ACC is a plug-in population-prevalence estimator: its target is $q_Y(1)$, not the realized finite-sample prevalence S_1 . Multiclass variants of ACC were proposed by Vucetic and Obradovic [23], Firat [5] and Bunse [2].

Bayesian Classify & Count (BCC). Ziegler and Czyż [24] propose a Bayesian formulation of quantification with black-box estimators. Akin to ACC, BCC avoids modeling the generally high-dimensional distribution of X directly and instead uses the finite classifier output $\hat{Y} = h(X)$. Under PPS, the class-conditional behavior of the classifier can be represented by the shared conditional PMFs

$$p_{\hat{Y}|Y=y}(\hat{y}) = P(\hat{Y} = \hat{y} \mid Y = y) = Q(\hat{Y} = \hat{y} \mid Y = y) \quad \forall y, \hat{y} \in \mathcal{Y}. \quad (6)$$

This model can be viewed as replacing the intractable posterior over target label distributions based on raw features by an approximate posterior based on classifier outputs. For a candidate target label distribution q_Y , the induced target distribution of predicted labels is

$$q_{\hat{Y}}(\hat{y}) = \sum_{y \in \mathcal{Y}} q_Y(y) p_{\hat{Y}|Y=y}(\hat{y}) \quad \forall \hat{y} \in \mathcal{Y}. \quad (7)$$

The relevant data can be summarized by finite counts. For $\mathcal{D}_L$, define the source label counts and joint label-prediction (i.e., confusion matrix) counts

$$n_y = \sum_{(x, y') \in \mathcal{D}_L} \mathbb{1}[y' = y], \quad n_{y\hat{y}} = \sum_{(x, y') \in \mathcal{D}_L} \mathbb{1}[y' = y \wedge h(x) = \hat{y}]. \quad (8)$$

For the unlabeled target sample, the predicted-label counts $m_{\hat{y}}$ are the counts from Eq. (2), with $\hat{y}$ used to emphasize predicted labels. Here, $\text{Mult}(\cdot; N, \rho)$ denotes the multinomial PMF evaluated at the observed count vector. The sufficient-statistic likelihood contributions can then be written as

$$\ell_Y(p_Y) = \text{Mult}((n_y)_{y \in \mathcal{Y}}; n, p_Y), \quad (9)$$

$$\ell_h(p_{\hat{Y}|Y}) = \prod_{y \in \mathcal{Y}} \text{Mult}((n_{y\hat{y}})_{\hat{y} \in \mathcal{Y}}; n_y, p_{\hat{Y}|Y=y}), \quad (10)$$

$$\ell_U(q_Y, p_{\hat{Y}|Y}) = \text{Mult}((m_{\hat{y}})_{\hat{y} \in \mathcal{Y}}; m, q_{\hat{Y}}). \quad (11)$$

Let $\pi_p(p_Y)$, $\pi_q(q_Y)$, and $\pi_h(p_{\hat{Y}|Y})$ denote the prior densities for the source label distribution, target label distribution, and calibration probabilities, respectively. Then the BCC posterior is

$$\begin{aligned} & \pi_{\text{BCC}}(p_Y, q_Y, p_{\hat{Y}|Y} \mid \mathcal{D}_L, \mathcal{X}_U) \\ & \propto \pi_p(p_Y) \pi_q(q_Y) \pi_h(p_{\hat{Y}|Y}) \ell_Y(p_Y) \ell_h(p_{\hat{Y}|Y}) \ell_U(q_Y, p_{\hat{Y}|Y}). \end{aligned} \quad (12)$$

The prior densities π_p , π_q , and π_h are free parameters in BCC. Since the factor $\pi_p(p_Y)\ell_Y(p_Y)$ is separate from the target-distribution and calibration terms, p_Y is a nuisance quantity for posterior inference on q_Y under this factorization and can be omitted. The reference implementation in QuaPy [16] does this and uses uniform priors over q_Y and over the rows of $p_{\hat{Y}|Y}$. Inference in Eq. (12) yields a posterior over the target-population label distribution q_Y . Thus, BCC represents uncertainty about the target-population label distribution; note that the realized finite-sample distribution $\mathbf{S}$ from Eq. (1) is *not* the direct posterior target. In practice, posterior samples can be obtained using *Hamiltonian Monte Carlo* (HMC) algorithms such as the *No-U-Turn sampler* (NUTS) [10].

2.3 Prior Work on Finite-sample Quantification

Although most work in QL and label-shift estimation targets $q_Y = Q(Y)$, the distinction between population and finite-sample targets is visible in parts of the literature. Hopkins and King [11] emphasize aggregate proportions in collections of documents rather than individual document labels, although their estimand is naturally interpreted as a population or collection-level quantity. Tasche [20] explicitly distinguishes confidence intervals for population class prevalences from prediction intervals for finite-sample relative frequencies under PPS. This distinction is close to the one made here: population quantification concerns $Q(Y)$, whereas finite-sample quantification concerns the random vector $\mathbf{S}$ from Eq. (1).

The closest finite-sample approaches are uncertainty-aware prevalence estimators that define the target as the class proportion in a specific group or target sample. Keith and O’Connor [13] model document-level prevalence for a fixed group and construct uncertainty intervals for the group prevalence. Igarane et al. [12] similarly distinguish a prior prevalence parameter from the target-sample prevalence and perform Bayesian inference for prevalence with uncertainty. These works show that finite-sample prevalence is a meaningful inferential target in its own right. However, they do not develop the particular finite-sample formulation pursued in this paper, namely an estimator that assigns probability mass directly to the realizable values of $\mathbf{S}$ while using the PPS-based correction logic of ACC.

