Weighted Conformal Clustering

Anirban Nath
Department of Statistics
Columbia University
an3145@columbia.edu

YoonHaeng Hur
Department of Statistics
Columbia University
yoonhaeng.hur@columbia.edu

Genevera I. Allen
Department of Statistics
Columbia University
genevera.allen@columbia.edu

Abstract

Clustering is a central tool for discovering latent structure in unlabeled data; yet modern clustering pipelines often end with a hard assignment of each observation to a cluster without rigorous measures of assignment uncertainty. We propose a novel weighted conformal approach for constructing valid confidence sets for cluster labels. The key difficulty is that the labels available for calibration are not observed ground-truth labels, but synthetic labels produced by a data-dependent clustering algorithm. Our method develops a conformal inference algorithm that corrects the resulting mismatch with the latent target labels through weights by formulating conformal clustering as a conditional label-distribution shift problem. We first derive an oracle procedure that attains finite-sample marginal coverage and then develop a computationally tractable and implementable version using estimated conditional label probabilities and novel augmented calibration. We show that the coverage of the estimated-weight procedure depends on the estimator, giving an explicit bound on the loss relative to the nominal level. Empirical studies demonstrate that the proposed weighted approach offers improvements over the recently proposed split conformal clustering procedure in terms of informative confidence set size, especially in nonlinear and high-dimensional clustering applications.

1 Introduction

Clustering is a central tool deployed to uncover latent group structures, which has been integrated as a fundamental component of modern data analysis pipelines across many scientific and industrial domains [15, 7]. Despite the advances in methodology and software, clustering often culminates in a hard assignment of each observation to a cluster without rigorous measures of assignment uncertainty, which can be problematic when the resulting cluster labels are used for subsequent scientific interpretation or operational decisions. Accordingly, it is instrumental to develop rigorous methods for uncertainty quantification for clustering, which has been approached from macroscopic or structural perspectives in the literature. Prominent methodologies include assessing centroid variability via bootstrapping [12, 9, 20], testing the global existence and separation of clusters [21, 10, 14, 27, 33, 4, 8], and generating Bayesian credible sets over entire partitions [31, 6]. However, these global structural measures largely overlook a crucial, more granular target: the uncertainty inherent in the specific cluster assignment of an individual observation.

Quantifying cluster assignment uncertainty is uniquely challenging because cluster labels are categorical, invariant to permutations, and typically generated by deterministic procedures deeply dependent on the observed dataset. Hence, these assignments lack the properties of typical continuous parame-

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

ters, making the standard error-based analysis inapplicable. A further difficulty is that the partition returned on the observed data does not, by itself, define a calibrated rule for labeling new points or for expressing uncertainty about those labels. To address this, we take a viewpoint that frames label uncertainty as the reliability of a predictive labeling function and introduce a versatile methodology that transforms any base clustering algorithm into a system capable of generating statistically valid confidence sets for cluster membership anywhere in the feature space. These sets offer highly interpretable uncertainty quantification: singletons indicate clear, unambiguous assignments, while multi-label sets highlight ambiguous regions where several cluster identities are possible.

Recently, [11] proposed a conformal-inference approach to quantify uncertainty in cluster labels. Conformal inference is a general framework for constructing prediction sets with finite-sample, distribution-free coverage guarantees under exchangeability, without requiring a correctly specified model for the data distribution [30]. This framework has been especially influential in supervised learning, where it has been used to quantify predictive uncertainty in regression [18, 24], classification [17, 26, 25], and other applications. Its appeal lies in the fact that conformal methods can be applied on top of many predictive algorithms, converting fitted scores or predictions into set-valued outputs with rigorous coverage guarantees using a suitable calibration sample. [11] demonstrated that conformal inference can be adapted to unsupervised clustering in a principled way by framing label assignment as a predictive problem.[1] At a high level, their approach adapts split conformal classification to clustering by using stochastic cluster labels to capture uncertainty, fitting a predictive rule for these labels, and calibrating conformity scores to return confidence sets for cluster membership at new points. However, [11] specifically states that since the calibration labels are synthetically generated by the clustering algorithm rather than sampled from a true ground-truth distribution, this procedure fundamentally violates the exchangeability assumption required for standard conformal validity. The paper then quantifies the resulting undercoverage in terms of the consistency and replace-one stability of the clustering algorithm, with a particular focus on smooth parametric mixture models. While this framework provides a valuable first step towards conformal clustering, important theoretical and methodological questions remain, with the need for investigation beyond the mixture-model setting and for more general clustering algorithms.

