Estimating Model Performance Under Covariate Shift Without Labels

Jakub Białek[*]
NannyML NV
Belgium
jakub@nannyml.com

Juhani Kivimäki[*]
University of Helsinki
Finland
juhani.kivimaki@helsinki.fi

Wojtek Kuberski[*]
NannyML NV
Belgium
wojtek@nannyml.com

Nikolaos Perrakis[*]
NannyML NV
Belgium
nikos@nannyml.com

Abstract

After deployment, machine learning models often experience performance degradation due to shifts in data distribution. It is challenging to assess post-deployment performance accurately when labels are missing or delayed. Existing proxy methods, such as data drift detection, fail to measure the effects of these shifts adequately. To address this, we introduce a new method for evaluating binary classification models on unlabeled tabular data that accurately estimates model performance under covariate shift and call it Probabilistic Adaptive Performance Estimation (PAPE). It can be applied to any performance metric defined with elements of the confusion matrix. Crucially, PAPE operates independently of the original model, relying only on its predictions and probability estimates, and does not need any assumptions about the nature of covariate shift, learning directly from data instead. We tested PAPE using over 900 dataset-model combinations from the US census data, assessing its performance against several benchmarks through various metrics. Our findings show that PAPE outperforms other methodologies, making it a superior choice for estimating the performance of binary classification models.

1 Introduction

The final step in the machine learning (ML) model development process is to evaluate the model on an unseen dataset, known as *test set*. Successful performance on this dataset, indicating potential business value, typically leads to the model's deployment to *production*. The model's performance on the production data is expected to match the performance measured on the test set [1]. Yet, this assumption often fails due to data distribution shifts that lead to model performance degradation [2]. Therefore, ongoing monitoring of the model's performance is essential to ensure it continues to meet expected outcomes. However, measuring real-time performance is challenging in many scenarios because true labels are either unavailable or significantly delayed.

In scenarios where labels are unavailable, performance is commonly estimated using proxy methods that primarily monitor changes in the input data distribution [3]. However, such changes in input data often have only a minimal effect on the model's actual performance [4], and even when there is a notable effect, it might not be harmful [5]. Additionally, the methods used to measure changes in the input distribution typically provide estimates in units that do not correspond to performance metrics.

[*]These authors contributed equally to this work

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Thus, at best, input data monitoring can offer some qualitative understanding of the performance stability. When a distribution shift does occur, these methods do not provide any insight into the direction or magnitude of the impact on performance. The true extent of performance changes only becomes apparent once the labels become available, which may sometimes never happen. Until then, the model might incur significant financial losses [6].

In recent years, a new approach of *unsupervised accuracy estimation* [7] has produced several methods to estimate model performance directly without access to labels. A notable shortcoming of these methods has been that they focus solely on estimating the accuracy of a given classifier model. However, accuracy is often not the most appropriate metric for assessing model performance, which has recently motivated the shift of focus to *unsupervised performance estimation* [8], where estimators should be applicable to any classification metric, not just accuracy. In this paper, we extend this line of work and present Probabilistic Adaptive Performance Estimator (PAPE), a novel method for estimating any classification metric that can be defined using the elements of the *confusion matrix*. PAPE is provably robust against *covariate shift*. Furthermore, PAPE models the full distribution of the estimated metric, which can be leveraged to produce valid confidence intervals for the estimates.

With these estimates, the impact of changes in model performance on business operations can be quantified, allowing informed decision-making about corrective adaptations. These may include altering how model predictions are utilized in downstream processes, determining if model retraining is required, or whether some fail-safe mechanism should trigger.

The main contributions of this paper are the following:

- We introduce PAPE, a novel method for estimating model performance under covariate shift. PAPE can be applied to any classification metric that can be defined with elements of the confusion matrix. Importantly, PAPE autonomously learns from the data without any user input regarding the nature of the covariate shift.
- We provide theoretical guarantees for the estimation quality of PAPE and derive bounds for its estimation error under approximate calibration for composable metrics.
- We demonstrate PAPE's effectiveness empirically. Our experiments show that PAPE significantly outperforms all previous methods across all metrics evaluated.

2 Related Work

In recent years, there has been a surge of new suggested methods seeking to solve the unsupervised accuracy estimation problem. Importance Weighting (IW) [9] is a simple but powerful method used in model adaptation [10] that can also be used to estimate model performance. IW calculates a likelihood ratio of observing test set input data in production and uses this ratio to estimate performance for production data as a weighted metric calculated for the test set data. An extended version of IW [11] incorporates knowledge about the differences between test and production data distributions and their impact on performance, but requires the user to specify this information a priori.

A set of methods is based on training an ensemble of models and comparing their predictions [12; 7; 13]. Other approaches requiring additional training include Contrastive Automated Model Evaluation (CAME) [14], where the model training objective is augmented with a contrastive learning component, and reverse testing (RT) [15], which uses the monitored model predictions as labels to train a reverse model on production data. The reverse model's performance on the test dataset is then assumed to indicate the monitored model's performance on the production data. Whilst these approaches offer promising results, they cannot be applied off the shelf but require additions or alterations in training the model being estimated, and often come with a significant computational overhead.

Another set of methods measures distributional distances between the test and production data distributions [16; 17; 18; 19] and either tries to identify the change in performance directly based on this distance or by training a light-weight regressor model that is used to map the distance into a change in performance. The main challenge of these methods is in translating the measured distance into a meaningful performance estimate, either requiring the creation of multiple synthetic shifted data distributions to train the regressor [16] or resorting to quantifying the shift in performance in terms of some secondary metric such as correlation [18].

In this work, our main interest lies in methods that utilize the model's confidence in estimating its performance. Average Confidence (AC) [20] was originally suggested for Out-of-Distribution (OoD) detection but has since become the de facto baseline of confidence-based estimators [21]. Difference of Confidence (DoC) [16] uses the difference between confidence scores from test to production data and fits a regression model that maps this difference to accuracy. Average Thresholded Confidence (ATC) [22] learns a threshold on confidence scores and estimates accuracy as the fraction of predictions that exceed this threshold. Confidence Optimal Transport (COT) [19] measures prediction error as a Wasserstein distance between the test set's label distribution and the production set's pseudo-label distribution.

A key limitation of all of the above methods is that they are developed to estimate only classification accuracy, whereas other classification metrics are often more appropriate and descriptive of model performance. For example, in situations with severe class imbalance, a dummy model, which always predicts the majority class, can achieve deceptively high accuracy [23]. Recently, a new approach of Confidence-based Performance Estimation (CBPE) [8] has been proposed to address this shortcoming of previous estimators.

Since PAPE is essentially an extension of the CBPE method, we describe its key properties here (see also Appendix A). In CBPE, confidence scores produced by the model for a sample of predictions are used to estimate the elements of the confusion matrix. Each element is treated as a random variable following a Poisson binomial distribution whose full probability distribution is derived using the confidence scores as parameters. Using the expected values of the distributions as point estimates for the elements of the confusion matrix, these estimates have been shown to be unbiased and consistent under perfect calibration [8]. Having estimated the elements of the confusion matrix, one can take any classification metric that can be defined in terms of these elements and derive the distribution of the metric based on the distributions of the elements using an appropriate algorithm [8]. Again, the expected value of this distribution can then be used as a point estimate for that metric. Additionally, one can produce valid confidence intervals for these metrics from the derived probability distributions [8].

Although CBPE is a theoretically sound approach to solving the unsupervised performance estimation problem, its main downside is its reliance on model calibration, which is known to deteriorate under certain distributional shifts [24]. This phenomenon undermines the usability of CBPE and all other confidence-based estimators in a way that is currently not well understood. There have even been conflicting reports on whether calibration in confidence-based estimators is useful or not [16; 22]. In this work, we offer a remedy to this problem by augmenting the CBPE approach to make its calibration more robust against distributional shifts, resulting in PAPE, which we will describe in the following section.

3 Methodology

In this section, we present PAPE, a new algorithm for estimating model performance under covariate shift. We begin by describing the setting in Section 3.1. Then, in Section 3.2, we define α-approximate calibration along with some theoretical insights. Finally, In Section 3.3, we describe the PAPE method and give a bound for its estimation error for certain metrics.

3.1 Unsupervised Performance Estimation Under Covariate Shift

Suppose we have trained a probabilistic binary classifier, where $f : \mathcal{X} \to [0, 1]$ outputs a confidence score for the positive class and $g : [0, 1] \to \{0, 1\}$ maps these scores to binary predictions. The classifier is trained using data from some source distribution $\mathcal{D}_s = p_s(\boldsymbol{x}, y)$, where we have access to labels. We further assume that the image of f is a countable set $R(f)$, where each value $v \in R(f)$ defines a *levelset* $\{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) = v\}$. We seek to estimate our model's performance in a potentially shifted target distribution $\mathcal{D}_t = p_t(\boldsymbol{x}, y)$, where we have access only to samples from $p_t(\boldsymbol{x})$.

If no assumptions about the nature of the shift are made, the unsupervised performance estimation task is impossible [25; 19]. In this work, we make the most commonly used *covariate shift assumption* [26], where the shift from the source to target distribution can be attributed solely to changes in the marginal distribution of the covariates. That is, since any distribution can be factor-

ized as $p(\boldsymbol{x}, y) = p(y|\boldsymbol{x})p(\boldsymbol{x})$, we assume that the label-assigning conditional distribution remains unchanged $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x})$ while the marginal distribution of covariates shifts $p_s(\boldsymbol{x}) \neq p_t(\boldsymbol{x})$.

3.2 Approximate Confidence Calibration

As mentioned, a key shortcoming with confidence-based estimators has been the dependence on a small calibration error, which f is not guaranteed to achieve out-of-the-box. The simplest way to fix this is to use a small amount of validation data from the source distribution to train a post-hoc calibration mapping $c : [0, 1] \to [0, 1]$ to minimize the expected absolute calibration error $\mathbb{E}_{\mathcal{D}_s}\left[|P_{\mathcal{D}_s}(Y = 1 \mid c(f(X))) - c(f(X))|\right]$. Unfortunately, even if calibration error is eliminated in the source distribution, it typically remanifests in the target distribution under covariate shift [24].

Previous research has provided theoretical guarantees for confidence-based performance estimation in the ideal situation of perfect calibration [8]. In this work, we generalize this scope to approximate calibration. Recently, in AI-fairness research, researchers have borrowed ideas from the field of differential privacy [27], which has led to the notion of α-approximate calibration defined as follows:

Definition 3.1. *Assume f is a model $f : \mathcal{X} \to [0, 1]$ with a countable set of values $R(f)$, and that $\alpha \geq 0$. We say that f is α-approximately calibrated in $\mathcal{D}$, if:*

$$K(f, \mathcal{D}) = \sum_{v \in R(f)} P_{\mathcal{D}}(f(\boldsymbol{x}) = v)\left|\mathbb{E}_{\mathcal{D}}[y \mid f(\boldsymbol{x}) = v] - v\right| \leq \alpha. \tag{1}$$

Note that by setting $\alpha = 0$, we get the commonly used definition of perfect calibration [28] (marginalized over all possible values f can take) as a special case. Calibration in this sense is a marginal guarantee, since the left-hand side of the Inequality (1) is an average over the whole distribution. In fairness research, it has become apparent that such guarantees should also apply conditionally on some properties of the instance, such as group membership. Otherwise, a model might be well-calibrated overall, but yield a high bias for some (possibly protected) minority groups. This has led to the notion of *multicalibration*, which we will define in its general form.