3 A Bayesian Finite-sample Quantification Method

We now develop a finite-sample analog of the Bayesian population model from Section 2.2. The guiding idea is to keep the ACC correction logic, but to change the posterior target from the population label distribution q_Y to the realized finite-sample label distribution $\mathbf{S}$ from Eq. (1). The construction proceeds in three steps, successively modifying Bayesian CC. First, finite-sample BCC replaces the population target q_Y in Fig. 1(a) by the finite-sample target $\mathbf{S}$ in Fig. 1(b), represented through the count vector $\mathbf{J}$ (Section 3.1). Second, *finite-sample uncertainty-propagating ACC* (FS-UP-ACC) keeps this finite-sample target but makes the calibration posterior analytically tractable by cutting feedback

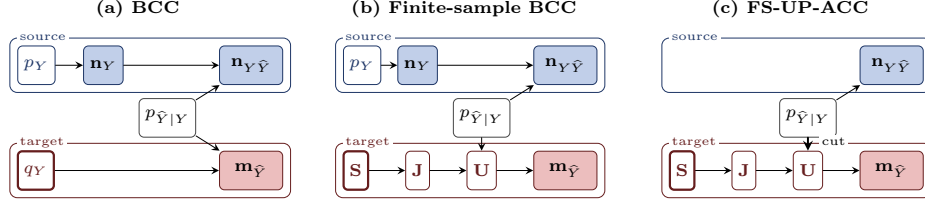


Figure 1. Dependency graphs for the Bayesian quantification models discussed in Section 3. Filled nodes denote observed count vectors: source label counts $\mathbf{n}_Y$, source joint predicted-label counts $\mathbf{n}_{Y\hat{Y}}$, and target predicted-label counts $\mathbf{m}_{\hat{Y}}$. The node $p_{\hat{Y}|Y}$ abbreviates the class-conditional predicted-label probabilities. In panels (b) and (c), $\mathbf{S}$ is represented by the finite count vector $\mathbf{J}$, which carries a prior directly. In the FS-UP-ACC model, only variables used by this cut posterior are shown. The arrow marked “cut” indicates that target counts do not update the calibration posterior for $p_{\hat{Y}|Y}$.

from the unlabeled target predictions to the calibration posterior, as shown in Fig. 1(c) (Section 3.2). Third, we describe the computationally tractable binary variant of FS-UP-ACC (Section 3.3). Figure 1 summarizes these dependency structures.

3.1 A Direct Finite-sample Bayesian Target

A direct approach to a finite-sample BCC model is to exchange the target variable. As shown in Fig. 1(a), the target label distribution q_Y in BCC induces the multinomial likelihood for the observed predicted-label counts. In finite-sample BCC (see Fig. 1(b)), this population target is replaced by the finite-sample label-distribution vector $\mathbf{S}$. Since $\mathbf{S}$ is fully determined by target-sample label counts, we work with the count vector

$$J_y = \sum_{j=1}^m \mathbb{1}[Y_j^U = y] \quad \forall y \in \mathcal{Y}, \quad (13)$$

and write $\mathbf{J} = (J_y)_{y \in \mathcal{Y}}$ for the random vector of target-sample label counts. Then $\mathbf{S} = \mathbf{J}/m$, with $\mathbf{J}$ taking values in the finite count simplex

$$\mathcal{J}_{m,C} = \left\{ \mathbf{j} \in \mathbb{N}_0^C : \sum_{y \in \mathcal{Y}} j_y = m \right\}. \quad (14)$$

A direct finite-sample Bayesian formulation therefore assigns posterior mass to the grid points $\mathbf{j}/m$, rather than a posterior density over population simplex points. The corresponding prior $\pi_{\mathbf{J}}$ is a PMF on the finite count simplex $\mathcal{J}_{m,C}$.

The source side remains unchanged. It consists of the source label likelihood from Eq. (9), the calibration likelihood from Eq. (10), and the class-conditional prediction probabilities from Eq. (6). Finite-sample BCC replaces the population

target likelihood $\ell_U(q_Y, p_{\hat{Y}|Y})$ from Eq. (11) by a likelihood conditional on the finite count vector $\mathbf{J}$. Let $\mathbf{M} = (M_{\hat{y}})_{\hat{y} \in \mathcal{Y}}$ denote the random target predicted-label count vector, with realization $(m_{\hat{y}})_{\hat{y} \in \mathcal{Y}}$ as defined in Eq. (2). For a candidate finite-sample count vector $\mathbf{J} = \mathbf{j}$, introduce latent table rows $\mathbf{U}_y = (U_{y\hat{y}})_{\hat{y} \in \mathcal{Y}}$ that allocate the j_y examples of true class y to predicted labels $\hat{y}$. A direct model contains the conditional table likelihoods

$$\begin{aligned} \mathbb{P}(\mathbf{U}_y = \mathbf{u}_y \mid J_y = j_y, p_{\hat{Y}|Y=y}) &= \text{Mult}(\mathbf{u}_y; j_y, p_{\hat{Y}|Y=y}) \quad \forall y \in \mathcal{Y}, \\ M_{\hat{y}} &= \sum_{y \in \mathcal{Y}} U_{y\hat{y}} \quad \forall \hat{y} \in \mathcal{Y}. \end{aligned} \quad (15)$$

Let $\mathcal{U}(\mathbf{j}, \mathbf{m})$ denote the set of nonnegative contingency tables with row margins $\mathbf{j}$ and column margins $\mathbf{m}$,

$$\mathcal{U}(\mathbf{j}, \mathbf{m}) = \left\{ \mathbf{u} \in \mathbb{N}_0^{C \times C} : \begin{array}{ll} \sum_{\hat{y} \in \mathcal{Y}} u_{y\hat{y}} = j_y & \forall y \in \mathcal{Y}, \\ \sum_{y \in \mathcal{Y}} u_{y\hat{y}} = m_{\hat{y}} & \forall \hat{y} \in \mathcal{Y} \end{array} \right\}. \quad (16)$$

Consequently, the target likelihood contribution for a candidate $\mathbf{j}$ is

$$\begin{aligned} \ell_{\text{FS}}(\mathbf{j}, p_{\hat{Y}|Y}) &= \mathbb{P}(\mathbf{M} = \mathbf{m} \mid \mathbf{J} = \mathbf{j}, p_{\hat{Y}|Y}) \\ &= \sum_{\mathbf{u} \in \mathcal{U}(\mathbf{j}, \mathbf{m})} \prod_{y \in \mathcal{Y}} \text{Mult}(\mathbf{u}_y; j_y, p_{\hat{Y}|Y=y}). \end{aligned} \quad (17)$$