In this paper, we develop a novel conformal inference framework for uncertainty quantification of cluster labels by utilizing weighted conformal prediction to address the distributional mismatch. Introduced by [29] in the context of covariate shift, weighted conformal prediction extends conformal inference beyond iid settings by reweighting the calibration distribution with likelihood-ratio weights, and has since motivated a broader line of work on conformal prediction under distribution shift. This perspective is well-suited to clustering because the labels used for calibration are generated by a clustering algorithm, whereas the desired coverage is with respect to the unobserved latent labels. We address this first by theoretically capturing the oracle conformal weights, which guarantee exact finite-sample coverage when the true conditional label distribution is known. Then, we develop a computationally tractable and fully data-driven algorithm that replaces the oracle quantities with estimated conditional probabilities and show that the coverage loss due to estimation is explicitly bounded by the estimation error of the chosen conditional label estimator. Finally, we empirically demonstrate the advantages of this method over [11] in different settings.

Our Contribution We make three main contributions. First, we introduce a novel framework for uncertainty quantification of cluster labels that formulates this as a conditional label-distribution shift problem. We use a weighted conformal framework beyond exchangeable settings to quantify the validity of this procedure. Secondly, we propose a new augmented calibration mechanism that enjoys the benefits of both full-conformal and split-conformal approaches. This is an innovative idea with potential applications in conformal prediction for unsupervised learning beyond clustering, which broadens the scope of weighted conformal approaches in general. Finally, we show that this new framework achieves valid and more informative confidence sets in high-dimensional and nonlinear clustering problems, including spectral clustering applications, thereby extending the conformal clustering framework of [11] beyond its original setting.

[1] As noted in [11], their framework is different from related approaches under the name of "conformal clustering" [5, 22, 13], which use conformal p-values [2, 19, 16] to characterize and cluster outliers, rather than quantifying uncertainty.

2 Weighted Conformal Clustering: Coverage Guarantees

2.1 Problem Formulation

Adopting the inference framework for cluster labels proposed by [11], we formally state the problem as follows. Given covariates $X_1, \ldots, X_n \in \mathbb{R}^p$, clustering assigns labels in $[K] := \{1, \ldots, K\}$, where K is fixed and known. We assume a latent-label model in which $(X_i, Y_i^*)_{i=1}^n$ are i.i.d. from an unknown distribution P^* on $\mathbb{R}^p \times [K]$, but only the covariates are observed. For a new test point $(X_{n+1}, Y_{n+1}^*) \sim P^*$, independent of the observed data, our goal is to construct a set-valued map $\hat{\mathcal{C}} \colon \mathbb{R}^p \to 2^{[K]}$ that contains the true cluster label with high probability. Because cluster labels are identifiable only up to permutation, coverage must be stated after aligning the output labels with the latent ground-truth labels. To this end, the notion of oracle alignment was introduced in [11] as follows: $\hat{\sigma}_o^* \in \arg\max_{\sigma \in \mathcal{S}_K} \mathbb{P}(\sigma(Y^*) \in \hat{\mathcal{C}}(X) \mid \hat{\mathcal{C}})$, where $\mathcal{S}_K$ is the set of all permutations of $[K]$ and $(X, Y^*) \sim P^*$ is independent of the randomness used to construct $\hat{\mathcal{C}}$. We seek a confidence set satisfying, for a prescribed level $\alpha \in (0, 1)$,

$$\mathbb{P}\left(\hat{\sigma}_o^*(Y_{n+1}^*) \in \hat{\mathcal{C}}(X_{n+1})\right) \geq 1 - \alpha. \tag{1}$$

Thus, $\hat{\mathcal{C}}(x)$ quantifies pointwise cluster-label uncertainty: a singleton set indicates a relatively certain assignment, while a larger set indicates that multiple cluster labels remain plausible.

