

On Binary Quantification Calibration

Isabella Caroline Sachini Lorena¹, André Maletzke², and Willian Zalewski^{1,3}

Western Paraná State University, Foz do Iguaçu, Brazil

{isabella.lorena, andre.maletzke}@unioeste.br

Federal University for Latin American Integration, Foz do Iguaçu, Brazil

willian.zalewski@unila.edu.br

Abstract. Quantification is a supervised learning task whose goal is to estimate the class distribution of unlabeled samples. Many quantification methods rely on a classifier as an intermediate component and therefore inherit its biases, especially under label shift, where class prevalences differ between training and test data. Calibration, the requirement that a predicted quantity match its true value on average, has been studied extensively for classifiers, but the literature has treated it almost exclusively as a property of the classifier rather than of the quantifier itself. In this work, we extend the classical calibration condition from classifiers to quantifiers, and show that existing correction methods such as Adjusted Classify and Count (ACC) and its probabilistic counterpart PACC already instantiate this notion, though only through a fixed parametric correction tied to a specific classifier error-rate model. Building on this formalization, we propose *QuantCal* (QCal), a post-hoc calibration method that learns a non-parametric correction directly from a quantifier’s own prevalence estimates, without assuming a fixed functional form and without requiring a separate derivation for each quantifier. We evaluate QCal on 106 binary datasets across eight base quantifiers. The results show that QCal significantly reduces the mean absolute error of every base quantifier considered.

Keywords: Quantification · Calibration · Label shift

1 Introduction

Calibration is a general requirement for any predictor that outputs a quantity meant to be interpreted probabilistically: the predicted value should match the true value it purports to estimate, on average, across all instances for which that same value is predicted. This notion, originating in the evaluation of probabilistic weather forecasts [18,19,20], has been studied extensively for classifiers [22], where it requires that, among all instances assigned a given confidence score, the fraction correctly classified matches that score. A wide range of post-hoc techniques has been developed to enforce this property when it fails to hold, correcting a classifier’s scores after training without altering the underlying model [26].

Quantification, or counting, is a supervised learning task whose goal is to estimate the class distribution of an unlabeled sample, rather than the class of

each individual instance [6,8]. Most quantification methods rely on a classifier as an intermediate step, either through its hard predictions or its posterior probabilities. The canonical baseline, *Classify and Count* (CC) [6], simply counts the classifier’s predictions, but this strategy is known to produce systematically biased estimates whenever the class distribution shifts between training and test data [8]. This limitation motivated a large body of work specifically designed for quantification, establishing it as a task in its own right rather than a by-product of classification [5].

Throughout this literature, calibration is framed almost exclusively as a property of the classifier. Yet methods such as *Adjusted Classify and Count* (ACC) [7] and PACC [3] already correct the quantifier’s own prevalence estimate, using a linear adjustment derived from the classifier’s estimated error rates; this correction, however, has not been framed as an instance of the same calibration principle introduced above, nor generalized beyond its fixed affine form. To the best of our knowledge, no prior work formalizes what it means for a binary quantifier itself to be calibrated. This is the gap addressed in this paper: we extend the classical calibration condition from classifiers to quantifiers, and propose *QuantCal* (QCal), a post-hoc method that learns to correct a binary quantifier’s prevalence estimates directly, without assuming a parametric form for the correction and without requiring a separate derivation for each quantifier. The main contributions of this work are: (i) a formalization of quantifier calibration as an extension of Murphy’s general calibration condition, under which ACC and PACC are shown to be parametric instances of a broader framework; (ii) QCal, a non-parametric instantiation of this framework that estimates the correction function directly from data, and applies without modification to any binary quantifier; and (iii) an empirical evaluation of QCal across 106 binary datasets and eight base quantifiers, assessing its effectiveness as a general-purpose calibration method.

2 Background

2.1 Quantification

Quantification is a supervised learning task whose goal is to estimate the class distribution of an unlabeled sample, rather than to predict the class of individual instances [6,8]. Formally, let $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ be a labeled training set, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y} = \{c_1, \dots, c_\ell\}$. A quantifier q is induced from D such that

$$q : \mathcal{A}^{\mathcal{X}} \rightarrow [0, 1]^\ell, \quad (1)$$

where $\mathcal{A}^{\mathcal{X}}$ denotes the universe of possible samples of $\mathcal{X}$. Given a test sample $a \in \mathcal{A}^{\mathcal{X}}$, the quantifier produces a vector $\hat{p} = [\hat{p}_1, \dots, \hat{p}_\ell]$ with $\sum_j \hat{p}_j = 1$. In the binary case, $\hat{p} = \hat{p}_\oplus$ denotes the estimated prevalence of the positive class, and p the corresponding true prevalence in a [15]. The goal is thus to produce $\hat{p}_\oplus$ as close as possible to p .

Most quantification methods rely on a classifier as an intermediate step, either a probabilistic classifier $h : \mathcal{X} \rightarrow [0, 1]^\ell$ or the corresponding crisp classifier $\hat{h}(x) = \arg \max_j h(x)_j$. The *Classify and Count* (CC) method [6] is the canonical baseline: given $\hat{h}$ trained on D , CC counts the predicted labels in a ,

$$\hat{p}_\oplus^{\text{CC}} = \frac{|\{x \in a : \hat{h}(x) = \oplus\}|}{|a|}. \quad (2)$$

CC has a well-known limitation: classification errors propagate directly to the prevalence estimate, producing systematically biased results whenever the class distribution shifts between training and test data, precisely the condition under which quantification is most needed [6,8]. This motivated the development of methods specifically designed for quantification, establishing it as a task in its own right rather than a by-product of classification.

Since quantification is a relatively recent area, methods have appeared under different assumptions and modeling strategies. *Aggregative* methods rely on the outputs of an underlying classifier, either hard predictions, posterior probabilities, or scores; *non-aggregative* methods estimate prevalences directly, without a classifier as an intermediate step [5].