Thus, finite-sample BCC is obtained from Eq. (12) by replacing the prior factor $\pi_q(q_Y)$ with the finite-count prior $\pi_{\mathbf{J}}(\mathbf{j})$, and by replacing the target likelihood $\ell_U(q_Y, p_{\hat{Y}|Y})$ with $\ell_{\text{FS}}(\mathbf{j}, p_{\hat{Y}|Y})$. For a candidate count vector $\mathbf{j}$, its posterior is

$$\begin{aligned} \pi_{\text{FS-BCC}}(p_Y, p_{\hat{Y}|Y}, \mathbf{J} = \mathbf{j} \mid \mathcal{D}_L, \mathcal{X}_U) \\ \propto \pi_p(p_Y) \pi_h(p_{\hat{Y}|Y}) \pi_{\mathbf{J}}(\mathbf{j}) \ell_Y(p_Y) \ell_h(p_{\hat{Y}|Y}) \ell_{\text{FS}}(\mathbf{j}, p_{\hat{Y}|Y}). \end{aligned} \quad (18)$$

This direct model makes clear why the finite-sample formulation is computationally expensive. The posterior is mixed discrete-continuous: HMC algorithms such as the NUTS sampler used for continuous Bayesian quantification [10] cannot directly update $\mathbf{J}$ as a differentiable parameter. One must either sample the finite count vector with separate discrete moves or marginalize it out. Both options require evaluating probabilities for the observed predicted-label counts, and those probabilities involve sums over latent contingency tables as in Eq. (17).

3.2 Finite-sample Uncertainty-propagating ACC

Next, we address the computational challenges of direct finite-sample BCC by introducing *finite-sample uncertainty-propagating ACC* (FS-UP-ACC), shown in Fig. 1(c). The underlying calibration assumption is the same as in ACC and BCC: for a fixed classifier h , predicted labels are generated from stable class-conditional confusion rates. That is, conditional on $Y = y$, classifier outputs are

modeled as conditionally independent draws from a categorical distribution with probability vector $p_{\hat{Y}|Y=y}$ from Eq. (6). For the labeled source data, this gives the multinomial calibration likelihood ℓ_h in Eq. (10); for the target data, the same probabilities govern the latent table rows $\mathbf{U}_y$ conditional on J_y in Eq. (15). Under PPS, the source-calibrated probabilities can therefore be used for the target sample. ACC estimates these row probabilities from labeled source data and plugs them into a correction formula. BCC and finite-sample BCC instead assign priors to these probabilities; in the fully joint finite-sample model, the unlabeled target predictions can also update them through $\ell_{\text{FS}}(\mathbf{j}, p_{\hat{Y}|Y})$.

FS-UP-ACC keeps the same multinomial confusion-rate model, but imposes the ACC-style modularization: source labels calibrate the classifier, while target predicted counts are used to infer the finite-sample count vector $\mathbf{J}$. Graphically, this is the cut edge in Fig. 1(c): target predictions depend on calibration uncertainty, but they do not update the calibration posterior. The nuisance source-label-distribution block $\pi_p(p_Y)\ell_Y(p_Y)$ from Eq. (18) is also dropped, since it is separate from the finite-sample target and calibration terms. In terms of the direct posterior in Eq. (18), the cut removes the finite-sample target likelihood $\ell_{\text{FS}}(\mathbf{j}, p_{\hat{Y}|Y})$ from the calibration update and retains only the BCC calibration block,

$$\pi_{\text{cut}}(p_{\hat{Y}|Y} \mid \mathcal{D}_L) \propto \pi_h(p_{\hat{Y}|Y}) \ell_h(p_{\hat{Y}|Y}). \quad (19)$$

Thus, relative to finite-sample BCC, FS-UP-ACC removes feedback from the unlabeled target batch into the classifier-error model while keeping the same finite-sample target chain $\mathbf{S} \rightarrow \mathbf{J} \rightarrow \mathbf{U} \rightarrow \mathbf{M}$.

To propagate calibration uncertainty under this cut, we instantiate the prior π_h over the class-conditional confusion rates. Since each row $p_{\hat{Y}|Y=y}$ parameterizes a categorical model for classifier outputs and the calibration likelihood ℓ_h is multinomial row by row, we use independent symmetric Dirichlet row priors $p_{\hat{Y}|Y=y} \sim \text{Dir}((\lambda)_{\hat{y} \in \mathcal{Y}})$ with $\lambda > 0$, where Dir denotes the Dirichlet distribution. Under the cut, these priors are conjugate to the calibration likelihood; this yields a closed-form representation of calibration uncertainty. Combining Eq. (19) with the source joint counts $n_{y\hat{y}}$ from Eq. (8) gives the row-wise posterior

$$p_{\hat{Y}|Y=y} \mid \mathcal{D}_L \sim \text{Dir}((n_{y\hat{y}} + \lambda)_{\hat{y} \in \mathcal{Y}}) \quad \forall y \in \mathcal{Y}. \quad (20)$$

For target inference, note that, conditional on $J_y = j_y$ and $p_{\hat{Y}|Y=y}$, each row $\mathbf{U}_y$ is multinomial. After integrating the Dirichlet posterior in Eq. (20), its posterior predictive distribution is Dirichlet-multinomial:

$$\mathbf{U}_y \mid J_y = j_y, \mathcal{D}_L \sim \text{DirichletMultinomial}(j_y, (n_{y\hat{y}} + \lambda)_{\hat{y} \in \mathcal{Y}}). \quad (21)$$