Definition 3.2. *Assume f is a model $f : \mathcal{X} \to [0, 1]$ with a countable set of values $R(f)$, $\mathcal{H}$ is a collection of functions $h : \mathcal{X} \to \mathbb{R}$, and $\alpha \geq 0$. We say that f is α-approximately multicalibrated in $\mathcal{D}$ with respect to $\mathcal{H}$, if $\forall h \in \mathcal{H}$:*

$$K(f, \mathcal{D}, \mathcal{H}) = \sum_{v \in R(f)} P_{\mathcal{D}}(f(\boldsymbol{x}) = v)\left|\mathbb{E}_{\mathcal{D}}[h(\boldsymbol{x})(y - v) \mid f(\boldsymbol{x}) = v]\right| \leq \alpha. \tag{2}$$

Here, the functions h were originally indicator functions for group membership for groups of interest, but this was later generalized to any real-valued functions to allow for weighted memberships [29]. Recently, $\mathcal{H}$ was used as a hypothesis space for *density ratio estimation* (DRE) models, which are used to approximate the true likelihood ratios $w_{s\to t}(\boldsymbol{x}) = \frac{p_t(\boldsymbol{x})}{p_s(\boldsymbol{x})}$ under subpopulation shift [30]. For this setting, we have the following theorem:

Theorem 3.1. *Assume that $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x})$ and that f is α-approximately multicalibrated in $\mathcal{D}_s$ with respect to $\mathcal{H}$. If $w_{s\to t} \in \mathcal{H}$, then $K(f, \mathcal{D}_t) \leq \alpha$.*

In addition to providing a proof of this theorem in Appendix B.1, we will also give an upper bound for the calibration error in $\mathcal{D}_t$ when $w_{s\to t} \notin \mathcal{H}$ and some $h \in \mathcal{H}$ is used to approximate $w_{s\to t}$ instead (in Appendix B.3). Multicalibration in this setting is used to anticipate potential changes in the marginal distribution $p(\boldsymbol{x})$ a priori and to approximately adapt to all such changes simultaneously. Although algorithms for achieving multicalibration exist, they are computationally demanding [27]. This problem becomes increasingly difficult with the size of $\mathcal{H}$ [29], which can be infinite when considering all possible ways a distribution can shift. PAPE circumvents this problem by extracting data from an actual shifted distribution and fitting a calibration mapping to exactly that distribution, essentially becoming multicalibrated with respect to an $\mathcal{H}$ that has only one member. We will describe the details of this process next.

3.3 Probabilistic Adaptive Performance Estimation (PAPE)

In this section, we introduce PAPE and explain how it extends CBPE by addressing CBPE's main limitation: maintaining calibration under covariate shift. Let $\hat{Y} = g(f(X))$ denote the binary

prediction and $m : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a binary classification performance metric with some unknown expected value $m_{(f,g,\mathcal{D}_t)} = \mathbb{E}_{\mathcal{D}_t}[m(\hat{Y}, Y)]$ in distribution $\mathcal{D}_t$, where we don't have access to labels. However, we do have access to labels in some source distribution $\mathcal{D}_s$. Assume that f is already trained with data from $\mathcal{D}_s$. We start by collecting (i.i.d.) random samples $\{(X_i^s, Y_i^s)\}_{i=1}^{n_s} \sim p_s(\boldsymbol{x}, y)$ and $\{X_j^t\}_{j=1}^{n_t} \sim p_t(\boldsymbol{x})$, and training a DRE model from a hypothesis space of binary classifiers $\mathcal{H} \subseteq \{h \mid h : \mathcal{X} \to \{0, 1\}\}$ defined by the learning algorithm of user's choice as follows.

First, we assign indicator labels $z_i^s = 0$ to all X_i^s and $z_j^t = 1$ to all X_j^t. Next, we concatenate the features $X^{st} = [X^s; X^t] \in \mathbb{R}^{(n_s+n_t)\times d}$ (where d is the dimensionality of $\mathcal{X}$) and their corresponding indicator labels $z^{st} = [z^s; z^t] \in \mathbb{R}^{n_s+n_t}$. Then, we use (X^{st}, z^{st}) as training data for the DRE model that learns to discriminate between samples from the source and target distributions as described in [31, chapter 2.7.5]. Once the best-fit DRE model $h^* \in \mathcal{H}$ is found, we can use it to estimate the probabilities of observing instances $\boldsymbol{x}_i^s$ from the source distribution within the target distribution, formally $P(z_i = 1 \mid X_i^s = \boldsymbol{x}_i^s) \approx h^*(\boldsymbol{x}_i^s)$. Finally, we can approximate $w_{s\to t}$ with

$$\widehat{w}_{s\to t}(\boldsymbol{x}_i^s) = \frac{n_s}{n_t} \cdot \frac{h^*(\boldsymbol{x}_i^s)}{1 - h^*(\boldsymbol{x}_i^s)}, \tag{3}$$

and train a weighted calibration mapping c by fitting it to $\{(f(X_i^s), Y_i^s)\}_{i=1}^{n_s}$ using $\widehat{w}_{s\to t}(\boldsymbol{x}_i^s)$ as weights. Once the calibrated scores $c(f(X_j^t))$ are available, they can be used to derive performance estimates with CBPE without access to labels from the target distribution [8].

As a side note, one would typically choose a dedicated calibration mapping [32; 33; 34] for this purpose, as they enforce monotonicity, keeping the ranking of the scores and the resulting binary predictions unchanged. However, with PAPE, one can use any regression model of choice as the calibration mapping since the calibrated scores are used solely for performance estimation purposes and don't affect the binary predictions in any way.

If $c \circ f$ is perfectly calibrated in $\mathcal{D}_t$, the theoretical guarantees from CBPE [8] carry over, and PAPE estimates are guaranteed to be unbiased and consistent. However, since perfect calibration is unattainable in any real-life situation, let us next explore the relation between calibration error and PAPE estimation error in a limited setting: Some performance metrics, such as accuracy and precision, can be calculated as a mean of observation-level values. We call such metrics *composable*. For any composable metric m, the PAPE estimate can be written as

$$\widehat{m}_{(c\circ f,g,\mathcal{D}_t)} = \mathbb{E}_{p_t(\boldsymbol{x})}\left[c(f(\boldsymbol{x}))m\left(\hat{y}, 1\right) + \left(1 - c(f(\boldsymbol{x}))\right) m\left(\hat{y}, 0\right)\right]. \tag{4}$$

In practice, this expectation is approximated with the sample mean. For any composable metric, the estimation error of PAPE is bounded by the calibration error as described by the following theorem, which we will prove in Appendix B.2:

Theorem 3.2. *Let $c \circ f$ be α-calibrated in $\mathcal{D}_t$. Also, assume that f has a countable image set $R(f)$, m is a composable metric with $0 \leq m(\hat{y}, y) \leq 1$, and that $\hat{m}$ is its PAPE estimate. Then,*

$$|m_{(f,g,\mathcal{D}_t)} - \widehat{m}_{(c\circ f,g,\mathcal{D}_t)}| \leq \alpha.$$

For metrics that are not composable, such as the F_1 score, the above bound is not guaranteed to hold. However, our empirical experiments show that the PAPE estimates are superior to any other unsupervised performance estimation method for these metrics as well. We will present these findings next.

4 Experimental Evaluation

This section describes our experimental setting, where we evaluated and compared the proposed method against existing benchmarks. We describe the datasets we used in Section 4.1. The experimental setup, along with the ML models whose performance was estimated, is described in Section 4.2. We provide practical implementation details of the evaluated benchmarks in Section 4.3 (see also Appendix A). In Section 4.4 we propose novel evaluation metrics suitable for aggregating over multiple dissimilar evaluation cases and present the results of our experiments in terms of these metrics. We ran additional experiments with data from the recently published TableShift benchmark [35] and explored the effect of sample size on the estimators. These experiments are described in Appendix D.

4.1 Datasets

The datasets we used to evaluate the method come from Folktables [36]. Folktables uses US census data and preprocesses it to create a set of binary classification problems. We used the following tasks: ACSIncome, ACSPublicCoverage, ACSMobility, ACSEmployment, ACSTravelTime. For each of the five prediction tasks listed above, we fetched Folktables data for all 50 US states spanning four consecutive years (2015-2018). This gave us 250 datasets.

4.2 Experimental Setup

We started by separating the first-year data (2015) in each fetched dataset and used it as a training period. For each resulting training data set, we fitted five commonly used binary classification algorithms: Logistic Regression, Neural Network Model [37], Random Forest [38], XGBoost [39], and LGBM [40] with default parameters. We used these models to predict labels on the remaining part of the datasets. We recorded both binary predictions and confidence scores for the positive class. This resulted in 1,250 dataset-model pairs, which we call evaluation cases.

The rest of the data for each case was further divided into two periods - *reference* (the year 2016) and *production* (2017, 2018). Reference data was used to fit the performance estimation methods with model inputs, model outputs, and actual labels. Production data was further split into data chunks of 2,000 observations each, maintaining the order of the observations. The realized performance of the monitored model was recorded for each data chunk based on the monitored model's outputs and actual labels. The performance of the monitored model for the same chunks was then estimated based on the monitored model's inputs and outputs.

For each production data chunk, we compared the realized performance and the estimated performance for each performance estimation method. We filtered out evaluation cases with fewer than 6,000 observations (3 chunks) in the reference data set and cases where the performance of the monitored model on the reference data was worse than random (that is, with an AUROC lower than 0.5). After this filtering, we ended up with 959 evaluation cases, containing 36,557 production data chunks (evaluation points) for each evaluated method[2]. We measured the estimation performance of each method with three metrics: Accuracy, F_1 score, and AUROC. Figure 1 shows an example of an AUROC estimation result.

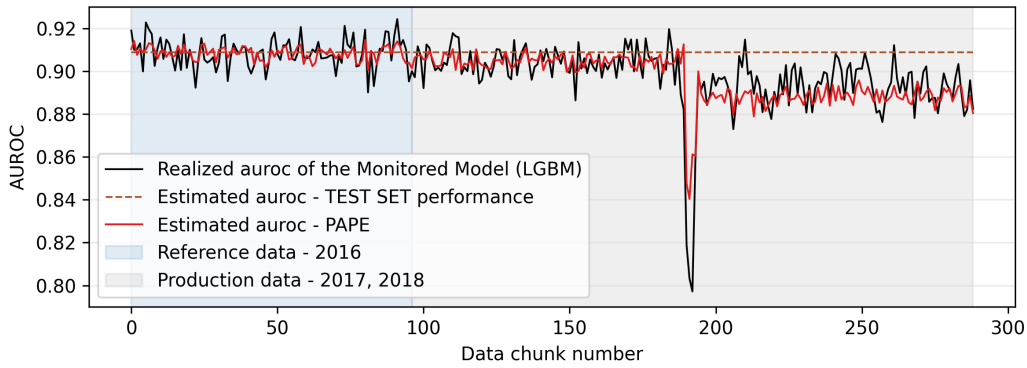

Figure 1: Estimation of AUROC for ACSIncome data (California) and LGBM as the monitored model. The black line is the realized AUROC of the monitored model for each data chunk. The red line is the AUROC estimated with PAPE. The brown dashed line is the TEST SET performance.

4.3 Benchmarks

Below, we describe the benchmarks that were evaluated against PAPE and their implementation details. We omit MANDOLINE [11] as it requires user input on the nature of the covariate shift.

[2]We used a single 11th Gen Intel i7-11800H 2.30GHz machine; computation took over 120 hours.

PAPE We implement PAPE as described in Section 3.3. We use an LGBM [40] Classifier as the DRE model, and an LGBM Regressor for the calibration mapping, both with default hyperparameters. We assume that the reference data originates from the source distribution $\mathcal{D}_s$ and that each production data chunk originates from some target distribution $\mathcal{D}_t$, which is potentially different for each chunk.

TEST SET performance For each evaluation data chunk from production data, the performance estimated by this benchmark equals the performance calculated on reference data (and typically, the test set is chosen as the reference set). It is our baseline benchmark, representing the initial assumption under which the model's performance on the production data is constant and equal to the performance calculated on the test set with reference data.

Confidence-based Performance Estimation (CBPE) CBPE [8] is a simpler version of PAPE. We train a calibration mapping as with PAPE, using reference data from $\mathcal{D}_s$, but we do not use likelihood ratios in training the mapping (it is equivalent to PAPE with $w_{s\to t}$ fixed to 1 for all observations). This results in a classifier that has a small calibration error with the reference data, but is not guaranteed to be well-calibrated with the production data from $\mathcal{D}_t$.

Average Threshold Confidence (ATC) With ATC [22], we take the raw scores provided by the monitored model $f(\boldsymbol{x})$ and apply the maximum confidence function to them, denoting it as MC:

$$\mathrm{MC}\left(f(\boldsymbol{x})\right) = \begin{cases} f(\boldsymbol{x}), & f(\boldsymbol{x}) \geq 0.5 \\ 1 - f(\boldsymbol{x}), & \text{otherwise} \end{cases} \tag{5}$$

Then we use it to learn a threshold on reference data such that the fraction of observations above the threshold is equal to the performance metric value calculated on reference data. When inferring, we apply MC to the evaluation data chunk and calculate the fraction of observations above the learned threshold. The estimated metric is equal to the calculated fraction.