A first line of work responds directly to CC's bias under label shift, which depends on the classifier's true and false positive rates, $\text{tpr} = P(\hat{h}(x) = \oplus \mid y = \oplus)$ and $\text{fpr} = P(\hat{h}(x) = \oplus \mid y = \ominus)$. Since these rates remain stationary under label shift, *Adjusted Classify and Count* (ACC) estimates them from held-out data to correct CC's raw counts [7]; in practice, however, this estimate can be fragile under imbalanced or small-sample conditions [8]. A second line replaces hard assignments with probabilistic information: PCC and PACC estimate prevalences from posterior probabilities rather than hard predictions [3], and EMQ [24] iteratively adjusts posteriors to account for shifting class priors, though both introduce a stronger dependence on the quality of the classifier's probabilistic outputs. A third, influential line is based on distribution matching, estimating the prevalences that best align the training and test mixtures of class-conditional distributions; recent comparative studies show these methods are highly competitive, although no single method dominates across all types of shift [25,9].

Overall, quantification has moved from simple counting strategies toward methods that explicitly model dataset shift, yet a large part of this literature still depends on the reliability of an underlying classifier's scores or posterior probabilities, a dependence that raises the broader question of calibration.

2.2 Calibration

Formally, a probabilistic predictor is said to be perfectly calibrated if, for every predicted value $v \in [0, 1]$,

$$P(\text{event occurs} \mid \text{prediction} = v) = v. \quad (3)$$

That is, among all occasions on which the predictor outputs v , the event must occur with frequency exactly v . This is a condition of statistical consistency

in aggregate, not per-instance accuracy, and the two notions are independent: a predictor may be well calibrated yet imprecise, or accurate on average yet miscalibrated at specific values.

Classifier calibration. Applied to classifiers by Platt [22], this condition requires that, given a classifier $h : \mathcal{X} \rightarrow [0, 1]^\ell$ with prediction $\hat{y}_i = \arg \max_j h(x_i)_j$ and confidence $\hat{s}_i = \max_j h(x_i)_j$, the fraction of instances correctly classified among those assigned confidence v equals v , for every $v \in [0, 1]$. Post-hoc techniques enforce this by fitting a transformation $\tau : [0, 1]^\ell \rightarrow [0, 1]^\ell$ on held-out data such that

$$h^{\text{cal}}(x_i)_j = \tau(h(x_i))_j, \quad (4)$$

where h^{cal} is the calibrated classifier. Proposed transformations range from parametric families, Platt Scaling [22], Beta calibration [12], Temperature Scaling [10], and Dirichlet calibration [13], to the non-parametric isotonic regression [2, 29], which imposes only monotonicity.

Calibration under dataset shift. When training and test data no longer follow the same distribution, calibration itself must be adapted. Under covariate shift, TransCal [28] and CPCS [21] reweight validation examples by importance weights. Under label shift, Bias-Corrected Temperature Scaling (BCTS) [1] augments Temperature Scaling with per-class bias terms, and LasCal [23] proposes a label-shift-robust objective; Head2Tail [30] instead targets shift arising from long-tailed class distributions. More recently, Moreo [16] proposed DMCAL, built directly on top of a quantification method to construct a corrected calibration map.

3 Related Works

The theoretical connection between calibration and quantification under dataset shift has recently been formalized. Tasche [27] shows that different dataset-shift assumptions lead to structurally distinct adjustment methods, some of which extend naturally from classification to quantification. Moreo et al. [16] further demonstrate that classifier calibration, quantification, and accuracy prediction are intrinsically related under dataset shift; and factors specific to each quantifier modulate how much calibration of the base classifier actually improves prevalence estimates.

In practice, this connection has mainly been exploited indirectly, by calibrating the base classifier before quantification. Probabilistic quantifiers such as PCC, PACC, and EMQ [24] depend directly on posterior probability estimates. A separate line of work corrects prevalence estimates directly, without recalibrating the base classifier, though without framing this correction as a calibration problem in its own right. The *Adjusted Classify and Count* (ACC) method [7] estimates the classifier’s true and false positive rates on held-out data and applies a linear adjustment to the raw CC estimate; PACC [3] applies the same adjustment to PCC using posterior-probability-based estimates of these rates. Viewed through the lens of calibration, both methods already correct the quantifier’s output

rather than the classifier’s scores, but they do so through a fixed, affine correction whose parameters are tied to a specific error-rate model. This restricts their applicability: the correction does not generalize to quantifiers whose bias is not well captured by an affine model, and it presupposes reliable estimates of the underlying error rates.

The idea of generalizing ACC and PACC’s linear correction through a learned regressor was first proposed by Moreo *et al.* [17] in the multi-label setting, where a regression function is trained to correct a quantifier’s prevalence estimates by exploiting correlations between class prevalences, without assuming the correction to be linear. This regression-based view is conceptually the closest antecedent to the present work, since it also learns a correction directly from the quantifier’s own output rather than from the classifier’s scores. The two approaches, however, address different problems and differ structurally rather than merely in scope. Moreo *et al.* use regression as a multi-label aggregation mechanism whose main purpose is to model dependencies among classes and improve their joint prevalence estimates; this mechanism requires at least two independent prevalence values to define a correlation, and is therefore only defined for three or more classes. In the binary case, the positive and negative prevalences are complementary by construction, so no such correlation exists to be exploited. The present work reinterprets this general idea from the perspective of calibration instead: it formalizes quantifier calibration as a distinct task, defines the condition a calibrated quantifier should satisfy, and introduces a generic post-hoc framework for learning correction functions over prevalence estimates.

Taken together, calibration at the quantifier level is theoretically motivated and, through ACC and PACC, already present in the literature, though only in a restricted parametric form; the closest attempt to generalize this correction targets multi-label class correlation rather than quantifier calibration itself. What remains open is a calibration method that imposes no parametric form and applies uniformly to any binary base quantifier. This is the gap the present work is designed to fill.

4 Method

This work proposes a post-hoc calibration method for binary quantifiers, called *QuantCal* (QCal), which systematically learns to correct prevalence estimates by exploiting the systematic relationship between the estimates produced by a quantifier and the true prevalence of validation samples with artificially controlled prevalences. By observing how the quantifier responds across the whole prevalence range, QCal characterizes the bias it leaves in its estimates and learns a monotone correction function that maps raw estimates to corrected ones directly at the prevalence level [5]¹, thereby producing more reliable estimates under label shift.