The observed predicted-label counts are the column sums of the latent table, so the feasible latent tables are the fixed-margin tables $\mathcal{U}(\mathbf{j}, \mathbf{m})$ from Eq. (16). Writing $\text{DM}(\cdot; n, \boldsymbol{\alpha})$ for the Dirichlet-multinomial PMF, the finite-sample predictive

likelihood is

$$\mathbb{P}(\mathbf{M} = \mathbf{m} \mid \mathbf{J} = \mathbf{j}, \mathcal{D}_L) = \sum_{\mathbf{u} \in \mathcal{U}(\mathbf{j}, \mathbf{m})} \prod_{y \in \mathcal{Y}} \text{DM}(\mathbf{u}_y; j_y, (n_{y\hat{y}} + \lambda)_{\hat{y} \in \mathcal{Y}}). \quad (22)$$

Combining this likelihood with the finite-count prior $\pi_{\mathbf{J}}$ gives

$$\mathbb{P}(\mathbf{J} = \mathbf{j} \mid \mathbf{M} = \mathbf{m}, \mathcal{D}_L) = \frac{\pi_{\mathbf{J}}(\mathbf{j}) \mathbb{P}(\mathbf{M} = \mathbf{m} \mid \mathbf{J} = \mathbf{j}, \mathcal{D}_L)}{\sum_{\mathbf{j}' \in \mathcal{J}_{m,C}} \pi_{\mathbf{J}}(\mathbf{j}') \mathbb{P}(\mathbf{M} = \mathbf{m} \mid \mathbf{J} = \mathbf{j}', \mathcal{D}_L)}. \quad (23)$$

The posterior mass assigned to the finite-sample label-distribution vector $\mathbf{j}/m$ is this same quantity.

The closed-form posterior in Eq. (23) is the FS-UP-ACC update. Its support is the finite grid of realizable target-sample label distributions. However, exact computation is expensive. The number of candidate count vectors is $\binom{m+C-1}{C-1}$, and each likelihood evaluation in Eq. (22) requires summing over constrained nonnegative integer tables. The closed-form expression is therefore useful conceptually, but not practical as a multiclass algorithm for large m or C .

3.3 A Tractable Binary Special Case

The multiclass update in Eq. (23) has a closed form, but is not yet a practical general-purpose multiclass quantifier: exact evaluation still ranges over all candidate count vectors and all compatible fixed-margin tables. For binary crisp classification, this same construction simplifies substantially. There is only one free target count, and each compatible target table is determined by one latent count. Thus, the closed-form FS-UP-ACC update from Section 3.2 becomes an efficient uncertainty-aware binary quantifier.

Let $\mathcal{Y} = \{0, 1\}$, with class 1 denoting the positive class, and write

$$J = J_1, \quad K = M_1, \quad k = m_1.$$

Here, J is the random number of positives in the target sample, $S = J/m$, K is the random number of predicted positives, and k is its realization. The finite target grid is therefore $j \in \{0, \dots, m\}$. Let $\text{tp} = n_{11}$, $\text{fn} = n_{10}$, $\text{fp} = n_{01}$, and $\text{tn} = n_{00}$ denote the realized binary source confusion counts. The row-wise Dirichlet posteriors from Eq. (20) reduce to beta posteriors for the true-positive and false-positive rates from Eq. (4):

$$a \mid \mathcal{D}_L \sim \text{Beta}(\text{tp} + \lambda, \text{fn} + \lambda), \quad b \mid \mathcal{D}_L \sim \text{Beta}(\text{fp} + \lambda, \text{tn} + \lambda). \quad (24)$$

For a candidate finite-sample count $J = j$, let $U = U_{11}$ be the number of positive target examples predicted as positive, and let $V = U_{01}$ be the number of negative target examples predicted as positive. Then $K = U + V$. Integrating the beta calibration posteriors gives the binary analogue of Eq. (21), namely the beta-binomial posterior predictive distributions

$$\begin{aligned} U \mid J = j, \mathcal{D}_L &\sim \text{BetaBinomial}(j, \text{tp} + \lambda, \text{fn} + \lambda), \\ V \mid J = j, \mathcal{D}_L &\sim \text{BetaBinomial}(m - j, \text{fp} + \lambda, \text{tn} + \lambda). \end{aligned} \quad (25)$$

Consequently, the likelihood of the observed predicted-positive count is the one-dimensional convolution

$$\mathbb{P}(K = k \mid J = j, \mathcal{D}_L) = \sum_{u=u_{\min}}^{u_{\max}} \text{BB}(u; j, \text{tp} + \lambda, \text{fn} + \lambda) \cdot \text{BB}(k - u; m - j, \text{fp} + \lambda, \text{tn} + \lambda), \quad (26)$$

where $\text{BB}(\cdot; n, \alpha, \beta)$ is the beta-binomial PMF and

$$u_{\min} = \max(0, k - (m - j)), \quad u_{\max} = \min(j, k).$$

Combining this likelihood with a prior $\pi_j = \mathbb{P}(J = j)$ on the binary finite count grid gives

$$\mathbb{P}(J = j \mid K = k, \mathcal{D}_L) = \frac{\pi_j \mathbb{P}(K = k \mid J = j, \mathcal{D}_L)}{\sum_{j'=0}^m \pi_{j'} \mathbb{P}(K = k \mid J = j', \mathcal{D}_L)}. \quad (27)$$

The posterior over the finite-sample prevalence is trivially obtained by rescaling the count grid:

$$S \in \left\{0, \frac{1}{m}, \dots, 1\right\}, \quad \mathbb{P}\left(S = \frac{j}{m} \mid K = k, \mathcal{D}_L\right) = \mathbb{P}(J = j \mid K = k, \mathcal{D}_L). \quad (28)$$

This binary specialization is cheap to compute because exact inference requires only $m + 1$ candidate values of J and, for each candidate, one finite sum over $u_{\min}, \dots, u_{\max}$. There is no continuous posterior sampling and no enumeration of high-dimensional fixed-margin tables. The multiclass formula in Eq. (23) shows the same finite-sample principle, but a practical multiclass variant would require additional algorithmic optimizations or approximations such as dynamic programming, sparse enumeration, or (discrete) Monte Carlo methods.