Difference of Confidence (DoC) DoC [16] assumes that the performance change is proportional to the change in the mean of maximum confidence (5). To learn this relationship, we fit a Linear Regression model on the difference of confidence between the shifted and the reference data as input, and the difference in performance between these two as the target. In order to get datasets with synthetic distribution shifts, we perform multiple random resamplings of the reference dataset.

Confidence Optimal Transport (COT) In binary classification, COT [19] finds the optimal transport plan from the scores $f(X_i^t)$ with $\{X_i^t\} \sim p_t(\boldsymbol{x})^n$ to the labels y_j^s with $\{y_j^s\} \sim p_s(y)^n$, where n is the chunk size. The cost of this transportation plan is used as an estimate of the classification error of f in $\mathcal{D}_t$. We estimate only accuracy with this method (see Appendix A for details).

Modified Reverse Testing (RT-mod) Reverse Testing methods train a *reverse* model on production data with the monitored model inputs as features and monitored model predictions as targets. This reverse model is then used to make predictions on reference inputs, and its performance is evaluated with the reference labels. The reverse model's performance on reference data is a proxy for the monitored model's performance on the production data. We modify the method by additionally fitting Linear Regression to relate the reverse model performance change with the monitored model performance (just like in DoC).

Importance Weighting (IW) For IW, we first estimate density ratios $w_{s\to t}$ between reference and production data with the DRE classifier, the same as we use for PAPE. Then, we use them as weights to estimate the weighted performance metric of interest using reference data.

4.4 Evaluation

Performance estimation is a regression problem, which motivates the use of evaluation metrics such as the Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE). However, aggregating MAE/RMSE over multiple models and chunks of data from different datasets leads to skewed and uninterpretable results. Large MAE/RMSE for an evaluation case where the model's performance has a large variance might still be satisfactory. On the other hand, even small changes (in the absolute scale) in performance might be significant in cases where the model's performance is very stable.

To account for this scaling issue, we used normalized versions of MAE and RMSE, where we scaled absolute/squared errors by the standard error (SE) calculated for each evaluation case. We first measured SE as the standard deviation of the realized performance sampling distribution on the reference data. We did this by repeatedly sampling 2,000 observations (the size of the evaluation data chunk) at random from the reference data with replacement. Then, we calculated the realized performance metric on each sample. We repeated this 500 times and calculated the standard deviation of the obtained per-sample metrics - the standard error (SE). Then, for each evaluation case, we divided the MAE and RMSE by SE and squared SE, respectively, resulting in normalized mean absolute error (NMAE) and normalized root mean squared error (NRMSE), defined formally as

$$\text{NMAE} = \frac{1}{\sum_{i=1}^{k} c_i} \sum_{i=1}^{k} \sum_{j=1}^{c_i} \frac{|m_{i,j} - \widehat{m}_{i,j}|}{\text{SE}_i}$$

$$\text{NRMSE} = \sqrt{\frac{1}{\sum_{i=1}^{k} c_i} \sum_{i=1}^{k} \sum_{j=1}^{c_i} \frac{(m_{i,j} - \widehat{m}_{i,j})^2}{\text{SE}_i^2}},$$

where i is the evaluation case index ranging from 1 to k, j is the index of the production data chunk ranging from 1 to c_i (number of production chunks in a case i), and $m_{i,j}$ and $\widehat{m}_{i,j}$ are respectively realized and estimated performance for case i and chunk j. SE_i is the standard error of the i-th evaluation case. The results are shown in Table 1.

	Accuracy		AUROC		F_1	
	NMAE	NRMSE	NMAE	NRMSE	NMAE	NRMSE
TEST SET	1.62	2.88	1.45	2.30	2.53	8.23
RT-mod	2.31	5.41	1.85	4.29	2.06	4.74
COT	2.10	4.31	-	-	-	-
ATC	1.58	2.79	1.90	3.52	2.97	9.06
DOC	1.13	1.80	1.37	2.75	2.56	8.52
CBPE	1.08	1.75	1.07	1.68	1.03	2.12
IW	1.04	1.40	1.06	1.56	1.07	1.74
PAPE	**0.97**	**1.28**	**0.99**	**1.45**	**0.90**	**1.34**

Table 1: NMAE and NRMSE of the evaluated methods for each estimated metric.

PAPE, together with CBPE and IW, shows strong improvement over the TEST SET baseline. PAPE significantly outperforms the second-best method for each estimation and evaluation metric (paired Wilcoxon signed-rank test $p < 0.001$). PAPE result of NMAE equal to 0.97 for accuracy estimation means that the estimation is, on average, less than 1 SE away from the realized performance. For intuition, the NMAE of PAPE for the evaluation case shown in Figure 1 is equal to 1.11. The TEST SET performance NMAE for accuracy is 1.68, indicating that accuracy changes significantly in the production data chunks. If the performance was stable with only random normal fluctuations, the NMAE of the TEST SET performance would be around 0.8[3]. The changes for F_1 are much stronger, as shown by the increased NMAE of TEST SET - 2.49. This explains why performance estimation methods provide the strongest improvement compared to the TEST SET performance for this metric. Additionally, we calculated NMAE and NRMSE for each monitored model type - PAPE consistently outperformed IW for each of the monitored models and each estimation and evaluation metric.

We also analyzed each performance estimation algorithm's accuracy across different levels of realized performance changes. For relatively small realized performance changes (within $\pm$2SE), the TEST SET performance baseline is sufficiently accurate. Any useful performance estimator should outperform this baseline at least when the realized performance changes are significant (not within $\pm$2SE). We sorted the data by the magnitude of the performance change and calculated the rolling NMAE for 2SE-wide intervals. The results are shown in Figure 2.

[3]That is because the average absolute deviation for a normal distribution is $\sqrt{\frac{2}{\pi}} \approx 0.8$ of its standard deviation.

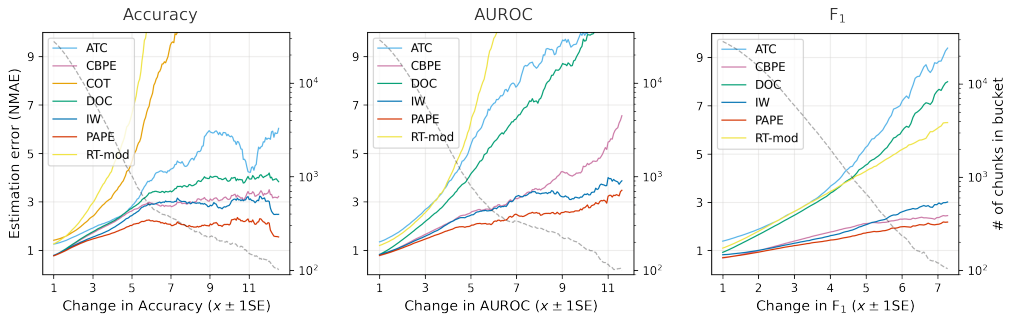

Figure 2: Estimation errors (NMAE) of estimated metric vs. realized absolute change as SE for all estimators. The x-axis indicates the center of the data bucket - for example, value 1 indicates a bucket that contains data chunks for which the absolute performance change was between 0 - 2 SE. The left y-axis shows NMAE of the evaluated method for the data bucket. The right y-axis shows the number of data chunks in each bucket on a logarithmic scale as depicted by the grey dashed line.

For all the estimated metrics, PAPE shows the best estimation quality in nearly the whole evaluated region. It gets significantly better than other methods when the change in realized performance is large. IW and CBPE show comparable performance, with IW being more accurate in most buckets. The RT and COT methods produce estimation errors, which render them unusable for all metrics used in the experiment. The DOC and ATC methods are somewhat competitive when estimating accuracy (for which they were designed) and when the change in realized accuracy is small enough. For the other metrics, the estimation errors are too high for practical monitoring purposes.

5 Discussion

PAPE is perhaps best understood as an improvement over the CBPE method [8]. PAPE retains all of the benefits of CBPE, such as applicability to multiple metrics instead of just accuracy, but also addresses the problem of deteriorating calibration under covariate shift, which is the major shortcoming of CBPE. Our experimental findings show that PAPE consistently outperforms CBPE in all experiments. Alternatively, PAPE can be viewed as an in-between solution, trying to achieve the best of both worlds of IW and multicalibration approaches.

Multicalibration requires post-processing, where the original model is iteratively updated through an auditing process [29]. This might result in predictions different from those of the original model. Also, the hypothesis space $\mathcal{H}$ needs to be large enough to contain $w_{s\to t}$, or at least good approximations of them, for any potential shifts. Unfortunately, especially in high-dimensional settings, this might be infeasible or at least increase the computational burden significantly. In contrast, PAPE focuses on one target distribution at a time, learning the density ratios required directly from the data. Thus, PAPE does not need to compromise performance by trying to deal with multiple distributions at the same time. PAPE is also non-invasive, making no alterations to the monitored model, which makes it more suitable for monitoring purposes. In fact, the model being monitored does not need to be calibrated at all, since PAPE provides calibration on the fly.

On the other hand, IW is known to suffer from high variance estimates when there is a significant discrepancy between the source and target distributions [41]. Since the labels used with IW are either 0 or 1, it is also susceptible to large sampling errors, especially in small sample settings that are typical in model monitoring. Since PAPE uses soft scores from the interval $[0, 1]$, it tends to smooth out these sampling effects. We provide a more comprehensive comparative analysis of the variances of PAPE and IW in Appendix C.

5.1 Limitations

PAPE will not work under *concept shift*, that is, if $p_s(y|\boldsymbol{x}) \neq p_t(y|\boldsymbol{x})$. Also, when operating in a covariate shift setting, the data may shift outside the support of the source distribution, which means

that there is a region $S \subseteq \mathcal{X}$, where for all $\boldsymbol{x} \in S$ we have $p_t(\boldsymbol{x}) > 0$ and $p_s(\boldsymbol{x}) = 0$. Weighted calibration for instances from such regions is not possible as the weights $w_{s \to t}(\boldsymbol{x})$ are not defined.

PAPE hinges on calibration performance within the target distribution $\mathcal{D}_t$, which in turn depends on good enough density ratio estimates. Both can be hard to achieve if not enough labeled reference data is available to train the DRE model and the calibration mapping. This data demand increases with the dimensionality of the covariates. As such, PAPE is most performant with low-dimensional data.

5.2 Future Work

Although we have described PAPE as a method for estimating the performance of binary classifiers, extending it to multiclass classifiers is straightforward. For instance, in the case of macro-averaged metrics - where performance is computed separately for each class in a one-vs-all manner and then averaged - PAPE can be applied directly by estimating the per-class performance and averaging the results. In fact, PAPE can be used to monitor the performance of any model that produces confidence scores in addition to its predictions. We leave the details of this for future work.

One challenge with PAPE or any other method relying on density-ratio estimation is that estimating these ratios becomes increasingly hard with high-dimensional data. Examining the sample complexity requirements and relating those to the estimation quality of PAPE is an interesting and important research question to be addressed by later research.

By using AUROC as an estimated metric in our experiments, we expanded the scope of CBPE to a previously unestimated metric. We explain in Appendix A how this was done precisely. Contrary to CBPE, where a full distribution for each metric is estimated, our approach with AUROC results only in an approximation of the expected value of the metric. We intend to examine metrics such as AUROC and AUPR further to provide a way to estimate the full distributions of these metrics, which require calculations over multiple thresholds.

6 Conclusion

We introduced PAPE, an innovative approach for accurately estimating the performance of binary classification models when faced with covariate shift. We examined its theoretical properties and provided a bound for its estimation error for composable metrics under approximate calibration. We performed rigorous testing for PAPE using real-world datasets drawn from US Census data, introducing novel evaluation metrics essential for aggregating results over multiple datasets and model monitoring scenarios. We analyzed over 900 model-dataset pairs and generated more than 36,000 data chunks for thorough evaluation. The results demonstrated that PAPE significantly outperforms existing methods across all metrics assessed.