¹ We use the term *batch* (also referred to as *sample* in the quantification literature [5]) to denote a set of unlabeled instances for which the class prevalence is to be estimated.

Before describing the proposal, we formalize the notion of quantifier calibration and show that ACC and PACC already instantiate this framework through a restricted parametric correction.

4.1 Quantifier Calibration

Section 2.2 established that classifier calibration is an instance of Murphy’s general calibration condition (Equation (3)), applied to the classifier’s confidence scores as the predicted quantity and the correctness of the prediction as the outcome. We now extend this condition to quantifiers, where the predicted quantity is the prevalence estimate $\hat{p}_\oplus = q(a)_\oplus$ and the outcome is the true prevalence p of the positive class in the test sample a .

Definition 1 (Quantifier Calibration). *A quantifier q is said to be perfectly calibrated if, for every prevalence value $p \in [0, 1]$:*

$$\mathbb{E}[\hat{p}_\oplus \mid p] = p, \quad (5)$$

where $\hat{p}_\oplus = q(a)_\oplus$ is the prevalence estimate produced by q on a test sample a , and conditioning on p denotes that the true prevalence of the positive class in a is p . That is, among all test samples whose true positive-class prevalence is p , the expected prevalence estimate produced by q must equal p exactly.

Definition 1 is a direct instantiation of Equation (3): the predictor is the quantifier q , the prediction is $\hat{p}_\oplus$, and the outcome is the true prevalence p . The two equations express the same condition in different but equivalent forms. For binary outcomes, such as the occurrence of an event or the correctness of a classification, the calibration condition is naturally stated as a conditional probability, since $P(A \mid B) = \mathbb{E}[1[A] \mid B]$ for any event A . For quantifiers, the outcome, namely the true prevalence, is a continuous quantity in $[0, 1]$ rather than a binary event; thus, the condition is stated directly as a conditional expectation. In both cases, the requirement is the same: the predicted value must be, on average, exactly the value it purports to predict.

Remark 1 (Calibration and accuracy are independent notions). A quantifier may achieve low expected error while systematically violating Equation (5) at specific prevalence values, that is, it may be accurate on average yet miscalibrated across the prevalence range. Conversely, a quantifier may satisfy Equation (5) at every prevalence point yet exhibit high variance, yielding poor accuracy. Calibration and accuracy are therefore complementary properties, mirroring their well-known independence for classifiers.

Post-hoc quantifier calibration. By analogy with post-hoc classifier calibration (Equation (4)), a post-hoc quantifier calibration method learns a correction function $\phi : [0, 1] \rightarrow [0, 1]$ from held-out data, such that the calibrated quantifier q^{cal} satisfies:

$$\hat{p}_\oplus^{\text{cal}} = \phi(\hat{p}_\oplus), \quad (6)$$

with the goal that $\mathbb{E}[\phi(\hat{p}_{\oplus}) \mid p] = p$ for all $p \in [0, 1]$. The following result establishes that such a correction function exists for the CC quantifier, and that ACC and PACC already instantiate this general framework through a specific parametric choice of ϕ .

Proposition 1 (Existence of a perfect correction for CC). *Let h be a binary classifier with $tpr \neq fpr$, and let $\hat{p}^{CC}$ denote the CC estimator. Then there exists a correction function $\phi^* : [0, 1] \rightarrow [0, 1]$ such that the calibrated quantifier q^{cal} obtained by applying ϕ^* to $\hat{p}^{CC}$ is perfectly calibrated in the sense of Definition 1. Moreover, ϕ^* is strictly monotone.*

Proof. Under label shift, the expected CC estimate conditional on the true prevalence p is:

$$\mathbb{E}[\hat{p}^{CC} \mid p] = p \cdot tpr + (1 - p) \cdot fpr = fpr + p(tpr - fpr). \quad (7)$$

The assumption $tpr \neq fpr$ holds for any classifier that performs better than random, since $tpr = fpr$ would mean that the probability of predicting the positive class is the same regardless of the true class, and the classifier's predictions would carry no information about the true label. Under this assumption, Equation (7) is a strictly monotone affine function of p , hence invertible. Its inverse

$$\phi^*(z) = \frac{z - fpr}{tpr - fpr} \quad (8)$$

is strictly monotone and satisfies $\mathbb{E}[\phi^*(\hat{p}^{CC}) \mid p] = p$ for all $p \in [0, 1]$, which is precisely the condition of Definition 1.

Note that Equation (7) also shows that CC itself is not calibrated: its expected estimate equals p only when $tpr = 1$ and $fpr = 0$, that is, only for a perfect classifier.

Corollary 1 (ACC and PACC are parametric instances). *ACC and PACC are post-hoc quantifier calibration methods in the sense of Equation (6), with a parametric affine correction function.*

Proof. The correction applied by ACC [7] is:

$$\hat{p}^{ACC} = \frac{\hat{p}^{CC} - fpr_c}{tpr_c - fpr_c}, \quad (9)$$

where tpr_c and fpr_c are estimated from held-out data. This is exactly ϕ^* of Equation (8) with estimated parameters, applied to the output of CC. PACC [3] applies the same affine correction to the output of PCC, with parameters estimated from posterior probability scores rather than hard predictions. By Proposition 1, both yield perfectly calibrated quantifiers when their parameters are known exactly, which is the sense in which they are Fisher-consistent.

Corollary 1 characterizes ACC and PACC as instances of the general post-hoc quantifier calibration framework of Equation (6), in which ϕ is restricted to a parametric affine family and its two parameters are estimated from class-conditional error rates. A general-purpose post-hoc quantifier calibration method need not impose either restriction: it may instead learn ϕ directly from data, without assuming an affine (or any other fixed) parametric form and without requiring explicit estimation of tpr and fpr . Section 4.2 instantiates this broader framework by proposing QCal, a post-hoc quantifier calibration method that learns ϕ nonparametrically from data.