4 Evaluation

To evaluate the proposed FS-UP-ACC, we focus on the binary variant described in Section 3.3 and compare it against ACC and BCC. We do not compare against quantification methods that have access to richer distributional information, e.g., in the form of a probabilistic classifier or stronger shift assumptions; this includes PACC [1], as well as *distribution matching* methods, such as the *Mixture Model* approach [6], HDy [9], DyS [15], or KDEy [17]. While the additional information available to these methods can improve quantification performance in practice, our goal is not to demonstrate that the proposed FS-UP-ACC method is generally superior. Instead, we aim to show the following:

1. Finite-sample and population-level quantification are two distinct problems and require distinct approaches.
2. Given a hard classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$, FS-UP-ACC is a more appropriate method for finite-sample quantification than existing aggregative quantifiers using the same hard classifier h .

Since the multiclass methods described in Sections 3.1 and 3.2 require additional algorithmic optimizations or approximations to make them computationally feasible, we only evaluate the binary case and leave the development of such methods to subsequent research.

4.1 Synthetic Evaluation

We first isolate finite-sample quantification in a controlled count-level experiment. Instead of generating feature vectors, we simulate the output of a fixed binary classifier directly. For each requested prevalence p , we draw a random finite target count $J \sim \text{Binomial}(m, p)$ and evaluate estimates against the realized finite-sample prevalence J/m . Thus, the population prevalence p is only a data-generating parameter; it is not the evaluation target. This deliberately focuses the synthetic evaluation on the estimand addressed by FS-UP-ACC.

We model the synthetic binary classifier using a true-positive rate a and a false-positive rate b . Given a target sample of size m with j positive examples, the observed number of predicted positives is generated as

$$K = U + V, \quad \text{where } U \sim \text{Binomial}(j, a) \text{ and } V \sim \text{Binomial}(m - j, b).$$

The calibration data are generated analogously from the same classifier channel: for each class, we draw $n_{\text{cal}} = 1000$ calibration examples and record the corresponding classifier-output counts. This setup removes feature-modeling effects and directly tests whether a method is aligned with the finite-sample target. We vary the classifier TPR a and FPR b , the target sample size m , and the population sampling prevalence p . For each configuration of the resulting grid, calibration and target samples are generated and evaluated 30 times. To summarize the posterior distributions over prevalences by a single point estimate, we use the posterior mean for both methods. For uncertainty summaries, both methods use central 95% posterior credible intervals. The goal of this overall setup is to compare the differences between BCC and FS-UP-ACC.

Figure 2 summarizes the comparison. Each panel reports FS-UP-ACC minus BCC; negative values therefore favor FS-UP-ACC. The point-error panels in the top row show that the small-sample cells are consistently negative for both *absolute error* (AE) and *Kullback-Leibler divergence* (KLD), with the largest advantages at $m = 10$ and $m = 25$. This is exactly the regime in which the realized prevalence J/m can differ substantially from the population prevalence p . A population-target method such as BCC is therefore systematically mismatched to the finite-sample loss, while FS-UP-ACC places its posterior mass directly on the finite count J . As m grows, the sampling variation in J/m decreases and the differences move toward zero or become mixed. The figure therefore supports both evaluation claims: finite-sample quantification is a distinct target, and it is best addressed by a finite-sample estimator, especially for small target samples.

The panels in the bottom row of Fig. 2 compare FS-UP-ACC and BCC from the perspective of coverage and interval size. In the small-sample regime, FS-UP-ACC has smaller absolute coverage gaps than BCC, meaning that its nominal

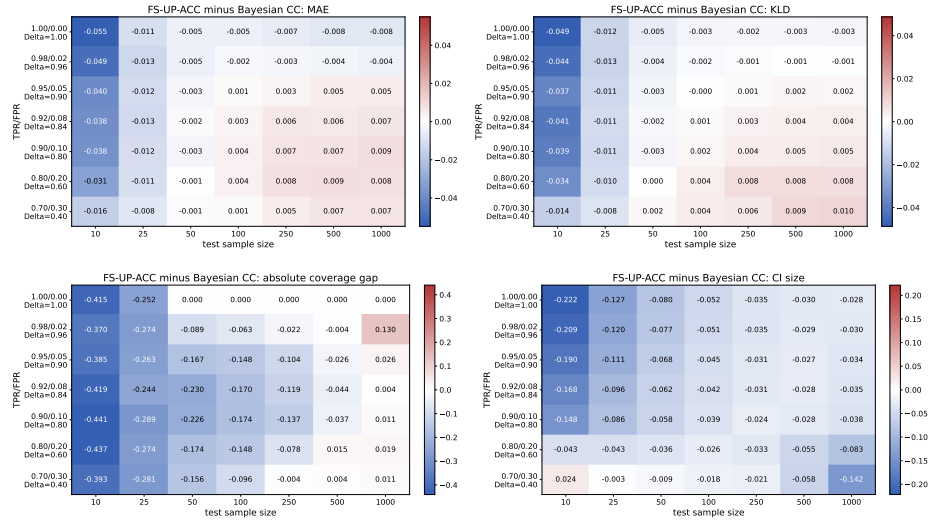


Figure 2. Synthetic comparison on the random finite-sample target. Each cell shows FS-UP-ACC minus BCC; negative values favor FS-UP-ACC. Top row: mean AE and mean KLD. Bottom row: absolute coverage gap from nominal 95% central credible-interval coverage and average central credible-interval size. Rows within each panel vary the classifier TPR and FPR.

95% intervals are closer to their intended coverage. At the same time, the intervals are almost always shorter. The uncertainty results therefore reinforce the point-estimate results: when the target is the realized finite-sample prevalence, modeling the finite count is most valuable precisely when the target sample is small.

4.2 Real-data Evaluation

We next ask whether the same small-sample pattern appears with real classifiers. The benchmark contains 29 binary datasets from the QuaPy UCI collection [16]. For the UCI datasets, we use a train/validation/test-pool split with proportions 0.50/0.25/0.25. The classifier is a logistic-regression pipeline with median imputation and standardization for numeric features, and most-frequent imputation plus one-hot encoding for categorical features. The validation split is used to estimate the class-conditional classifier-output probabilities required by ACC, BCC, and FS-UP-ACC.