Acknowledgments and Disclosure of Funding

This work was partly supported by local authorities ("Business Finland") under grant agreement 23004 ELFMo of the ITEA4 programme, which funded one of the authors. The remaining research was conducted at NannyML - a venture-backed open-source software company focused on post-deployment data science solutions - which has received public R&D funding from Flanders Innovation & Entrepreneurship (VLAIO) under project number HBC.2022.0846.

References

[1] Michael A. Lones. How to avoid machine learning pitfalls: a guide for academic researchers. *arXiv preprint arXiv:2108.02497*, 2021.

[2] Daniel Vela, Andrew Sharp, Richart Zhang, Trang Nguyen, An Hoang, and Oleg S. Pianykh. Temporal quality degradation in ai models. *Scientific Reports*, 12(11654), 2022.

[3] Janis Klaise, Arnaud Van Looveren, Clive Cox, Giovanni Vacanti, and Alexandru Coca. Monitoring and explainability of models in production. *arXiv preprint arXiv:2007.06299*, 2020.

[4] Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[5] Tom Ginsberg, Zhongyuan Liang, and Rahul G Krishnan. A learning based hypothesis test for harmful covariate shift. *arXiv preprint arXiv:2212.02742*, 2022.

[6] Samson Tan, Araz Taeihagh, and Kathy Baxter. The risks of machine learning systems. *arXiv preprint arXiv:2204.09852*, 2022.

[7] Jiefeng Chen, Frederick Liu, Besim Avci, Xi Wu, Yingyu Liang, and Somesh Jha. Detecting errors and estimating accuracy on unlabeled data with self-training ensembles. *Advances in Neural Information Processing Systems*, 34:14980–14992, 2021.

[8] Juhani Kivimäki, Jakub Białek, Wojtek Kuberski, and Jukka K. Nurminen. Performance estimation in binary classification using calibrated confidence. *arXiv preprint arXiv:2505.05295*, 2025.

[9] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[10] Nan Lu, Tianyi Zhang, Tongtong Fang, Takeshi Teshima, and Masashi Sugiyama. Rethinking importance weighting for transfer learning. In *Federated and Transfer Learning*, pages 185–231. Springer, 2022.

[11] Mayee Chen, Karan Goel, Nimit S Sohoni, Fait Poms, Kayvon Fatahalian, and Christopher Re. Mandoline: Model evaluation under distribution shift. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1617–1629. PMLR, 18–24 Jul 2021.

[12] Christina Baek, Yiding Jiang, Aditi Raghunathan, and J Zico Kolter. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. *Advances in Neural Information Processing Systems*, 35:19274–19289, 2022.

[13] Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. Assessing generalization of SGD via disagreement. In *International Conference on Learning Representations*, 2022.

[14] Ru Peng, Qiuyang Duan, Haobo Wang, Jiachen Ma, Yanbo Jiang, Yongjun Tu, Xiu Jiang, and Junbo Zhao. Came: Contrastive automated model evaluation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20121–20132, October 2023.

[15] Wei Fan and Ian Davidson. Reverse testing: An efficient framework to select amongst classifiers under sample selection bias. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 147–156, New York, NY, USA, 2006. Association for Computing Machinery.

[16] Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1134–1144, October 2021.

[17] Weijian Deng and Liang Zheng. Are labels always necessary for classifier accuracy evaluation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15069–15078, June 2021.

[18] Weijian Deng, Yumin Suh, Stephen Gould, and Liang Zheng. Confidence and dispersity speak: Characterising prediction matrix for unsupervised accuracy estimation. *arXiv preprint*, arXiv:2302.01094, 2023.

[19] Yuzhe Lu, Yilong Qin, Runtian Zhai, Andrew Shen, Ketong Chen, Zhenlin Wang, Soheil Kolouri, Simon Stepputtis, Joseph Campbell, and Katia Sycara. Characterizing out-of-distribution error via optimal transport. *Advances in Neural Information Processing Systems*, 36:17602–17622, 2023.

[20] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

[21] Juhani Kivimäki, Jakub Białek, Jukka K. Nurminen, and Wojtek Kuberski. Confidence-based estimators for predictive performance in model monitoring. *Journal of Artificial Intelligence Research*, 82:209–240, 2025.

[22] Saurabh Garg, Sivaraman Balakrishnan, Zachary Chase Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *International Conference on Learning Representations*, 2022.

[23] Mohamed Bekkar, Hassiba Kheliouane Djemaa, and Taklit Akrouf Alitouche. Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and Applications*, 3(10), 2013.

[24] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[25] Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 129–136. JMLR Workshop and Conference Proceedings, 2010.

[26] Jose G. Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

[27] Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.

[28] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1321–1330. JMLR.org, 2017.

[29] Ira Globus-Harris, Declan Harrison, Michael Kearns, Aaron Roth, and Jessica Sorrell. Multicalibration as boosting for regression. In *International Conference on Machine Learning*, pages 11459–11492. PMLR, 2023.

[30] Michael P. Kim, Christoph Kern, Shafi Goldwasser, Frauke Kreuter, and Omer Reingold. Universal adaptability: Target-independent inference that competes with propensity scoring. *Proceedings of the National Academy of Sciences*, 119(4), 01 2022.

[31] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023.

[32] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10:61–74, 03 1999.

[33] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 609–616, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[34] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 623–631. PMLR, 20–22 Apr 2017.

[35] Josh Gardner, Zoran Popovic, and Ludwig Schmidt. Benchmarking distribution shift in tabular data with tableshift. *Advances in Neural Information Processing Systems*, 36, 2024. Folktables library is MIT licensed.

[36] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021. Folktables library is MIT licensed.

[37] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *CoRR*, abs/2106.11959, 2021.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[39] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

[40] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154, 2017.

[41] Fengpei Li, Henry Lam, and Siddharth Prusty. Robust importance weighting for covariate shift. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 352–362. PMLR, 26–28 Aug 2020.

A Implementation Details

In this section, we first explain how F_1 and AUROC were estimated for PAPE (and CBPE) in the experiments in Section 4. Then, we describe our implementation for estimating the same metrics with ATC and DOC, since these methods were not originally intended for estimating any other metrics besides accuracy.

A.1 Estimating F_1 and AUROC with PAPE

We omitted confidence intervals in our experiments to relieve the computational burden and because only CBPE and PAPE are capable of providing these intervals. Thus, with CBPE and PAPE, we resorted to deriving only point estimates. Let us recall equation (4):

$$\widehat{m}_{(c\circ f,g,\mathcal{D}_t)} = \mathbb{E}_{p_t(\boldsymbol{x})}\left[c(f(\boldsymbol{x}))m\left(\hat{y},1\right) + \left(1-c(f(\boldsymbol{x}))\right)m\left(\hat{y},0\right)\right].$$

One can use the sample mean to approximate this expectation and use the approximation in turn to estimate the performance of *composable* metrics, such that can be calculated as a mean of observation-level values, like accuracy. For other metrics, we start by estimating elements of the confusion matrix. Assume we have a sample of n instances and that there are n_+ positive and n_- negative predictions in the sample. Then, as explained in [8], the elements of the confusion matrix can be estimated with:

$$\widehat{TP} = \frac{1}{n_+}\sum_{i=1}^{n} c\big(f(\boldsymbol{x_i})\big)\cdot\hat{y}_i$$

$$\widehat{FP} = \frac{1}{n_+}\sum_{i=1}^{n} \Big(1-c\big(f(\boldsymbol{x_i})\big)\Big)\cdot\hat{y}_i$$

$$\widehat{TN} = \frac{1}{n_-}\sum_{i=1}^{n} c\big(f(\boldsymbol{x_i})\big)\cdot(1-\hat{y}_i)$$

$$\widehat{TN} = \frac{1}{n_-}\sum_{i=1}^{n} \Big(1-c\big(f(\boldsymbol{x_i})\big)\Big)\cdot(1-\hat{y}_i)$$

With the estimated elements of the confusion matrix, one can estimate F_1 with:

$$\widehat{\mathrm{F}_1} \approx \frac{2\cdot\widehat{TP}}{\widehat{TP}+\widehat{FN}+n_+} \tag{6}$$

If we treat the value of F_1 as a random variable X_{F_1}, the approximation given in (6) converges to $\mathbb{E}[X_{F_1}]$ rapidly when n grows. This expectation, in turn, is an unbiased and consistent estimator of F_1 under perfect calibration. [8]

To estimate AUROC, the model's binary predictions $\hat{y}$ are not needed. We use the scores $f(x)$ returned by the classifier to define a new thresholding function h_v:

$$h_v\big(f(\boldsymbol{x}),v\big) = \begin{cases} 1, & f(x)\geq v \\ 0, & f(x)<v \end{cases}$$

For each $v \in R(f)$ and for each $\boldsymbol{x}_i$ in the sample, we can use $\hat{y}_i = h_v(f(\boldsymbol{x}_i),v)$ and estimate the elements of the confusion matrix as explained above. Then, using these estimates, we can approximate true positive rates (TPR) and false positive rates (FPR) as

$$\widehat{TPR}_v \approx \frac{\widehat{TP}_v}{\widehat{TP}_v+\widehat{FN}_v}$$

$$\widehat{FPR}_v \approx \frac{\widehat{TN}_v}{\widehat{TN}_v+\widehat{FP}_v}$$

Finally, we can calculate an estimate for AUROC with the approximated TPR and FPR. In our experiments, we tried several different calibration mappings and settled on LGBM, since it gave the best overall performance.

A.2 Estimating F_1 and AUROC with Other Estimators

Since COT [19] is essentially estimating the 0/1 classification error, there is no clear way of how it could be used to estimate metrics other than accuracy. For this reason, we left it out of experiments where F_1 and AUROC were estimated. Although ATC [22] and DOC [16] were also originally designed for estimating only accuracy, we used the following procedures to expand them for estimating other metrics as well.

When estimating the F_1 score and AUROC with ATC, we applied the same logic as with accuracy. For accuracy, ATC finds a confidence threshold such that the fraction of instances from the reference data set with predicted confidence score above the threshold matches model accuracy in the reference set. With production data, the fraction of confidence scores above the same confidence threshold is taken as the estimated accuracy for the production data. When estimating the F_1 and AUROC, we similarly found confidence thresholds matching these metrics with reference data and used the fraction of production data instances with confidence scores above those thresholds as estimates.

With DOC, we simulated distribution shifts by resampling reference data, as we did with accuracy. We collected the F_1 and AUROCs and fitted a linear regression model with the difference of average confidence between the reference and simulated sets as the single feature and the metric of interest as the target (just as with accuracy). We did not apply calibration to the confidence scores for either the DOC or ATC methods. Our reasoning behind this is that with DOC, the linear regression model is indifferent with respect to any rescaling of the confidence scores. On the other hand, with ATC, we found that the reported differences between the calibrated and uncalibrated versions of ATC with binary classification in [21] using the models we included in our experiments were so minimal that we chose to include only the variant that had a better reported overall performance, in this case, the uncalibrated version.

B Proofs

In this section, we provide proofs for the theorems presented in the main paper. We begin with Theorem 3.1.

B.1 Proof of Theorem 3.1

To improve readability, we start by proving two lemmas, which we can then leverage in the main proof[4]. First, we make use of the following connection between the expectations of the source $\mathcal{D}_s = p_s(\boldsymbol{x}, y)$ and target $\mathcal{D}_t = p_t(\boldsymbol{x}, y)$ distributions:

Lemma B.1. *Assume $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x})$ and fix any $S \subseteq \mathcal{X}$. Then, for any integrable function $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, we have:*

$$P_{\mathcal{D}_s}(\boldsymbol{x} \in S)\mathbb{E}_{\mathcal{D}_s}[w_{s\to t}(\boldsymbol{x}) \cdot F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S] = P_{\mathcal{D}_t}(\boldsymbol{x} \in S)\mathbb{E}_{\mathcal{D}_t}[F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S].$$

Proof.

$$\begin{aligned}
&P_{\mathcal{D}_s}(\boldsymbol{x} \in S)\mathbb{E}_{\mathcal{D}_s}[w_{s\to t} \cdot F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S] \\
=&\int_{\boldsymbol{x}\in S} p_s(\boldsymbol{x}) \cdot w_{s\to t}(\boldsymbol{x}) \cdot \mathbb{E}_{p(y|\boldsymbol{x})}[F(\boldsymbol{x}, y)]\, d\boldsymbol{x} \\
=&\int_{\boldsymbol{x}\in S} p_s(\boldsymbol{x}) \cdot \frac{p_t(\boldsymbol{x})}{p_s(\boldsymbol{x})} \cdot \mathbb{E}_{p(y|\boldsymbol{x})}[F(\boldsymbol{x}, y)]\, d\boldsymbol{x} \\
=&\int_{\boldsymbol{x}\in S} p_t(\boldsymbol{x}) \cdot \mathbb{E}_{p(y|\boldsymbol{x})}[F(\boldsymbol{x}, y)]\, d\boldsymbol{x} \\
=&P_{\mathcal{D}_t}(\boldsymbol{x} \in S)\mathbb{E}_{\mathcal{D}_t}[F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S].
\end{aligned}$$

□

Now, we can leverage Lemma B.1 to prove the following:

[4]Acknowledgement: Our derivations follow closely those of Prof. Aaron Roth on his "Learning with Conditional Guarantees" course, which he teaches at the University of Pennsylvania.