4.2 Proposal

Section 4.1 established that ACC and PACC instantiate the general post-hoc quantifier calibration framework of Equation (6) through a parametric affine correction function, estimated from class-conditional error rates. We now describe how QCal instantiates this same framework without either restriction, learning ϕ directly from data. The central challenge is learning the correction function ϕ without access to labeled test batches. QCal addresses this by exploiting two properties of the base quantifier that can be measured on the training data: the variability of its prevalence estimates across different prevalence conditions, and the systematic relationship between those estimates and the true prevalence of validation samples with artificially controlled prevalences. By collecting the quantifier’s behavior over validation batches with artificially controlled prevalences, QCal builds a characterization of how the quantifier errs, and fits a monotone correction function ϕ that maps raw estimates to corrected ones directly at the prevalence level. The complete architecture, illustrated in Fig. 1, proceeds in six steps.

- Step 1 - Dataset splitting:** In this step, given the training set D , we split it into two parts, D_{train} and D_{val} .
- Step 2 - Generation of validation batches:** From D_{val} , the Uniform Prevalence Protocol (UPP) is applied [5], producing a collection of batches. Unlike the Artificial Prevalence Protocol (APP), which sweeps a fixed grid of target prevalences, the UPP draws the target prevalence of each batch uniformly at random from the unit simplex by means of the Kraemer algorithm [5], thereby covering the prevalence range continuously rather than at a discrete set of points. Here, R denotes the number of batches sampled and *batch_size* the number of instances per batch.
- Step 3 - Quantifier training:** In this step, the quantifier, along with its underlying base classifier, is trained using the training data D_{train} .
- Step 4 - Validation (Repetition of Steps 1–3):** The three previous steps are executed n_{val} times, where $n_{\text{val}} \in \mathbb{N}^*$, each with a distinct stratified partition. Each repetition generates a new internal quantifier $q^{(k)}$ and produces a new collection of pairs $(\hat{p}_{j,\oplus}, p_{j,\oplus})$ from the batches generated by the UPP, where $\hat{p}_{j,\oplus}$ is the positive-class prevalence predicted by $q^{(k)}$ for the j -th batch and $p_{j,\oplus}$ is its corresponding true positive-class prevalence. Since the task is

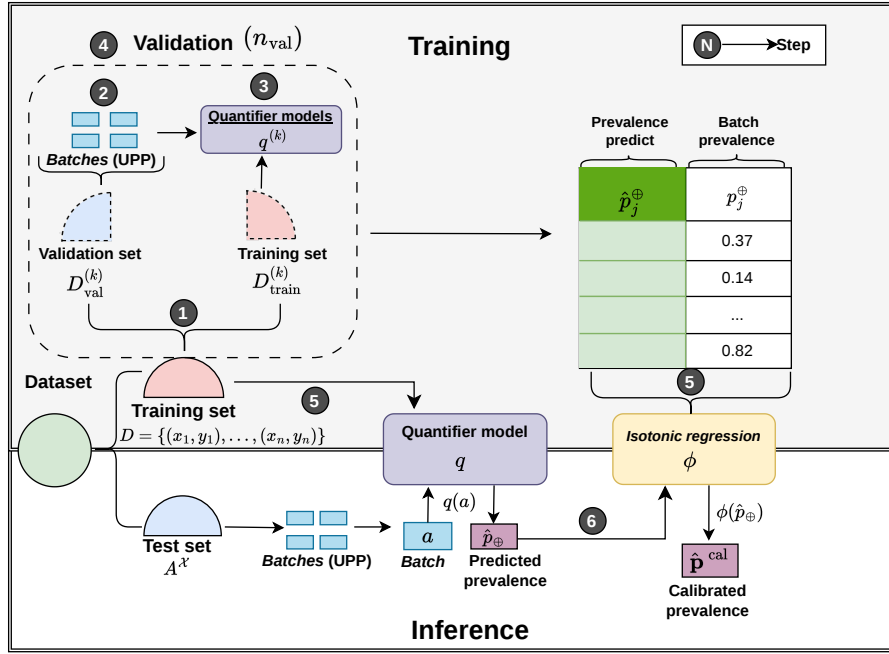


Fig. 1: Architecture of the method

binary, the negative-class component is fully determined by complement and is therefore redundant; only the positive-class component is retained. At the end of the n_{val} validation repetitions, all pairs across all repetitions and all batches are accumulated into a single collection $\mathcal{D}_{\phi}$, without any aggregation, yielding $n_{\text{val}} \cdot R$ pairs in total. These pairs depart systematically from the true prevalence across the whole prevalence range, as illustrated in Fig. B.1.

Step 5 - Fitting the correction function and the final quantifier: Using the pairs $(\hat{p}_{j,\oplus}, p_{j,\oplus}) \in \mathcal{D}_{\phi}$ accumulated across all n_{val} repetitions in Step 4, the correction function ϕ is fitted by isotonic regression, taking each raw estimate $\hat{p}_{j,\oplus}$ as input and the corresponding true prevalence $p_{j,\oplus}$ as target. The function is constrained to be non-decreasing and bounded to $[0, 1]$. Monotonicity encodes the natural requirement that a higher raw estimate should never be mapped to a lower corrected prevalence: it preserves the ordering induced by the base quantifier and acts as a strong regularizer, allowing ϕ to absorb systematic bias while remaining resistant to overfitting the calibration pairs. The $[0, 1]$ bound guarantees that every corrected value is a valid prevalence, so no external clipping is required. In parallel, the final base quantifier q is trained on the complete set D , maximizing the data available for the production model; this quantifier is entirely distinct from the internal quantifiers of Steps 3 and 4, which served only for calibration.

The trained method corresponds to the pair (q, ϕ) . The complete training procedure is summarized in Algorithm A.1.

Step 6 - Inference: At inference time, detailed in Algorithm A.2, given an unlabeled test sample $a \in \mathcal{A}^{\mathcal{X}}$, the base quantifier q is first applied to obtain an initial estimate, from which the positive-class component $\hat{p}_{\oplus} = (q(a))_{\oplus}$ is taken. The correction function ϕ is then applied to this scalar, and the negative-class prevalence is recovered by complement: $\hat{\mathbf{p}}^{\text{cal}} = (1 - \phi(\hat{p}_{\oplus}), \phi(\hat{p}_{\oplus}))$. By construction, $\hat{\mathbf{p}}^{\text{cal}}$ has non-negative components that sum to one, so the calibrated output is guaranteed to lie in the probability simplex.