Target samples are generated using the *artificial prevalence protocol* (APP): for requested prevalences 0.00, 0.04, $\dots$, 1.00, we draw small target samples of size $m = 25$ from the test pool, with 25 repeats per prevalence. The experiment is repeated for three split seeds; in all cases the evaluation target is the realized sample prevalence. We report *absolute error* (AE) and *Kullback-Leibler divergence* (KLD), as well as 95% central posterior credible-interval width and

Table 1. Per-dataset small-sample real-data point-estimate performance. Entries are means $\pm$ standard errors over repeated APP runs. Lower values are better; bold values mark the minimum-mean method and methods whose mean is not significantly higher than that minimum with one-sided 95% normal-approximation confidence. The final row reports the number of datasets won or tied by each method under the same criterion. FUA denotes FS-UP-ACC.

Dataset	AE				KLD			
	CC	ACC	BCC	FUA	CC	ACC	BCC	FUA
balance.1	0.039 $\pm$ 0.001	0.035 $\pm$ 0.001	0.039 $\pm$ 0.001	0.035 $\pm$ 0.001	0.014 $\pm$ 0.001	0.042 $\pm$ 0.005	0.011 $\pm$ 0.000	0.008 $\pm$ 0.000
balance.3	0.041 $\pm$ 0.001	0.036 $\pm$ 0.001	0.043 $\pm$ 0.001	0.037 $\pm$ 0.001	0.014 $\pm$ 0.001	0.020 $\pm$ 0.003	0.012 $\pm$ 0.000	0.009 $\pm$ 0.000
breast-cancer	0.020 $\pm$ 0.001	0.025 $\pm$ 0.000	0.024 $\pm$ 0.000	0.025 $\pm$ 0.000	0.006 $\pm$ 0.001	0.029 $\pm$ 0.004	0.007 $\pm$ 0.000	0.005 $\pm$ 0.000
cmc.1	0.223 $\pm$ 0.004	0.215 $\pm$ 0.004	0.176 $\pm$ 0.003	0.173 $\pm$ 0.003	0.182 $\pm$ 0.005	2.066 $\pm$ 0.081	0.121 $\pm$ 0.003	0.119 $\pm$ 0.003
cmc.2	0.418 $\pm$ 0.006	0.317 $\pm$ 0.006	0.224 $\pm$ 0.003	0.223 $\pm$ 0.003	1.481 $\pm$ 0.059	5.482 $\pm$ 0.163	0.178 $\pm$ 0.005	0.178 $\pm$ 0.005
cmc.3	0.347 $\pm$ 0.006	0.336 $\pm$ 0.006	0.235 $\pm$ 0.003	0.235 $\pm$ 0.003	0.538 $\pm$ 0.025	5.918 $\pm$ 0.166	0.188 $\pm$ 0.005	0.189 $\pm$ 0.005
ctg.1	0.115 $\pm$ 0.002	0.072 $\pm$ 0.001	0.072 $\pm$ 0.001	0.069 $\pm$ 0.001	0.064 $\pm$ 0.002	0.092 $\pm$ 0.008	0.029 $\pm$ 0.001	0.027 $\pm$ 0.001
ctg.2	0.207 $\pm$ 0.003	0.101 $\pm$ 0.002	0.104 $\pm$ 0.002	0.100 $\pm$ 0.002	0.168 $\pm$ 0.005	0.231 $\pm$ 0.018	0.051 $\pm$ 0.002	0.048 $\pm$ 0.001
ctg.3	0.157 $\pm$ 0.003	0.081 $\pm$ 0.002	0.080 $\pm$ 0.001	0.075 $\pm$ 0.001	0.118 $\pm$ 0.005	0.058 $\pm$ 0.005	0.035 $\pm$ 0.001	0.032 $\pm$ 0.001
german	0.243 $\pm$ 0.004	0.168 $\pm$ 0.003	0.152 $\pm$ 0.002	0.149 $\pm$ 0.002	0.222 $\pm$ 0.006	1.285 $\pm$ 0.062	0.095 $\pm$ 0.003	0.093 $\pm$ 0.003
haberman	0.392 $\pm$ 0.006	0.261 $\pm$ 0.005	0.236 $\pm$ 0.003	0.235 $\pm$ 0.003	0.863 $\pm$ 0.036	3.867 $\pm$ 0.127	0.185 $\pm$ 0.004	0.184 $\pm$ 0.004
ionosphere	0.167 $\pm$ 0.003	0.110 $\pm$ 0.002	0.109 $\pm$ 0.002	0.105 $\pm$ 0.002	0.109 $\pm$ 0.003	0.069 $\pm$ 0.004	0.054 $\pm$ 0.002	0.051 $\pm$ 0.001
iris.2	0.214 $\pm$ 0.004	0.195 $\pm$ 0.004	0.181 $\pm$ 0.003	0.177 $\pm$ 0.003	0.177 $\pm$ 0.005	0.935 $\pm$ 0.048	0.129 $\pm$ 0.004	0.126 $\pm$ 0.004
iris.3	0.049 $\pm$ 0.001	0.044 $\pm$ 0.001	0.053 $\pm$ 0.001	0.047 $\pm$ 0.001	0.021 $\pm$ 0.002	0.027 $\pm$ 0.003	0.017 $\pm$ 0.000	0.014 $\pm$ 0.000
mammographic	0.089 $\pm$ 0.001	0.097 $\pm$ 0.002	0.090 $\pm$ 0.001	0.089 $\pm$ 0.002	0.041 $\pm$ 0.001	0.442 $\pm$ 0.028	0.039 $\pm$ 0.001	0.040 $\pm$ 0.001
pageblocks.5	0.380 $\pm$ 0.005	0.197 $\pm$ 0.004	0.173 $\pm$ 0.003	0.172 $\pm$ 0.003	0.869 $\pm$ 0.028	0.515 $\pm$ 0.030	0.121 $\pm$ 0.003	0.123 $\pm$ 0.004
semeion	0.092 $\pm$ 0.002	0.050 $\pm$ 0.001	0.059 $\pm$ 0.001	0.054 $\pm$ 0.001	0.045 $\pm$ 0.002	0.114 $\pm$ 0.012	0.020 $\pm$ 0.001	0.016 $\pm$ 0.001
sonar	0.129 $\pm$ 0.002	0.110 $\pm$ 0.002	0.119 $\pm$ 0.002	0.116 $\pm$ 0.002	0.070 $\pm$ 0.002	0.409 $\pm$ 0.028	0.059 $\pm$ 0.001	0.057 $\pm$ 0.001
spambase	0.061 $\pm$ 0.001	0.049 $\pm$ 0.001	0.051 $\pm$ 0.001	0.048 $\pm$ 0.001	0.024 $\pm$ 0.001	0.062 $\pm$ 0.006	0.017 $\pm$ 0.000	0.015 $\pm$ 0.001
spectf	0.291 $\pm$ 0.005	0.255 $\pm$ 0.005	0.213 $\pm$ 0.003	0.210 $\pm$ 0.003	0.341 $\pm$ 0.013	3.543 $\pm$ 0.134	0.162 $\pm$ 0.004	0.160 $\pm$ 0.004
tictactoe	0.052 $\pm$ 0.001	0.035 $\pm$ 0.001	0.041 $\pm$ 0.001	0.035 $\pm$ 0.001	0.020 $\pm$ 0.001	0.027 $\pm$ 0.004	0.012 $\pm$ 0.001	0.009 $\pm$ 0.000
transfusion	0.435 $\pm$ 0.006	0.296 $\pm$ 0.005	0.223 $\pm$ 0.003	0.222 $\pm$ 0.003	2.093 $\pm$ 0.085	5.327 $\pm$ 0.150	0.169 $\pm$ 0.004	0.168 $\pm$ 0.004
wdbc	0.033 $\pm$ 0.001	0.040 $\pm$ 0.001	0.042 $\pm$ 0.001	0.040 $\pm$ 0.001	0.012 $\pm$ 0.001	0.018 $\pm$ 0.002	0.012 $\pm$ 0.000	0.010 $\pm$ 0.000
wine-q-red	0.152 $\pm$ 0.002	0.129 $\pm$ 0.002	0.126 $\pm$ 0.002	0.123 $\pm$ 0.002	0.092 $\pm$ 0.002	0.412 $\pm$ 0.028	0.071 $\pm$ 0.002	0.069 $\pm$ 0.002
wine-q-white	0.226 $\pm$ 0.004	0.143 $\pm$ 0.003	0.134 $\pm$ 0.002	0.131 $\pm$ 0.002	0.187 $\pm$ 0.005	0.705 $\pm$ 0.039	0.079 $\pm$ 0.002	0.077 $\pm$ 0.002
wine.1	0.000 $\pm$ 0.000	0.022 $\pm$ 0.001	0.020 $\pm$ 0.000	0.021 $\pm$ 0.000	0.000 $\pm$ 0.000	0.029 $\pm$ 0.003	0.005 $\pm$ 0.000	0.003 $\pm$ 0.000
wine.2	0.010 $\pm$ 0.001	0.022 $\pm$ 0.001	0.023 $\pm$ 0.001	0.024 $\pm$ 0.001	0.004 $\pm$ 0.001	0.007 $\pm$ 0.001	0.006 $\pm$ 0.000	0.004 $\pm$ 0.000
wine.3	0.019 $\pm$ 0.001	0.031 $\pm$ 0.001	0.040 $\pm$ 0.001	0.036 $\pm$ 0.001	0.008 $\pm$ 0.001	0.059 $\pm$ 0.006	0.011 $\pm$ 0.000	0.008 $\pm$ 0.000
yeast	0.309 $\pm$ 0.005	0.184 $\pm$ 0.004	0.162 $\pm$ 0.003	0.159 $\pm$ 0.003	0.392 $\pm$ 0.015	1.124 $\pm$ 0.056	0.107 $\pm$ 0.003	0.104 $\pm$ 0.003
# wins or ties	6/29	9/29	13/29	21/29	4/29	0/29	15/29	28/29