Lemma B.2. *Assume $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x})$. Suppose f is α-approximately multicalibrated in $\mathcal{D}_s$ with respect to $\mathcal{H}$. Then, f is also α-approximately multicalibrated in $\mathcal{D}_t$ with respect to $\mathcal{H}_{s\to t}$, where*

$$\mathcal{H}_{s\to t} = \left\{ \frac{h(\boldsymbol{x})}{w_{s\to t}(\boldsymbol{x})} : h(\boldsymbol{x}) \in \mathcal{H} \right\}.$$

Proof. Since f is α-approximately multicalibrated in $\mathcal{D}_s$ with respect to $\mathcal{H}$, for every $h \in \mathcal{H}$, we have:

$$\begin{aligned}
\alpha &\geq K(f, \mathcal{D}_s, h) \\
&= \sum_{v \in R(f)} P_{\mathcal{D}_s}(f(\boldsymbol{x}) = v) \Big| \mathbb{E}_{\mathcal{D}}[h(\boldsymbol{x})(y - v) \mid f(\boldsymbol{x}) = v] \Big| \\
&= \sum_{v \in R(f)} \Big| P_{\mathcal{D}_s}(f(\boldsymbol{x}) = v) \cdot \mathbb{E}_{\mathcal{D}_s}[h(\boldsymbol{x})(y - v) \mid f(\boldsymbol{x}) = v] \Big| \\
&= \sum_{v \in R(f)} \left| P_{\mathcal{D}_s}(f(\boldsymbol{x}) = v) \cdot \mathbb{E}_{\mathcal{D}_s} \left[w_{s\to t}(\boldsymbol{x}) \cdot \frac{h(\boldsymbol{x})}{w_{s\to t}(\boldsymbol{x})} \cdot (y - v) \;\middle|\; f(\boldsymbol{x}) = v \right] \right| \\
&= \sum_{v \in R(f)} \left| P_{\mathcal{D}_t}(f(\boldsymbol{x}) = v) \cdot \mathbb{E}_{\mathcal{D}_t} \left[\frac{h(\boldsymbol{x})}{w_{s\to t}(\boldsymbol{x})} (y - v) \;\middle|\; f(\boldsymbol{x}) = v \right] \right| \\
&= K\left(f, \mathcal{D}_t, \frac{h}{w_{s\to t}} \right),
\end{aligned}$$

where Lemma B.1 is applied to each of the terms

$$P_{\mathcal{D}_s}(f(\boldsymbol{x}) = v) \cdot \mathbb{E}_{\mathcal{D}_s} \left[w_{s\to t}(\boldsymbol{x}) \cdot \frac{h(\boldsymbol{x})}{w_{s\to t}(\boldsymbol{x})} \cdot (y - v) \;\middle|\; f(\boldsymbol{x}) = v \right]$$

using $S = \{\boldsymbol{x} : f(\boldsymbol{x}) = v\}$ and $F(\boldsymbol{x}, y) = \frac{h(\boldsymbol{x})}{w_{s\to t}(\boldsymbol{x})}(y - v)$. □

We are now ready to prove Theorem 3.1, which we will restate here:

Theorem 3.1. *Assume that $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x})$ and that f is α-approximately multicalibrated in $\mathcal{D}_s$ with respect to $\mathcal{H}$. If $w_{s\to t} \in \mathcal{H}$, then $K(f, \mathcal{D}_t) \leq \alpha$.*

Proof. Since we assumed that $w_{s\to t} \in \mathcal{H}$, we can choose $h = w_{s\to t}$ and apply Lemma B.2 to get

$$\begin{aligned}
\alpha &\geq K\left(f, \mathcal{D}_t, \frac{h}{w_{s\to t}} \right) \\
&= \sum_{v \in R(f)} \left| P_{\mathcal{D}_t}(f(\boldsymbol{x}) = v) \cdot \mathbb{E}_{\mathcal{D}_t} \left[\frac{h(\boldsymbol{x})}{w_{s\to t}(\boldsymbol{x})} (y - v) \;\middle|\; f(\boldsymbol{x}) = v \right] \right| \\
&= \sum_{v \in R(f)} P_{\mathcal{D}_t}(f(\boldsymbol{x}) = v) \Big| \mathbb{E}_{\mathcal{D}_t} \left[(y - v) \mid f(\boldsymbol{x}) = v \right] \Big| \\
&= K(f, \mathcal{D}_t).
\end{aligned}$$

□

B.2 Proof of Theorem 3.2

In this section, we will offer the proof of Theorem 3.2, which we restate below:

Theorem 3.2. *Let $c \circ f$ be α-calibrated in $\mathcal{D}_t$. Also, assume that f has a countable image set $R(f)$, m is a composable metric with $0 \leq m(\hat{y}, y) \leq 1$, and that $\hat{m}$ is its PAPE estimate. Then,*

$$|m_{(f,g,\mathcal{D}_t)} - \widehat{m}_{(c\circ f,g,\mathcal{D}_t)}| \leq \alpha.$$

Proof. We start by decomposing the calibration error as a sum of terms related to each levelset $\{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) = v\}$. For each v, we assume that a calibration error α_v remains after applying the calibration mapping c. Formally,

$$\mathbb{E}_{\mathcal{D}_t}[y \mid f(\boldsymbol{x}) = v] = P_{\mathcal{D}_t}[y = 1 \mid f(\boldsymbol{x}) = v] = c(v) + \alpha_v.$$

For any fixed v and deterministic c, clearly $\mathbb{E}_{\mathcal{D}_t}[c(v) \mid f(\boldsymbol{x}) = v] = c(v)$, which yields us

$$\begin{aligned}\mathbb{E}_{\mathcal{D}_t}[y \mid f(\boldsymbol{x}) = v] &= c(v) + \alpha_v \\ \mathbb{E}_{\mathcal{D}_t}[y \mid f(\boldsymbol{x}) = v] - c(v) &= \alpha_v \\ \mathbb{E}_{\mathcal{D}_t}[y - c(v) \mid f(\boldsymbol{x}) = v] &= \alpha_v.\end{aligned}$$

The total calibration error is then:

$$\sum_{v \in R(f)} P_{p_t(\boldsymbol{x})}(f(\boldsymbol{x}) = v)\left|\mathbb{E}_{\mathcal{D}_t}[y - c(v) | f(\boldsymbol{x}) = v]\right| = \sum_{v \in R(f)} P_{p_t(\boldsymbol{x})}(f(\boldsymbol{x}) = v)\,|\alpha_v| \leq \alpha.$$

Furthermore, notice that since the levelsets $\{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) = v\}$ form a partition of $\mathcal{X}$, for any integrable function $F : [0, 1] \to \mathbb{R}$ we have $\mathbb{E}_{p_t(\boldsymbol{x})}[F(v)] = \sum_{v \in R(f)} P_{p_t(\boldsymbol{x})}(f(\boldsymbol{x}) = v)F(v)$. Thus, we can write:

$$\begin{aligned}
m_{(f,g,\mathcal{D}_t)} &= \mathbb{E}_{\mathcal{D}_t}\left[m\left(g(f(\boldsymbol{x})), y\right)\right] \\
&= \sum_{v \in R(f)} P_{p_t(\boldsymbol{x})}\big(f(\boldsymbol{x}) = v\big)\mathbb{E}_{\mathcal{D}_t}\big[m\big(g(f(\boldsymbol{x})), y\big) \mid f(\boldsymbol{x}) = v\big] \\
&= \sum_{v \in R(f)} P_{p_t(\boldsymbol{x})}\big(f(\boldsymbol{x}) = v\big)\Big(m\big(g(v), 1\big) \cdot P_{\mathcal{D}_t}\big(y = 1 \mid f(\boldsymbol{x}) = v\big) + \\
&\qquad\qquad m\big(g(v), 0\big) \cdot P_{\mathcal{D}_t}\big(y = 0 \mid f(\boldsymbol{x}) = v\big)\Big) \\
&= \sum_{v \in R(f)} P_{p_t(\boldsymbol{x})}\big(f(\boldsymbol{x}) = v\big)\Big(m\big(g(v), 1\big)\big(c(v) + \alpha_v\big) + m\big(g(v), 0\big)\big(1 - (c(v) + \alpha_v)\big)\Big) \\
&= \sum_{v \in R(f)} P_{p_t(\boldsymbol{x})}\big(f(\boldsymbol{x}) = v\big)\Big(m\big(g(v), 1\big)c(v) + m\big(g(v), 0\big)\big(1 - c(v)\big)\Big) + \\
&\quad \sum_{v \in R(f)} P_{p_t(\boldsymbol{x})}\big(f(\boldsymbol{x}) = v\big)\Big(\alpha_v\big(m(g(v), 1) - m(g(v), 0)\big)\Big) \\
&= \widehat{m}_{(f,g,\mathcal{D}_t)} + \sum_{v \in R(f)} P_{p_t(\boldsymbol{x})}\big(f(\boldsymbol{x}) = v\big)\Big(\alpha_v\big(m(g(v), 1) - m(g(v), 0)\big)\Big),
\end{aligned}$$

which we can further manipulate as

$$m_{(f,g,\mathcal{D}_t)} - \widehat{m}_{(f,g,\mathcal{D}_t)} = \sum_{v \in R(f)} P_{p_t(\boldsymbol{x})}\big(f(\boldsymbol{x}) = v\big)\Big(\alpha_v\big(m(g(v), 1) - m(g(v), 0)\big)\Big). \tag{7}$$

For any metric m with $0 \leq m(\hat{y}, y) \leq 1$, we have

$$\begin{aligned}|m(g(v), 1) - m(g(v), 0)| &\leq 1 \\ |\alpha_v(m(g(v), 1) - m(g(v), 0))| &\leq |\alpha_v|,\end{aligned}$$

Thus, if we take the absolute values of both sides of Equation 7, we get

$$\begin{aligned}
|m_{(f,g,\mathcal{D}_t)} - \widehat{m}_{(f,g,\mathcal{D}_t)}| &= \left|\sum_{v \in R(f)} P_{p_t(\boldsymbol{x})}\big(f(\boldsymbol{x}) = v\big)\Big(\alpha_v\big(m(g(v), 1) - m(g(v), 0)\big)\Big)\right| \\
&= \sum_{v \in R(f)} P_{p_t(\boldsymbol{x})}\big(f(\boldsymbol{x}) = v\big) \cdot \big|\alpha_v\big(m(g(v), 1) - m(g(v), 0)\big)\big| \\
&\leq \sum_{v \in R(f)} P_{p_t(\boldsymbol{x})}\big(f(\boldsymbol{x}) = v\big)|\alpha_v| \\
&\leq \alpha,
\end{aligned}$$

completing the proof. □

B.3 Estimation Error Under Imperfect Weights

Our proof of Theorem 3.1 assumes that we have access to exact density ratios. Here, we will produce an upper bound for the calibration error when we resort to approximating the weights with some function $h \in \mathcal{H}$. To quantify the approximation error, we make use of the following definition:

Definition B.1. *Assume that $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x})$. For a function $h : \mathcal{X} \to \mathbb{R}$, we write*

$$\epsilon(h, w_{s\to t}) = \mathbb{E}_{p_s(\boldsymbol{x})}[|h(\boldsymbol{x}) - w_{s\to t}(\boldsymbol{x})|].$$