Fig. B.1 illustrates the calibration pairs of Step 4 and the correction function ϕ fitted on them in Step 5, for two representative datasets with CC as the base quantifier. In both cases the point clouds depart systematically from the identity, confirming that CC is not calibrated in the sense of Definition 1, and the magnitude of the required correction differs markedly between them. On balance.3 the fitted curve lies only slightly above the identity, indicating a mild underestimation that QCal barely needs to correct. On ctg.3 the same base classifier leaves a pronounced systematic underestimation, and the fitted curve rises well above the identity across most of the prevalence range. Even between two datasets on which the classifier separates the classes well, the correction that restores calibration differs substantially in shape and steepness. This is what motivates learning ϕ nonparametrically rather than fixing a parametric form in advance: the affine correction of ACC and PACC (Corollary 1) can only rescale and shift the estimate, whereas isotonic regression recovers an arbitrary monotone relationship and adapts its shape to each dataset.

Fig. B.2 shows the effect of this learned correction on held-out test samples, confirming that it transfers beyond the calibration pairs. On balance.3, where CC is already close to calibrated and ϕ is nearly the identity, the corrected estimates are almost indistinguishable from the raw ones and the MAE barely moves ($0.0504 \rightarrow 0.0352$): QCal correctly leaves an already-calibrated quantifier essentially untouched. On ctg.3, the raw CC estimates lie systematically below the diagonal (underestimation), and ϕ lifts them onto it, cutting the MAE by more than a factor of three ($0.1416 \rightarrow 0.0501$). The contrast is exactly the behaviour a bias-correction method should exhibit: it intervenes in proportion to the systematic error present, correcting the biased ctg.3 strongly while leaving the already-calibrated balance.3 essentially unchanged. The residual scatter that remains on ctg.3 is expected, since QCal removes the systematic component of the error but not its variance. The systematic evaluation over all datasets is reported in Section 6.

5 Experiments

This section describes the experimental setup used to evaluate QCal empirically as a general-purpose calibration method. We first present the collection of binary

datasets used throughout the evaluation (Section 5.1), and then detail how each step of the architecture introduced in Section 4 was instantiated, including the eight base quantifiers to which QCal is applied independently (Section 5.2).

5.1 Datasets

We selected 106 binary classification datasets from the OpenML [4] and UCI Machine Learning [11] repositories. As selection criterion, we retained only datasets whose minority class contains at least 150 instances, ensuring a sufficient number of minority-class examples for reliable estimation of quantification performance. The resulting collection is intentionally diverse: the number of instances ranges from 625 to 245,057, the number of attributes from 2 to 970, and the minority class proportion from $\approx 4\%$ to 50%, covering balanced, mildly imbalanced, and severely imbalanced scenarios. All experiments were repeated 30 times to mitigate the effect of individual run variability.

5.2 Setup

The configuration of each step in the architecture proposed in Section 4 is as follows: in **Step 1**, we apply a stratified split allocating 70% of the instances to D_{train} and 30% to D_{val} . In **Step 2**, we apply UPP over D_{val} by means of the Kraemer algorithm [5], with *batch_size* = 100 instances and $R = 1100$ batches, whose target prevalences are drawn uniformly at random from the unit simplex. In **Step 3**, the eight base quantifiers are CC, ACC, PACC, EMQ, FM, DyS, and MS, from the *mlquantify* library² [14], and KDEy in its maximum-likelihood variant (KDEy-ML), from the QuaPy library³, in which the mixture parameter is estimated by maximum likelihood, which is equivalent to minimizing the Kullback-Leibler divergence between the test distribution and the mixture of the class-conditional densities [16]. Each of them is calibrated by QCal independently, so that the base quantifier and the internal quantifiers of Step 4 always belong to the same method. In **Step 4**, we set $n_{\text{val}} = 3$, yielding $n_{\text{val}} \cdot R = 3300$ prediction pairs in $\mathcal{D}_\phi$. In **Step 5**, ϕ is fitted by isotonic regression from *scikit-learn*⁴, under the constraints described in Section 4. **Step 6** requires no additional configuration.

All base quantifiers are aggregative and share the same base classifier, a *Logistic Regression* with default parameters. Since it is fitted by minimizing the log-loss, a strictly proper scoring rule, it already yields approximately calibrated posteriors on the training distribution; no additional probability calibration was thus applied, leaving QCal as the only source of correction in the pipeline. Each dataset of Section 5.1 is evaluated under 30 stratified train/test splits (70%/30%), which mitigates the effect of individual run variability; UPP is applied over the test portion with the same configuration of Step 2. Performance is measured by

² <https://luizfernandolj.github.io/mlquantify/>

³ <https://hlt-isti.github.io/QuaPy/index.html>

⁴ <https://scikit-learn.org/stable/index.html>

the Mean Absolute Error (MAE), averaged over the 1100 batches of each split and then over the 30 splits, yielding one score per method and per dataset. Each dataset contributes a single observation, and each pair $(q, \text{QCal}(q))$ is compared with the two-sided Wilcoxon signed-rank test at $\alpha = 0.05$, with p -values Holm-corrected for the eight comparisons. Reproducibility: the source code and datasets used in this work are publicly available⁵.

6 Results

We first assess whether QCal improves the base quantifier it is applied to. Table 1 reports, for each of the eight base quantifiers q , the MAE of q and of $\text{QCal}(q)$ averaged over the 106 datasets, together with the number of datasets on which QCal wins or loses, the average rank of both methods among the 16 competitors, and the p -value of a two-sided Wilcoxon signed-rank test over the per-dataset scores.