coverage for the Bayesian methods. For both BCC and FS-UP-ACC, we use the posterior mean as the point estimate; for FS-UP-ACC, this mean is taken over the finite-sample posterior. The tables below first average over samples and split seeds within each dataset; the aggregate row shows the number of times each method performs best or is tied with the best-performing method.

Table 1 shows the per-dataset point-estimate results. The wins-or-ties row clearly shows that FS-UP-ACC is competitive on substantially more datasets than BCC: 21/29 versus 13/29 under AE, and 28/29 versus 15/29 under KLD. This mirrors the synthetic result: when target samples are small and the evaluation target is the realized sample prevalence, the finite-sample posterior is the better match. The individual rows also show that the advantage is not uniform, while FS-UP-ACC does not dominate every task, it is the most reliable hard-classifier method in the finite-sample regime.

The results in Table 2 show the same uncertainty pattern as the synthetic plots. Compared with BCC, FS-UP-ACC produces tighter intervals on average while also moving coverage closer to the nominal 95% level. Thus, the real-data evidence supports the same conclusion as the synthetic study: finite-sample quantification is best handled by a method whose posterior target is the sample prevalence, and the advantage is most visible when the target sample is small.

Table 2. Small-sample real-data central posterior credible-interval performance. Entries are equal-dataset means $\pm$ standard errors. Coverage is empirical coverage of nominal 95% central credible intervals.