Similarly, for any subset $S \subseteq \mathcal{X}$, we write:

$$\epsilon(h, w_{s\to t}, S) = \mathbb{E}_{p_s(\boldsymbol{x})}[|h(\boldsymbol{x}) - w_{s\to t}(\boldsymbol{x})| \mid \boldsymbol{x} \in S].$$

Now, we can prove the following lemma:

Lemma B.3. *Assume $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x})$ and fix any $S \subseteq \mathcal{X}$. Then, for any integrable functions $F : \mathcal{X} \times \mathcal{Y} \to [-1, 1]$ and $h : \mathcal{X} \to \mathbb{R}$, we have:*

$$\begin{aligned}
&\left|P_{\mathcal{D}_s}(\boldsymbol{x} \in S) \cdot \mathbb{E}_{\mathcal{D}_s}[h(\boldsymbol{x}) \cdot F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S]\right| \\
&\geq \left|P_{\mathcal{D}_t}(\boldsymbol{x} \in S) \cdot \mathbb{E}_{\mathcal{D}_t}[F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S]\right| - P_{\mathcal{D}_s}(\boldsymbol{x} \in S) \cdot \epsilon(h, w_{s\to t}, S).
\end{aligned}$$

Proof. We can use Lemma B.1 to write

$$\begin{aligned}
&\left|P_{\mathcal{D}_s}(\boldsymbol{x} \in S) \cdot \mathbb{E}_{\mathcal{D}_s}[h(\boldsymbol{x}) \cdot F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S] - P_{\mathcal{D}_t}(\boldsymbol{x} \in S) \cdot \mathbb{E}_{\mathcal{D}_t}[F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S]\right| \\
&= \left|P_{\mathcal{D}_s}(\boldsymbol{x} \in S)\mathbb{E}_{\mathcal{D}_s}[h(\boldsymbol{x}) \cdot F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S] - P_{\mathcal{D}_s}(\boldsymbol{x} \in S)\mathbb{E}_{\mathcal{D}_s}[w_{s\to t}(\boldsymbol{x})F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S]\right| \\
&= P_{\mathcal{D}_s}(\boldsymbol{x} \in S) \cdot \left|\mathbb{E}_{\mathcal{D}_s}[(h(\boldsymbol{x}) - w_{s\to t}(\boldsymbol{x})) \cdot F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S]\right| \\
&\leq P_{\mathcal{D}_s}(\boldsymbol{x} \in S) \cdot \max_{(\boldsymbol{x},y)\in(\mathcal{X},\mathcal{Y})} |F(\boldsymbol{x}, y)| \cdot \mathbb{E}_{p_s(\boldsymbol{x}}[|h(\boldsymbol{x}) - w_{s\to t}(\boldsymbol{x})| \mid \boldsymbol{x} \in S] \\
&\leq P_{\mathcal{D}_s}(\boldsymbol{x} \in S) \cdot \epsilon(h, w_{s\to t}, S),
\end{aligned}$$

since clearly $\max_{(\boldsymbol{x},y)\in(\mathcal{X},\mathcal{Y})} |F(\boldsymbol{x}, y)| \leq 1$. Then, we can reverse the direction and use the reverse triangle inequality to write

$$\begin{aligned}
&P_{\mathcal{D}_s}(\boldsymbol{x} \in S) \cdot \epsilon(h, w_{s\to t}, S) \\
&\geq \left|P_{\mathcal{D}_s}(\boldsymbol{x} \in S) \cdot \mathbb{E}_{\mathcal{D}_s}[h(\boldsymbol{x}) \cdot F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S] - P_{\mathcal{D}_t}(\boldsymbol{x} \in S) \cdot \mathbb{E}_{\mathcal{D}_t}[F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S]\right| \\
&\geq \left|P_{\mathcal{D}_t}(\boldsymbol{x} \in S) \cdot \mathbb{E}_{\mathcal{D}_t}[F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S]\right| - \left|P_{\mathcal{D}_s}(\boldsymbol{x} \in S) \cdot \mathbb{E}_{\mathcal{D}_s}[h(\boldsymbol{x}) \cdot F(\boldsymbol{x}, y) \mid \boldsymbol{x} \in S]\right|,
\end{aligned}$$

from which the statement follows by simply rearranging the terms. □

This lemma can be used to prove the following theorem, which is a relaxed version of Theorem 3.1 and gives an upper bound for the calibration error with approximate likelihood ratios:

Theorem B.4. *Assume that $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x})$ and that f is α-approximately multicalibrated in $\mathcal{D}_s$ with respect to $\mathcal{H}$. Then,*

$$K(f, \mathcal{D}_t) \leq \alpha + \min_{h\in\mathcal{H}} \epsilon(h, w_{s\to t}).$$

Proof. Let $h^* = \arg\min_{h\in\mathcal{H}} \epsilon(h, w_{s\to t})$ and $S_v = \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) = v\}$ so that the collection $\{S_v\}_{v\in R(f)}$ forms a partition of $\mathcal{X}$. Thus, by the law of total probability,

$$\sum_{v\in R(f)} P_{\mathcal{D}_s}(f(\boldsymbol{x}) = v) \cdot \epsilon(h^*, w_{s\to t}, S_v) = \epsilon(h^*, w_{s\to t}).$$

Since f is α-approximately calibrated in $\mathcal{D}_s$ with respect to $\mathcal{H}$, we have:

$$\begin{aligned}
\alpha &\geq K(f, \mathcal{D}_t, h^*) \\
&= \sum_{v \in R(f)} \left| P_{\mathcal{D}_s}(f(\boldsymbol{x}) = v) \cdot \mathbb{E}_{\mathcal{D}_s}\left[h^*(\boldsymbol{x})(y - v) \;\middle|\; f(\boldsymbol{x}) = v \right] \right| \\
&\geq \sum_{v \in R(f)} \left| P_{\mathcal{D}_t}(f(\boldsymbol{x}) = v) \cdot \mathbb{E}_{\mathcal{D}_t}\left[y - v \;\middle|\; f(\boldsymbol{x}) = v \right] \right| - \\
&\quad \sum_{v \in R(f)} P_{\mathcal{D}_s}(f(\boldsymbol{x}) = v) \cdot \epsilon(h^*, w_{s \to t}, S_v\}) \\
&= K(f, \mathcal{D}_t) - \epsilon(h^*, w_{s \to t}).
\end{aligned}$$

Here, Lemma B.3 is applied to each of the terms

$$\left| P_{\mathcal{D}_s}(f(\boldsymbol{x}) = v) \cdot \mathbb{E}_{\mathcal{D}_s}\left[h^*(\boldsymbol{x})(y - v) \;\middle|\; f(\boldsymbol{x}) = v \right] \right|,$$

with $F(\boldsymbol{x}, y) = y - v$. □

C IW and PAPE Variance Comparison

In this section, we compare the variances of IW and PAPE when used to estimate the accuracy of a model that is trained with data from some source distribution $p_s(\boldsymbol{x}, y)$ in some target distribution $p_t(\boldsymbol{x}, y)$. We operate under the covariate shift assumption of $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x})$. We assume that the scores $S = f(X)$ produced by the classifier are perfectly calibrated in $p_s(\boldsymbol{x}, y)$ (but not necessarily in $p_t(\boldsymbol{x}, y)$). That is, if $(X, Y) \sim p_s(\boldsymbol{x}, y)$, then $P(Y = 1 \mid S = s) = s \quad \forall s \in [0, 1]$. In addition, we assume that there is some discriminator function $g : [0, 1] \to \{0, 1\}$ mapping the scores to binary predictions $\hat{Y} = g(S)$.

C.1 Estimators

We start by describing the estimators in this setting and the notation used.

C.1.1 Probabilistic Adaptive Performance Estimation

The CBPE [8] estimator for accuracy from a sample $(X_1, X_2, ..., X_n) \sim p_t(\boldsymbol{x})^n$ is defined as

$$X_{accuracy} = \frac{X_{correct}}{n}, \tag{8}$$

where $X_{correct}$ follows a Poisson binomial distribution with parameters Z_i defined as

$$Z_i = \begin{cases} S_i, & \hat{Y}_i = 1 \\ 1 - S_i, & \hat{Y}_i = 0. \end{cases} \tag{9}$$

Let $s : \mathcal{X} \to [0, 1]$ denote the mapping $s(X_i) = Z_i$. Under perfect calibration, using $\mathbb{E}_{p_t}[X_{accuracy}]$ as a point estimate yields an unbiased and consistent estimator for accuracy [8].

$$\widehat{Acc}_{\text{CBPE}} = \frac{1}{n} \sum_{i=1}^{n} Z_i. \tag{10}$$

However, the assumption of perfect calibration in the source distribution $p_s(\boldsymbol{x}, y)$ does not extend to the target distribution $p_t(\boldsymbol{x}, y)$, which typically results in a biased estimator in any target distribution. PAPE allows us to train a weighted calibrator $c : [0, 1] \to [0, 1]$ using the same (exact) density ratios $w(\boldsymbol{x})$ as IW (explained in the next section), ensuring that $f^w = c \circ f$ is perfectly calibrated in $p_t(\boldsymbol{x}, y)$. Now we can define

$$\widehat{Acc}_{\text{PAPE}} = \frac{1}{n} \sum_{i=1}^{n} Z_i^w, \tag{11}$$

where the difference to $\widehat{Acc}_{\text{CBPE}}$ is that the scores S_i^w used to define Z_i^w originate from f^w instead of f. Similarly, we let $s^w : \mathcal{X} \to [0, 1]$ denote the mapping $s^w(X_i) = Z_i^w$. Under the assumption of exact density ratios $w(\boldsymbol{x})$, $\widehat{Acc}_{\text{PAPE}}$ is unbiased and consistent in the target distribution.

C.1.2 Importance Weighting

The empirical importance-weighted (IW) estimator from a sample $(X_1, X_2, ..., X_n) \sim p_t(\boldsymbol{x})^n$ is defined as

$$\widehat{Acc}_{\text{IW}} = \frac{1}{n} \sum_{i=1}^{n} I_i \, w(\boldsymbol{x}_i), \tag{12}$$

where the indicator I_i is defined as

$$I_i = \begin{cases} 1, & \hat{Y}_i = Y_i \\ 0, & \hat{Y}_i \neq Y_i, \end{cases} \tag{13}$$

with $\hat{Y}_i = g(f(X_i))$, $\{Y_i\}_{i=1}^n \sim p_s(y \mid \boldsymbol{x})^n$, and for each $X_i = \boldsymbol{x}_i$

$$w(\boldsymbol{x}_i) = \frac{p_t(\boldsymbol{x}_i)}{p_s(\boldsymbol{x}_i)}, \tag{14}$$

is the density ratio relating the marginal source and target distributions, $p_s(\boldsymbol{x})$ and $p_t(\boldsymbol{x})$ respectively (with $p_s(\boldsymbol{x}) > 0$ whenever $p_t(\boldsymbol{x}) > 0$). Under the assumption of perfect calibration (in the source distribution), we have $I_i \sim \text{Bernoulli}(Z_i)$. The empirical importance-weighted estimator (under exact density ratios) is known to be unbiased and consistent in the target distribution.

C.2 Variance Comparison

If we have access to exact density ratios, both PAPE and IW are unbiased and consistent estimators for accuracy. Then, the choice between the two should depend only on their sample efficiencies. That is, which estimator has the smallest variance? On the other hand, if density ratios are not exact, it is a reasonable assumption (by the principle of insufficient reason) that both estimators would suffer roughly equally on average.