QCal reduces the mean MAE of every base quantifier, and all eight improvements remain significant at the 5% level after Holm correction. The magnitude of the gain, however, varies widely. It is largest for CC (48.9%), the only method that performs no correction of its own, and smallest for the quantifiers that already model the shift explicitly, such as FM (14.7%) and PACC (15.1%). The win/loss counts follow the same pattern: QCal wins on 102 of the 106 datasets when applied to CC, but the margin narrows to roughly 55–70 wins for the shift-aware quantifiers, and MS is the only case in which wins and losses are evenly split (53/53) despite a 21.0% reduction in mean MAE, an indication that, for this quantifier, QCal acts mainly on the datasets where the base method fails badly rather than uniformly across the benchmark. Taken together, these results are consistent with QCal acting as a bias-correction mechanism, as discussed in Section 4: the more residual bias the base quantifier leaves in its estimates, the more there is for QCal to remove.

The average ranks in Table 1 summarize each method with a single number and therefore hide how stable the improvement is across datasets. Fig. 2 shows the full distribution of the per-dataset ranks of the 16 methods, ordered from left to right by increasing average rank. Three observations follow.

First, the ordering is strictly *paired*: every $\text{QCal}(q)$ appears to the left of its own base quantifier q , without exception. The gain reported in Table 1 is thus not an artifact of a few datasets with large errors, it is visible in the bulk of the rank distribution.

Second, the top positions are held by $\text{QCal}(\text{DyS})$ (3.77), $\text{QCal}(\text{KDEy-ML})$ (3.83), and $\text{QCal}(\text{EMQ})$ (4.38), which interleave with and displace the uncorrected DyS (4.30), KDEy-ML (5.70), and EMQ (6.41). The violins of these three corrected variants are markedly concentrated in the first ranks, whereas their uncorrected counterparts spread over a wider range of positions: QCal not only lowers the expected rank of the strongest quantifiers but also makes their behaviour more consistent across datasets.

⁵ <https://github.com/isabellaloren4/Quantcal/>

Table 1: Effect of QCal on each base quantifier q over the 106 datasets. MAE: mean $\pm$ standard deviation across datasets; lower is better. Δ : relative reduction of the mean MAE obtained by QCal(q) over q . W/L: number of datasets on which QCal(q) achieves a lower / higher MAE than q (wins and losses of QCal; they sum to 106, as no exact ties occurred). Rank: average rank of the method among all 16 methods, lower is better. p-value: two-sided Wilcoxon signed-rank test over the 106 per-dataset MAE scores, Holm-corrected for the eight comparisons.

	MAE		Δ	W/L	Rank		
	q	QCal(q)			q	QCal(q)	p-value
CC	.1780 $\pm$.1543	.0910 $\pm$.0810	48.9%	102/4	15.27	11.72	6.5e-18
ACC	.1191 $\pm$.1361	.0913 $\pm$.0809	23.4%	62/44	13.10	12.18	8.8e-05
PACC	.0808 $\pm$.0972	.0686 $\pm$.0626	15.1%	57/49	8.93	8.41	2.3e-02
EMQ	.0820 $\pm$.0963	.0656 $\pm$.0644	20.0%	70/36	6.41	4.38	4.3e-06
FM	.0861 $\pm$.1004	.0735 $\pm$.0663	14.7%	58/48	10.34	9.95	4.1e-03
DyS	.0757 $\pm$.0991	.0637 $\pm$.0614	15.9%	59/47	4.30	3.77	2.3e-02
MS	.0962 $\pm$.1182	.0759 $\pm$.0682	21.0%	53/53	9.31	8.40	2.3e-02
KDEy-ML	.0849 $\pm$.1147	.0636 $\pm$.0629	25.2%	70/36	5.70	3.83	3.0e-07

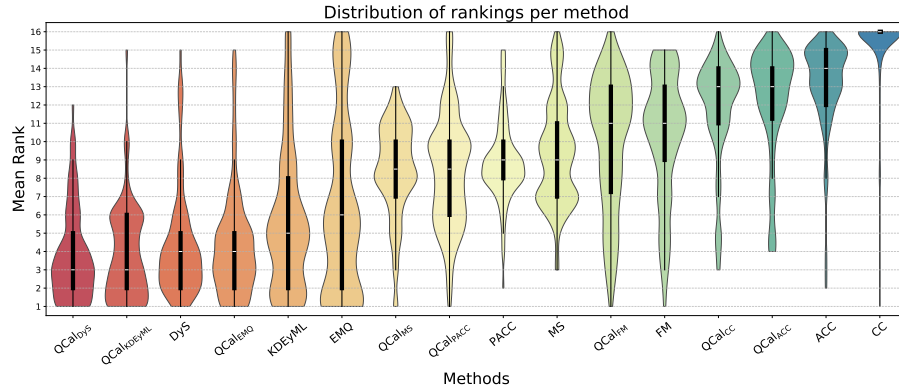


Fig. 2: Distribution of the ranks obtained by each of the 16 methods over the 106 datasets. For every dataset, the methods are ranked by MAE (rank 1 = best, rank 16 = worst); each violin shows the resulting distribution of ranks, and the inner box reports the median and the interquartile range. Methods are sorted from left to right by increasing average rank. QCal(q) denotes the base quantifier q corrected by QCal. Every corrected variant precedes its own base quantifier.

Third, the effect on the weakest methods is the most pronounced. CC has a rank distribution concentrated at the very bottom of the scale, as expected from a method that ignores the classifier’s bias entirely; once corrected, QCal(CC) moves to an average rank of 11.72, overtaking not only CC but also ACC (13.10), a

method explicitly designed to correct CC. The same holds for QCal(ACC) (12.18) with respect to ACC. In other words, a single post-hoc isotonic correction applied to the naive classify-and-count estimate is enough to outperform the classical adjusted count, which is precisely the behaviour one expects if the dominant error term of these quantifiers is a systematic, prevalence-dependent bias.

7 Conclusion

We extended calibration from classifiers to quantifiers, formalizing the condition a binary quantifier must satisfy to be calibrated and showing that ACC and PACC already instantiate it through a fixed parametric form. Building on this, we introduced QCal, which learns a monotone correction directly from a quantifier’s own prevalence estimates via isotonic regression, without assuming a parametric form or requiring a separate derivation for each base quantifier.

QCal significantly reduces the mean absolute error of all eight base quantifiers evaluated across 106 binary datasets. The gain is largest for CC, the only method with no correction of its own, and remains significant even for quantifiers that already model label shift explicitly. Notably, a single post-hoc correction applied to CC is enough to outperform ACC, indicating that the dominant error of these quantifiers is a systematic, prevalence-dependent bias that QCal is particularly well suited to remove.