Since we do not know the true labels, we have to rely on the labels generated by the clustering algorithm. For that, we define the following notation.

Definition 1 (Stochastic clustering algorithm)**.** Let $\mathcal{X}$ denote the domain on which cluster assignments are defined, and let $\Delta_K \subset \mathbb{R}^K$ be the probability simplex. A clustering algorithm $\mathcal{A}$ is called a *stochastic clustering algorithm* if, for any input dataset $\mathcal{D}_n = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^p$, it outputs a mapping $\mathcal{A}_{\mathcal{D}_n} : \mathcal{X} \to \Delta_K$. For each $x \in \mathcal{X}$, the vector $\mathcal{A}_{\mathcal{D}_n}(x) = \big(\mathcal{A}_{\mathcal{D}_n}(x)_1, \ldots, \mathcal{A}_{\mathcal{D}_n}(x)_K\big)$ represents the soft cluster assignment probabilities for x, where $[\mathcal{A}_{\mathcal{D}_n}(x)]_k$ is interpreted as the probability that x belongs to cluster k. A realized cluster label for x is then generated according to $Y \mid x, \mathcal{D}_n \sim \mathrm{Cat}\big(\mathcal{A}_{\mathcal{D}_n}(x)\big)$ where, for any $w \in \Delta_K$, Cat(w) denotes the categorical distribution on $[K]$ that draws $k \in [K]$ with probability w_k.

Remark 1. For this paper, we consider the map $\mathcal{D} \mapsto \mathcal{A}_{\mathcal{D}}$ that is permutation-invariant in its argument, in the sense that it depends only on the unordered set $\mathcal{D}$.

The framework in Definition 1 is broad enough to encompass all clustering procedures essentially. For non-generalizable methods such as k-means or hierarchical clustering, assignments are produced only for the observed data points, so $\mathcal{X} = \mathcal{D}_n$. In contrast, model-based methods such as Gaussian mixture models define assignment probabilities for any $x \in \mathbb{R}^p$, in which case $\mathcal{X} = \mathbb{R}^p$, and we call the procedure a *generalizable stochastic clustering algorithm*. Moreover, deterministic clustering algorithms can also be explained as a special case: if an algorithm assigns hard labels $y_1, \ldots, y_n \in [K]$, then we may set $[\mathcal{A}_{\mathcal{D}_n}(x_i)]_k = 1\{k = y_i\}$, so that $Y_i \sim \mathrm{Cat}(\mathcal{A}_{\mathcal{D}_n}(x_i))$ is degenerate at the assigned label. Thus, hard clustering corresponds to point-mass assignment probabilities, while soft clustering allows nondegenerate probabilities over multiple clusters.

2.2 Prelude: Oracle-Weighted Split Conformal Clustering and Its Limitations

As discussed in Section 1, the proposed uncertainty quantification framework of [11] through conformal prediction generates labels by a fitted clustering rule. Since the synthetic cluster labels are produced in a data-dependent manner, the cluster-labeled calibration sample is not exchangeable with the pair of test point and its true label, so the classical conformal validity argument does not apply directly. To make this precise, note that for any unordered covariate set $\mathcal{D}$, the stochastic clustering algorithm $\mathcal{A}_{\mathcal{D}}$ induces a conditional label distribution on $[K]$ via $Y \mid X = x, \mathcal{D} \sim \mathrm{Cat}(\mathcal{A}_{\mathcal{D}}(x))$, which along with the true covariate marginal P_X^*, yields a synthetic joint distribution, say $\tilde{P}$ on $\mathbb{R}^p \times [K]$, whereas the target distribution governing the true latent labels is P^* defined below:

$$\tilde{P}(dx, dy) := P_X^*(dx) \sum_{k=1}^{K} [\mathcal{A}_{\mathcal{D}}(x)]_k \, \delta_k(dy), \quad P^*(dx, dy) = P_X^*(dx) \sum_{k=1}^{K} p_{Y|X}^*(k \mid x) \, \delta_k(dy),$$

where $p^*_{Y|X}(\cdot \mid \cdot)$ is the conditional density corresponding to the oracle conditional distribution of the labels, namely $P^*_{Y|X}$. Thus, $\tilde{P}$ and P^* have different joint laws although they share the same marginals. The clustering procedure supplies calibration labels from $\tilde{P}$, whereas the desired guarantee in (1) concerns labels from P^*. This perspective frames the problem as a form of conditional label-distribution shift, or *concept shift*, where validity requires correcting the discrepancy between the synthetic label mechanism induced by $\mathcal{A}_{\mathcal{D}}$ and the true latent-label mechanism.