By using p_t as a shorthand for $X \sim p_t(\boldsymbol{x})$, the per-observation variance of PAPE is, by definition[5]

$$\text{Var}_{p_t}(Z^w) = \mathbb{E}_{p_t}[(Z^w)^2] - \mathbb{E}_{p_t}[Z^w]^2 = \mathbb{E}_{p_t}[s^w(\boldsymbol{x})^2] - \mathbb{E}_{p_t}[s^w(\boldsymbol{x})]^2. \tag{15}$$

Let us take a look at the per-observation terms of the IW estimator in a similar fashion. We start by denoting $W = Iw(X)$, where recall that I is a Bernoulli variable with parameter $Z = s(X)$. Conditioning on a fixed $X = \boldsymbol{x}$, we have $I \sim \text{Bernoulli}(s(\boldsymbol{x}))$ and the weight $w(\boldsymbol{x})$ is a constant. Using p_s as a shorthand for $Y \sim p_s(y \mid \boldsymbol{x})$, the conditional mean and variance of W are

$$\mathbb{E}_{p_s}[W \mid \boldsymbol{x}] = \mathbb{E}_{p_s}[Iw(\boldsymbol{x}) \mid \boldsymbol{x}] = w(\boldsymbol{x})\mathbb{E}_{p_s}[I \mid \boldsymbol{x}] = w(\boldsymbol{x})s(\boldsymbol{x}) \tag{16}$$

$$\text{Var}_{p_s}(W \mid \boldsymbol{x}) = \text{Var}_{p_s}(Iw(\boldsymbol{x}) \mid \boldsymbol{x}) = w(\boldsymbol{x})^2 \, \text{Var}_{p_s}(I \mid \boldsymbol{x}) = w(\boldsymbol{x})^2 s(\boldsymbol{x})(1 - s(\boldsymbol{x})). \tag{17}$$

Next, we can use the law of total variance[6] to get

$$\begin{aligned}
\text{Var}_{p_t}(W) &= \mathbb{E}_{p_t}\left[\text{Var}_{p_s}(W \mid \boldsymbol{x})\right] + \text{Var}_{p_t}\left(\mathbb{E}_{p_s}[W \mid \boldsymbol{x}]\right) \\
&= \mathbb{E}_{p_t}[w(\boldsymbol{x})^2 s(\boldsymbol{x})(1 - s(\boldsymbol{x}))] + \text{Var}_{p_t}(w(\boldsymbol{x})s(\boldsymbol{x})) \\
&= \mathbb{E}_{p_t}[w(\boldsymbol{x})^2 s(\boldsymbol{x})(1 - s(\boldsymbol{x}))] + \mathbb{E}_{p_t}\left[(w(\boldsymbol{x})s(\boldsymbol{x}))^2\right] - \mathbb{E}_{p_t}\left[w(\boldsymbol{x})s(\boldsymbol{x})\right]^2 \\
&= \mathbb{E}_{p_t}[w(\boldsymbol{x})^2 (s(\boldsymbol{x}) - s(\boldsymbol{x})^2) + w(\boldsymbol{x})^2 s(\boldsymbol{x})^2] - \mathbb{E}_{p_t}\left[w(\boldsymbol{x})s(\boldsymbol{x})\right]^2 \\
&= \mathbb{E}_{p_t}[w(\boldsymbol{x})^2 s(\boldsymbol{x}) - w(\boldsymbol{x})^2 s(\boldsymbol{x})^2 + w(\boldsymbol{x})^2 s(\boldsymbol{x})^2] - \mathbb{E}_{p_t}\left[w(\boldsymbol{x})s(\boldsymbol{x})\right]^2 \\
&= \mathbb{E}_{p_t}[w(\boldsymbol{x})^2 s(\boldsymbol{x})] - \mathbb{E}_{p_t}\left[w(\boldsymbol{x})s(\boldsymbol{x})\right]^2.
\end{aligned}$$

Now, we can compare the derived variances and state that the per-observation variance of the PAPE estimator is less than or equal to that of the IW estimator if and only if

$$\mathbb{E}_{p_t}[s^w(\boldsymbol{x})^2] - \mathbb{E}_{p_t}[s^w(\boldsymbol{x})]^2 \leq \mathbb{E}_{p_t}[w(\boldsymbol{x})^2 s(\boldsymbol{x})] - \mathbb{E}_{p_t}\left[w(\boldsymbol{x})s(\boldsymbol{x})\right]^2 \tag{18}$$

[5] In expectations such as $\mathbb{E}_{p_t}[s^w(\boldsymbol{x})]$, we use $\boldsymbol{x}$ as a dummy variable representing a specific realization of X. Thus, any reference to $\boldsymbol{x}$ in these expressions is to be understood as a placeholder that is integrated out.

[6] Here we again revert to slightly abused notation for compactness. In fact, we should write the total variance as $\text{Var}_{X \sim p_t(\boldsymbol{x}), Y \sim p_s(y|\boldsymbol{x})}(W)$. However, since we consider the source distribution to be fixed, we omit p_s from the left-hand side for readability.

Given that the value of the expression on the right-hand side depends heavily on the interplay between $w(\boldsymbol{x})$ and $s(\boldsymbol{x})$, and we don't know the exact relation between $s^w(\boldsymbol{x})$ and $s(\boldsymbol{x})$, it is impossible to verify whether Inequality (18) is true or not a priori. Here, we look at only one interesting special case, that being $p_s(\boldsymbol{x}) = p_t(\boldsymbol{x})$, which leads to a constant density ratio $w(\boldsymbol{x}) = 1$. In this setting, the calibrator c is the identity mapping so that $f = f^w$, which means that $s(\boldsymbol{x}) = s^w(\boldsymbol{x})$ pointwise. With these insights, criterion (18) can be written as

$$\mathbb{E}_{p_t}[s(\boldsymbol{x})^2] - \mathbb{E}_{p_t}[s(\boldsymbol{x})]^2 \leq \mathbb{E}_{p_t}[s(\boldsymbol{x})] - \mathbb{E}_{p_t}\left[s(\boldsymbol{x})\right]^2 \tag{19}$$

$$\mathbb{E}_{p_t}[s(\boldsymbol{x})^2] \leq \mathbb{E}_{p_t}[s(\boldsymbol{x})]. \tag{20}$$

Because $0 \leq s(\boldsymbol{x}) \leq 1$, it follows that

$$s(\boldsymbol{x})^2 \leq s(\boldsymbol{x})$$

$$\mathbb{E}_{p_t}[s(\boldsymbol{x})^2] \leq \mathbb{E}_{p_t}[s(\boldsymbol{x})],$$

which means that the special case criterion (20) is always satisfied. Thus, under no shift, the per-observation variance of PAPE is always at most that of the IW estimator, making the former more sample efficient. It also gives reason to believe that this is likely the case when the shift is small, and hence the weights $w(\boldsymbol{x})$ are close to one and $s^w(\boldsymbol{x}) \approx s(\boldsymbol{x})$ pointwise. Having said that, it is possible to envision situations where the IW estimator has lower per-observation variance.

Up to this point, we have looked only at the per-observation variances. However, it is straightforward to justify this approach as follows. We can write the variances of our estimators as variances of the means of n mutually independent observations as

$$\mathrm{Var}(\widehat{ACC}_{PAPE}) = \mathrm{Var}_{p_t}\left(\frac{1}{n}\sum_{i=1}^{n} Z_i^w\right) = \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{Var}_{p_t}(Z_i^w) = \frac{1}{n}\,\mathrm{Var}_{p_t}\left(Z^w\right)$$

$$\mathrm{Var}(\widehat{ACC}_{IW}) = \mathrm{Var}_{p_t}\left(\frac{1}{n}\sum_{i=1}^{n} W_i\right) = \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{Var}_{p_t}(W_i) = \frac{1}{n}\,\mathrm{Var}_{p_t}(W),$$

which means that in comparing the variances of the two estimators, it suffices to compare the per-observational variances $\mathrm{Var}_{p_t}(Z^w)$ and $\mathrm{Var}_{p_t}(W)$.

Finally, we suggest a pragmatic empirical approximation for criterion (18), which can be used as a heuristic guide in choosing which estimator to use for a given (i.i.d.) sample $(X_1, X_2, ..., X_n) \sim p_t(\boldsymbol{x})^n$. This approximation replaces the expectations with empirical means, leading to

$$\sum_{i=1}^{n} s^w(X_i)^2 - \left(\sum_{i=1}^{n} s^w(X_i)\right)^2 \leq \sum_{i=1}^{n} w(X_i)^2 s(X_i) - \left(\sum_{i=1}^{n} w(X_i)s(X_i)\right)^2$$

If this inequality is true, one would favor PAPE over IW, and conversely so if it is false. Although this heuristic is not guaranteed to result in better estimates in every case, for any sufficiently large n, it should result in better estimates on average. If (for whatever reason) one has to choose the estimator before observing any data, a rational agent would make the (uninformative) assumption about the weights, assigning $w(\boldsymbol{x}) = 1$, which by criterion (20) would lead them to choose PAPE.

D Additional Experiments

D.1 Sample Size Effect

The experiments described in Section 4 were run for an arbitrarily chosen evaluation data chunk size (2000 observations). In this section, we describe an ablation study, where we investigated the quality of performance estimates for different chunk sizes. We split production data into chunks of the following sizes: 100, 200, 500, 1000, 2000, and 5000. For each consecutive sample and each chunk size, the first instance of the data chunk was advanced 1,000 observations so that the first instances of each chunk for different chunk sizes were aligned. This was done to make the results between different chunk sizes comparable and to minimize the effect of changes in the underlying data. This meant that for chunk sizes of less than 1,000, not all data was used, and for chunk sizes larger than 1,000, there was some overlap between consecutive samples.

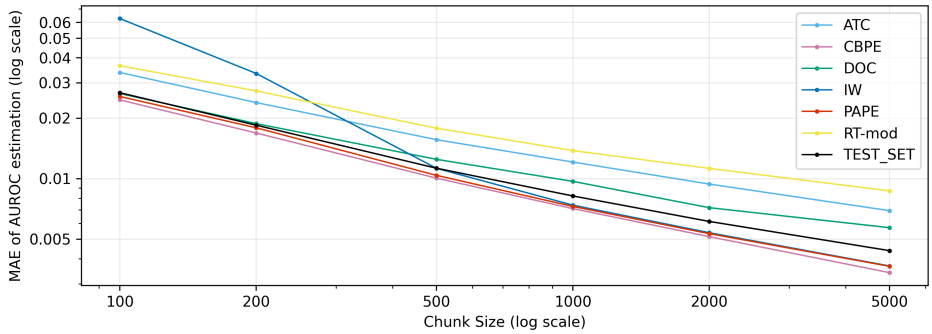

Figure 3: Effect of sample size on mean absolute error of AUROC estimation. Calculated for sample sizes of 100, 200, 500, 1000, 2000, and 5000, on data from California, for the prediction task: employment.

Due to computational complexity, we ran the experiment only for one evaluation case. We selected the biggest data set (California), the prediction task for which performance changes significantly (ACSEmployment), the default choice algorithm for the classification task on tabular data - LGBM, and the default metric - AUROC. Since the results come from a single evaluation case, we used a typical regression metric - mean absolute error (MAE). Figure 3 shows that all methods give less accurate AUROC estimates for smaller chunk sizes. This is expected as the random noise effects are more significant for small samples. For the evaluated case, this has the strongest impact on IW accuracy of estimation.

D.2 TableShift Experiments

In addition to our main experiments with US census data, we tested PAPE and our other benchmark methods with tabular data from the recently published TableShift benchmark [35]. Due to the defective functionality of the TableShift API, we were unable to extract all 15 datasets contained within the benchmark. Thus, we resorted to using a subset of 8 datasets we were able to retrieve, namely: ASSISTments, College Scorecard, Diabetes, Food Stamps, Hospital Readmission, Hypertension, Income, and Unemployment[7]. There is some overlap between these datasets and the datasets used in Folktables [36]. The main differences are that the datasets in TableShift come from a wider range of providers, the datasets are generally smaller, and the datasets are preprocessed differently, in particular, to create distributional shifts between the In-Domain (ID) and Out-of-Domain (OoD) portions. These shifts do not necessarily conform to our covariate shift assumption, giving us an interesting opportunity to experiment on how PAPE performs relative to other estimators in undefined types of shift scenarios.

	Accuracy		AUROC		F_1	
	NMAE	NRMSE	NMAE	NRMSE	NMAE	NRMSE
TEST SET	2.11	2.92	1.03	1.82	1.01	1.49
RT-mod	2.22	3.05	1.39	2.15	1.19	1.69
COT	2.80	3.49	-	-	-	-
ATC	1.94	3.20	1.32	2.06	1.50	2.11
DOC	1.74	2.54	0.94	1.67	1.03	1.45
CBPE	1.79	2.46	0.97	1.65	0.77	1.32
IW	1.59	2.06	0.88	1.13	0.75	0.95
PAPE	**1.52**	**1.98**	**0.83**	**1.06**	**0.67**	**0.86**

Table 2: NMAE and NRMSE of the evaluated performance estimation methods for each estimated metric.