Several directions remain open for future work: evaluating correction estimators beyond isotonic regression, assessing the effect of different batch sizes during calibration and testing, and extending QCal to the multiclass setting.

Acknowledgments. We thank the Graduate Program in Electrical and Computer Engineering (PGEEC) for the funding and support. We also thank the Computational Entomology Laboratory (COEN) for providing the mlquantify library.

References

1. Amr Alexandari, Anshul Kundaje, and Avanti Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *International conference on machine learning*, pages 222–232. PMLR, 2020.
2. Richard E. Barlow, David J. Bartholomew, James M. Bremner, and Hugh D. Brunk. *Statistical Inference under Order Restrictions: The Theory and Application of Isotonic Regression*. John Wiley & Sons, London and New York, 1972.
3. Antonio Bella, Cesar Ferri, Jose Hernandez-Orallo, and Maria Jose Ramirez-Quintana. Quantification via Probability Estimators. In *2010 IEEE International Conference on Data Mining*, pages 737–742, Sydney, Australia, December 2010. IEEE.
4. Bernd Bischl, Giuseppe Casalicchio, Taniya Das, Matthias Feurer, Sebastian Fischer, Pieter Gijsbers, Subhaditya Mukherjee, Andreas C Müller, László Németh, Luis Oala, et al. Openml: Insights from 10 years and more than a thousand papers. *Patterns*, 6(7), 2025.

5. Andrea Esuli, Alessandro Fabris, Alejandro Moreo, and Fabrizio Sebastiani. *Learning to Quantify*, volume 47 of *The Information Retrieval Series*. Springer International Publishing, 2023.
6. George Forman. Counting Positives Accurately Despite Inaccurate Classification. In *Machine Learning: ECML 2005*, volume 3720, pages 564–575. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
7. George Forman. Quantifying counts, costs, and trends accurately via machine learning. Technical report, Technical report, HP Laboratories, Palo Alto, CA, 2008.
8. Pablo González, Alberto Castaño, Nitesh V. Chawla, and Juan José Del Coz. A Review on Quantification Learning. *ACM Computing Surveys*, 50(5):74:1–74:40, September 2017.
9. Pablo González, Alejandro Moreo, and Fabrizio Sebastiani. Binary quantification and dataset shift: An experimental investigation. *Data Mining and Knowledge Discovery*, 38(4):1670–1712, 2024.
10. Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.
11. Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The UCI Machine Learning Repository. <https://archive.ics.uci.edu>, 2026. Last accessed: 2026/05/19.
12. Meelis Kull, Telmo M. Silva Filho, and Peter Flach. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 11(2):5052 – 5080, 2017.
13. Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32, 2019.
14. Luiz Fernando Luth. MLQuantify: A Python Package for Quantification, 2025.
15. André Maletzke, Denis Reis, Waqar Hassan, and Gustavo Batista. Accurately Quantifying under Score Variability. December 2021.
16. Alejandro Moreo. On the interconnections of calibration, quantification, and classifier accuracy prediction under dataset shift. *arXiv preprint arXiv:2505.11380*, 2025.
17. Alejandro Moreo, Manuel Francisco, and Fabrizio Sebastiani. Multi-label quantification. *ACM Transactions on Knowledge Discovery from Data*, 18(1):1–36, 2023.
18. Allan H Murphy. Scalar and vector partitions of the probability score: Part i. two-state situation. *Journal of Applied Meteorology and Climatology*, 11(2):273–282, 1972.
19. Allan H. Murphy and Robert L. Winkler. Reliability of subjective probability forecasts of precipitation and temperature. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 26(1):41–47, 1977.
20. Allan H. Murphy and Robert L. Winkler. A general framework for forecast verification. *Monthly Weather Review*, 115:1330–1338, 1987.
21. Sangdon Park, Osbert Bastani, James Weimer, and Insup Lee. Calibrated prediction with covariate shift via unsupervised domain adaptation. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 3219–3229. PMLR, 2020.
22. John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, volume 10, pages 61–74. MIT Press, 1999.

23. Teodora Popordanoska, Gorjan Radevski, Tinne Tuytelaars, and Matthew B. Blaschko. Lascal: Label-shift calibration without target labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.
24. Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002.
25. Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. A comparative evaluation of quantification methods. *Journal of Machine Learning Research*, 26(55):1–54, 2025.
26. Telmo Silva Filho, Hao Song, Miquel Perello-Nieto, Raul Santos-Rodriguez, Meelis Kull, and Peter Flach. Classifier calibration: a survey on how to assess and improve predicted class probabilities. *Machine Learning*, 112(9):3211–3260, 2023.
27. Dirk Tasche. Recalibrating binary probabilistic classifiers. In *Proceedings of the 5th International Workshop on Learning to Quantify, LQ 2025, ECML/PKDD Workshop*, pages 47–63, 2025.
28. Ximei Wang, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Transferable calibration with lower bias and variance in domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 19212–19223, 2020.
29. Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 694–699, 2002.
30. Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16489–16498, 2021.

A QCal Algorithm

A.1 Training

Algorithm A.1: QCal: Training

Input : D , labeled training set
Input : q , quantification algorithm
Input : $n_{\text{val}} \in \mathbb{N}^*$, R , *batch_size*, validation loop and UPP parameters
 /* Calibration pair extraction phase */
 1 $\mathcal{D}_\phi \leftarrow \emptyset$;
 2 **for** $k \leftarrow 1$ **to** n_{val} **do**
 3 $(D_{\text{train}}^{(k)}, D_{\text{val}}^{(k)}) \leftarrow \text{stratified-split}(D, 0.7)$;
 4 train $q^{(k)}$ on $D_{\text{train}}^{(k)}$ using algorithm q ;
 5 $\mathcal{S}^{(k)} \leftarrow \text{UPP}(D_{\text{val}}^{(k)}; R, \text{batch_size})$;
 6 **for each** $(S_j, p_{j,\oplus}) \in \mathcal{S}^{(k)}$ **do**
 7 $\hat{p}_{j,\oplus}^{(k)} \leftarrow (q^{(k)}(S_j))_{\oplus}$;
 8 $\mathcal{D}_\phi \leftarrow \mathcal{D}_\phi \cup \{(\hat{p}_{j,\oplus}^{(k)}, p_{j,\oplus})\}$;
 9 **end**
 10 **end**
 /* Isotonic regressor and final quantifier */
 11 $\phi \leftarrow \text{IsotonicRegression}(\mathcal{D}_\phi; y_{\min} = 0, y_{\max} = 1, \text{increasing})$;
 12 train q on D ;
Return: q, ϕ