Method	Credible interval width	Coverage
BCC	0.575 ± 0.041	0.912 ± 0.003
FS-UP-ACC	0.509 ± 0.055	0.980 ± 0.003

5 Conclusion

We argue that quantification has two related but distinct targets: the population label distribution and the label distribution realized in a particular finite target sample. The latter is the relevant estimand in many applications and in common finite-sample evaluation protocols, but it is not the posterior target of standard population-level methods such as ACC or BCC. To address this mismatch, we introduced finite-sample quantification and developed FS-UP-ACC, a Bayesian hard-classifier quantifier that places posterior mass directly on feasible finite target counts while propagating calibration uncertainty. We found that, across controlled synthetic experiments and small-sample real-data benchmarks, binary FS-UP-ACC generally improves over BCC when evaluated against the realized finite-sample target. This work focuses on the special case of hard binary classifier outputs and is based on the commonly made PPS assumption. For future work, we envision a more general framework that extends FS-UP-ACC efficiently to the multiclass setting, to probabilistic classifier outputs, and other types of distribution shift (e.g., covariate [21], structural [3], or concept shift).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bella, A., Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.J.: Quantification via Probability Estimators. In: IEEE International Conference on Data Mining, pp. 737–742 (2010), ISSN 2374-8486, <https://doi.org/10.1109/ICDM.2010.75>
2. Bunse, M.: On multi-class extensions of adjusted classify and count. In: The 2nd International Workshop on Learning to Quantify (LQ 2022), pp. 43–50 (2022)
3. Damke, C., Hüllermeier, E.: Adjusted Count Quantification Learning on Graphs. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
4. Esuli, A., Fabris, A., Moreo, A., Sebastiani, F.: Learning to Quantify, The Information Retrieval Series, vol. 1. Springer, Cham (2023), ISBN 978-3-031-20467-8
5. Firat, A.: Unified Framework for Quantification (2016), <https://doi.org/10.48550/arXiv.1606.00868>
6. Forman, G.: Counting positives accurately despite inaccurate classification. In: ECML, pp. 564–575, ECML’05, Springer-Verlag, Berlin, Heidelberg (2005), ISBN 978-3-540-29243-2, https://doi.org/10.1007/11564096_55

7. Forman, G.: Quantifying trends accurately despite classifier error and class imbalance. In: KDD, pp. 157–166, KDD '06, Association for Computing Machinery, New York, NY, USA (2006), ISBN 978-1-59593-339-3
8. Forman, G.: Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery* **17**(2), 164–206 (2008), ISSN 1573-756X, <https://doi.org/10.1007/s10618-008-0097-y>
9. González-Castro, V., Alaiz-Rodríguez, R., Alegre, E.: Class distribution estimation based on the Hellinger distance. *Information Sciences* **218**, 146–164 (2013), ISSN 0020-0255, <https://doi.org/10.1016/j.ins.2012.05.028>
10. Hoffman, M.D., Gelman, A.: The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.* **15**(47), 1593–1623 (2014), ISSN 1533-7928
11. Hopkins, D.J., King, G.: A Method of Automated Nonparametric Content Analysis for Social Science. *American Journal of Political Science* **54**(1), 229–247 (2010), ISSN 1540-5907, <https://doi.org/10.1111/j.1540-5907.2009.00428.x>
12. Igraneza, A.B., Fraser, C., Hinch, R.: Estimating prevalence with precision and accuracy (2025), <https://doi.org/10.48550/arXiv.2507.06061>
13. Keith, K., O'Connor, B.: Uncertainty-aware generative models for inferring document class prevalence. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4575–4585 (2018)
14. Lipton, Z., Wang, Y.X., Smola, A.: Detecting and Correcting for Label Shift with Black Box Predictors. In: *ICML*, pp. 3122–3130, PMLR (2018), ISSN 2640-3498
15. Maletzke, A., dos Reis, D., Cherman, E., Batista, G.: DyS: A Framework for Mixture Models in Quantification. *Proceedings of the AAAI Conference on Artificial Intelligence* **33**(01), 4552–4560 (2019), ISSN 2374-3468
16. Moreo, A., Esuli, A., Sebastiani, F.: QuaPy: A Python-Based Framework for Quantification. In: *CIKM*, pp. 4534–4543, *CIKM '21*, Association for Computing Machinery, New York, NY, USA (2021), ISBN 978-1-4503-8446-9, <https://doi.org/10.1145/3459637.3482015>
17. Moreo, A., González, P., del Coz, J.J.: Kernel density estimation for multiclass quantification. *Machine Learning* **114**(4), 1–38 (2025), ISSN 1573-0565, <https://doi.org/10.1007/s10994-024-06726-5>
18. Saerens, M., Latinne, P., Decaestecker, C.: Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure. *Neural Computation* **14**(1), 21–41 (2002), ISSN 0899-7667, <https://doi.org/10.1162/089976602753284446>
19. Tasche, D.: Fisher consistency for prior probability shift. *J. Mach. Learn. Res.* **18**(1), 3338–3369 (2017), ISSN 1532-4435
20. Tasche, D.: Confidence Intervals for Class Prevalences under Prior Probability Shift. *Machine Learning and Knowledge Extraction* **1**(3), 805–831 (2019), ISSN 2504-4990, <https://doi.org/10.3390/make1030047>
21. Tasche, D.: Class Prior Estimation under Covariate Shift: No Problem? (2022), <https://doi.org/10.48550/arXiv.2206.02449>
22. Vaz, A., Izbicki, R., Stern, R.B.: Prior Shift Using the Ratio Estimator. In: Polpo, A., Stern, J., Louzada, F., Izbicki, R., Takada, H. (eds.) *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, pp. 25–35, Springer International Publishing, Cham (2018), ISBN 978-3-319-91143-4
23. Vucetic, S., Obradovic, Z.: Classification on Data with Biased Class Distribution. In: De Raedt, L., Flach, P. (eds.) *ECML*, pp. 527–538, Springer, Berlin, Heidelberg (2001), ISBN 978-3-540-44795-5, https://doi.org/10.1007/3-540-44795-4_45
24. Ziegler, A., Czyż, P.: Bayesian Quantification with Black-Box Estimators. *Transactions on Machine Learning Research* (2024), ISSN 2835-8856