[7]These datasets are described in detail at `https://tableshift.org/datasets.html`

We used the same evaluation framework as with our main experiment in Section 4 with the same benchmarks, implementation details, and evaluation metrics. However, the data preprocessing was handled a bit differently. We used the "training" portion of the datasets for training our models, the "validation" portion (both for ID and OoD) to train our estimators, and the "test" portion (both for ID and OoD) for evaluation. We concatenated the ID and OoD sets because for some datasets, the OoD portions were very small, sometimes less than 2,000 instances, which we used as our chunk size. The results shown in Table 2 align with our main experiments, showing that PAPE is superior to other estimators for all metrics tested.

D.3 Experiments With Synthetic Data

Since all the other experiments thus far have been performed on real-world data, where we cannot fully control the nature of the datset shift, we conducted further experiments with synthetic data created to ensure that only covariate shift is present. We start by describing the data generation process.

D.3.1 Data Creation

We created a spherically symmetric distribution of inputs centered at the origin of $\mathbb{R}^d$, with dimensionality $d = 20$. More specifically, we generated feature vectors by first drawing Gaussian directions $G \in \mathbb{R}^{20}$ with independent $N(0, 1)$ entries and normalizing each vector to unit length, $U_i = G_i / \|G_i\|_2$, thereby ensuring isotropy on the unit sphere. Radial distances R_i were then sampled independently from a half-normal distribution with $\sigma = 0.15$. Each observed feature instance was then derived as $X_i = R_i \cdot U_i$, and only those with radius $R_i < 0.5$ were retained. This procedure yielded a base pool of 100,000 feature instances with uniform angular structure and a controlled radial distribution.

Next, labels for each feature instance were assigned based on a smooth, monotonic relationship between the features and the probability of the positive class. For each observed feature instance $\boldsymbol{x}$, the radius $r(\boldsymbol{x}) = \|\boldsymbol{x}\|_2$ was calculated and the conditional probability of the positive class was set to $P(y = 1 \mid \boldsymbol{x}) = 1 - r(\boldsymbol{x})$. The true labels y were then sampled from the corresponding Bernoulli distribution. The base dataset was randomly partitioned into training (80,000) and reference (20,000) subsets, and a LightGBM classifier with default parameters was trained on the $d = 20$ features using samples from the training set.

The distribution of the resulting distances in the training dataset is visualized in Figure 4. We see that most instances reside near the origin and are thus likely to bear the positive label. Instances farther from the origin get increasingly rarer. This results in an unbalanced dataset, where positive labels are far more common, with the fraction of positive labels in the training dataset being roughly 88%.

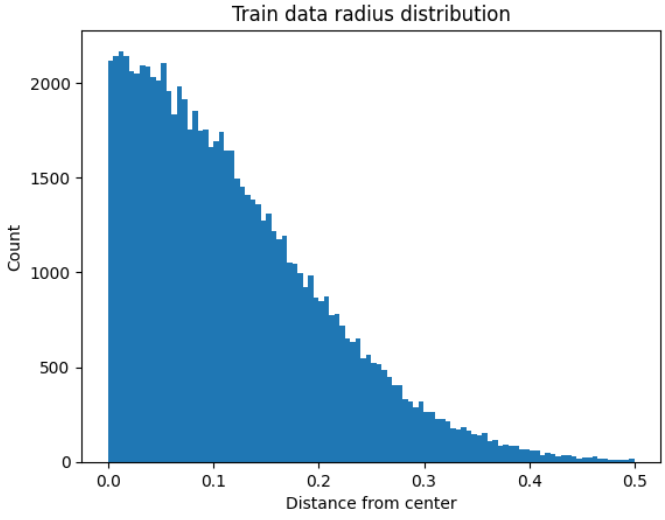

Figure 4: The distribution of distances from the origin in the training set for the synthetic data.

D.4 Experiment with Gradually Increasing Covariate Shift

In this experiment, we simulated covariate shift by repeatedly sampling production chunks of size 2,000 while gradually increasing a threshold for the radius of samples to include within the production data chunk. We let the threshold increase from 0 to 0.4 in increments of 0.025. For each resulting threshold t, we conducted 1,000 trials, where in each trial we sampled a chunk of data while enforcing $r(\boldsymbol{x}) > t$. Increasing the threshold is used to simulate a gradually increasing covariate shift. This makes predicting the right label by the monitored classifier increasingly difficult for two reasons. First, when the input data distribution shifts further away from the center, the label entropy increases since $P(y = 1 \mid \boldsymbol{x})$ is equal to 1 in the center and 0.5 at the hypersphere with $r(\boldsymbol{x}) = 0.5$. Second, the regions further away from the center are less dense in the reference distribution (see Figure 4). Thus, we expect the performance of the classifier to decrease with increasing shift magnitude, and we would like our estimators to be able to catch that.

After the 1,000 trials for each threshold, we averaged both the true model performances and the estimated performances for accuracy, F_1 score, and AUROC. The results are plotted in Figure 5.

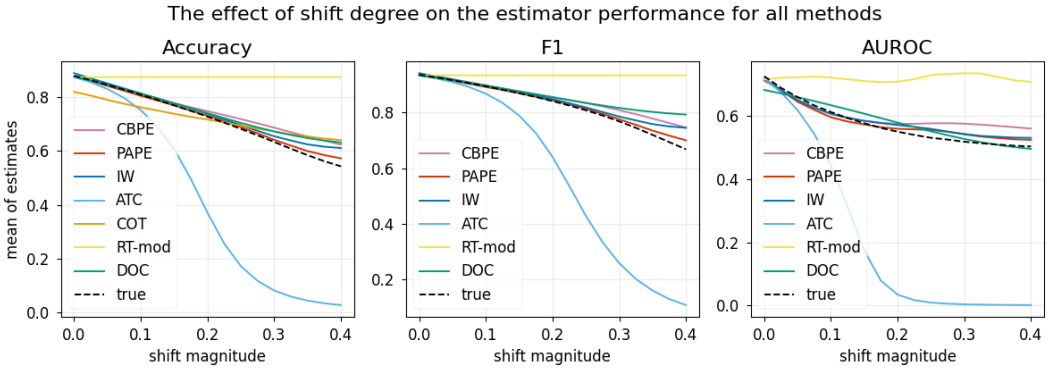

Figure 5: Estimator performance of each method for all three metrics with gradually increasing covariate shift. Shift magnitude corresponds to threshold t.

We see that both ATC and RT-mod are generating estimates that are unusable, so we decided to drop them from further analysis and focus on the rest. For the remaining methods, we plot the Mean Absolute Error (MAE) between the estimated and true metric values. The results are seen in Figure 6, where we see that PAPE is consistently able to generate estimates with low error even for the most intense shifts for all three metrics and has the best overall performance of all the methods tested.

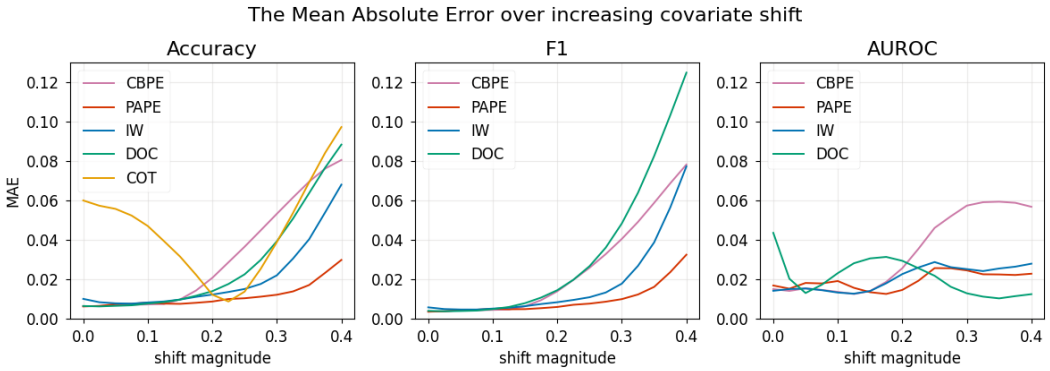

Figure 6: The Mean Absolute Errors of the estimators for all three metrics with gradually increasing covariate shift.

D.5 Experiment with Different Chunk Sizes

In this experiment, we fix the threshold to 0.25 and vary the chunk size to see how it affects the performance of the estimators. We test on chunk sizes of [100, 200, 400, 800, 1600, 3200], again, performing 1,000 trials for each chunk size. As in the previous experiment with increasing covariate shift, we noted that ATC and RT-mod performed so poorly that they were left out of the comparison. For the other estimators, we show the Mean Absolute Error (MAE) between the estimated and true metric values in Figure 7. Unsurprisingly, all methods provide better estimates with increasing chunk size, with PAPE and IW still improving while other estimators start to flat out. Overall, PAPE shows the best performance.

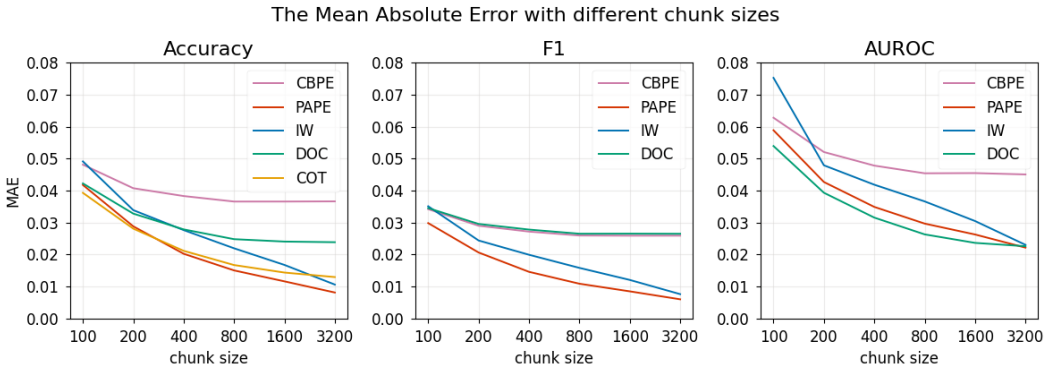

Figure 7: The Mean Absolute Errors for each estimator for all three metrics over different chunk sizes.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Claims made in the abstract and introduction describe the main characteristics of the proposed method and state that overall it has given more accurate performance estimates on the datasets used for testing compared to other methods. All of the claims are true and do not overgeneralize.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper contains a dedicated section (Section 5.1, Limitations), where the theoretical and practical limitations of the proposed method are discussed.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We present two theorems in the main paper. Formal proofs for these theorems are provided in Appendix B along with three additional lemmas and one additional theorem. The lemmas are needed to prove the theorems. All lemmas and theorems are explicitly numbered and properly cross-referenced, and all required assumptions are clearly and exhaustively expressed.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The proposed method is explained in detail in Section 3. The details of the conducted experiments are given in 4. More specifically, we describe the (open access) datasets in Section 4.1. Data preprocessing and model training are explained in Section 4.2. The implementation details of the benchmark methods are described concisely in Section 4.3 and in more detail in Appendix A. Finally, the evaluation methodology is clearly described in 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We make our code available as a public github repository (`https://github.com/pape-research/pape_r`). The code together with the paper enables reproducing all the experiments and their results described in the paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The description of the proposed method (Sections 3 and 4.3), other evaluated benchmark methods (Section 4.3), and experimental setup (Sections 4.1 and 4.2) are given in the main paper and Appendix A. The low-level details of the method implementations can be found in the code we provide in our public repository.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Statistical significance is provided for the experimental results supporting the main claim of the paper. It is provided in Section 4.4 and reproducible with the code we supply via our public repository. We did not produce error bars for our plotted graphs due to readability issues and the computational complexity it would have required.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The description of resources and computation time for the described calculations are provided in footnotes of the relevant sections.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We use public, anonymised datasets. To our best knowledge, there are no harmful consequences of our research.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The paper contributes to foundational research.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: To our best judgement, the paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly cite all sources of existing assets with their licenses. All standard py-data stack libraries we use (like numpy, matplotlib etc.) are not necessarily cited, but they are listed in the requirements.txt in our public code repository.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces a novel algorithm, and the code repository linked by the authors contains its implementation. This code is released under the CC BY-NC-SA 4.0 license, which is stated in the LICENSE file in the repository.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs in any way.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.