A.2 Inference

Algorithm A.2: QCal: Inference

Input : $a \in \mathcal{A}^{\mathcal{X}}$, unlabeled test sample
Input : q , trained base quantifier
Input : ϕ , trained isotonic correction function
 1 $\hat{p}_{\oplus} \leftarrow (q(a))_{\oplus}$;
 2 $\hat{p}_{\oplus}^{\text{cal}} \leftarrow \phi(\hat{p}_{\oplus})$;
 3 $\hat{p}_{\ominus}^{\text{cal}} \leftarrow 1 - \hat{p}_{\oplus}^{\text{cal}}$;
Return: $\hat{\mathbf{p}}^{\text{cal}} = (\hat{p}_{\ominus}^{\text{cal}}, \hat{p}_{\oplus}^{\text{cal}})$

B Charts

B.1 Calibration

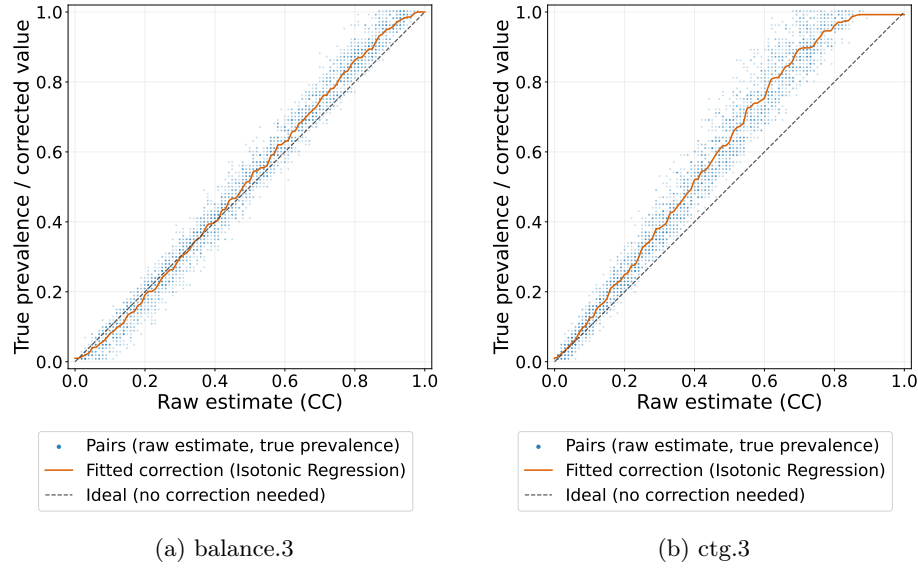


Fig. B.1: Calibration pairs and the resulting correction function ϕ on two representative datasets, with CC as the base quantifier. Each point is a pair $(\hat{p}_{j,\oplus}, p_{j,\oplus}) \in \mathcal{D}_\phi$ from a UPP validation batch over the n_{val} repetitions of Step 4: the horizontal axis is the raw estimate produced by CC, and the vertical axis the true prevalence of the batch. The solid line is the isotonic regression fitted on these pairs, i.e. the correction function ϕ of Step 5; the dashed line marks the identity, along which no correction would be needed. The pairs shown are those on which ϕ was fitted, so the figure is a diagnostic of the learned correction rather than evidence of generalization; the latter is assessed on held-out test samples in Fig. B.2 and Section 6.

B.2 Calibrated Predictions

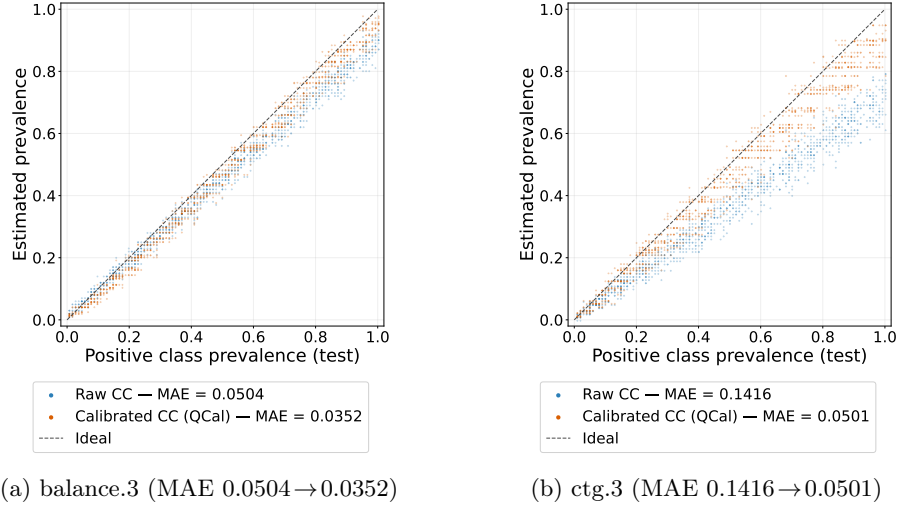


Fig. B.2: Effect of the correction function ϕ of Fig. B.1 on held-out test samples, for the same two datasets and CC as the base quantifier. Each panel plots the estimated prevalence against the true test prevalence: raw CC estimates (blue) versus QCal-corrected estimates (orange), each annotated with its mean absolute error (MAE); the dashed line marks the identity. Unlike Fig. B.1, these estimates come from test samples not seen during fitting, so the figure measures generalization of the learned correction. On both datasets the corrected estimates lie closer to the identity and the MAE decreases, with the larger gain on the more strongly biased ctg.3. A full evaluation across all datasets is reported in Section 6.