Distribution shift has been widely studied in the conformal inference literature, with much of the focus on covariate shift [29] and label shift [23, 28], which can be extended to more general forms of shift. The key idea behind this is to reweight the calibration scores when forming the empirical score distribution. With correctly specified weights, the resulting weighted distribution matches the conditional empirical distribution of the test-point score, and its weighted quantile can be used to determine whether the test point is included in the prediction set. The standard conformal procedure is a special case of this, where we choose uniform weights for all points. However, many existing methods rely on the exchangeability of the augmented calibration-test sample, which is absent in our setting. The problem considered here is thus more delicate because we not only encounter a distribution shift problem but also a non-exchangeable data-generating procedure. Therefore, we must view the problem through a more general lens of conformal prediction under arbitrary joint distributions, where validity does not require exact exchangeability of the augmented sample. We briefly review this framework in Appendix A.

In [11], the failure of exchangeability is solely treated as a source of error whose effect can be quantified, leading to undercoverage bounds and asymptotic validity under additional stability and model assumptions. In contrast, we want to address this exchangeability issue more directly by seeking a conformal construction that accounts for the distributional mismatch induced by synthetic cluster labels. To this end, we attempt to derive the oracle-weighted version of the split conformal clustering [11]. Instead of assigning uniform weights to the conformity scores, we calculate the weights that correct the discrepancy discussed above, leading to a weighted empirical score distribution and a corresponding weighted prediction rule.

Based on the general theory of conformal prediction adapted to our setting, we can choose the desired weights as follows. Let $\mathcal{D}_{ca} = \{X_i\}_{i \in \mathcal{I}_{ca}}$ denote the calibration covariates with $\mathcal{I}_{ca}$ denoting their indices. Let $(X_i, Y_i)_{i \in \mathcal{I}_{ca}}$ be the corresponding calibration sample, where the labels are generated by the stochastic clustering algorithm $\mathcal{A}_{\mathcal{D}_{ca}}$ as defined in Definition 1. Let x be the test covariate of interest. For $i \in \mathcal{I}_{ca}$, let $\mathcal{D}^{-i}_{ca,\{x\}} = \mathcal{D}_{ca} \cup \{x\} \setminus \{X_i\}$ and define the unnormalized weights as

$$\tilde{w}_i(y) \propto p^*_{Y|X}\left(Y_i \mid X_i\right) \left(\prod_{j \in \mathcal{I}_{ca} \setminus \{i\}} \left[\mathcal{A}_{\mathcal{D}^{-i}_{ca,\{x\}}}(X_j) \right]_{Y_j} \right) \left[\mathcal{A}_{\mathcal{D}^{-i}_{ca,\{x\}}}(x) \right]_y , \tag{2}$$

while the test-point weight is

$$\tilde{w}(y) \propto p^*_{Y|X}(y \mid x) \prod_{j \in \mathcal{I}_{ca}} \left[\mathcal{A}_{\mathcal{D}_{ca}}(X_j) \right]_{Y_j} , \tag{3}$$

which are normalized to $\{w_i(y)\}_{i \in \mathcal{I}_{ca}} \cup \{w(y)\}$ that sum to one. Using these weights, we derive the oracle-weighted version of split conformal clustering proposed in [11]; see Algorithm 0.

Algorithm 0 modifies the split conformal clustering procedure (Algorithm 1 of [11]) by introducing the weights in Step 5 and consequently constructs $\hat{\mathcal{C}}$ by calculating the weighted quantile $\mathsf{Q}_{1-\alpha}$ of the conformity scores; see Definition 2. Here, s in Step 6 is any suitable conformity score function used in conformal classification. In Algorithm 1 of [11], Step 5 was the alignment between the calibration labels and the training classifier, which is skipped in Algorithm 0 for the reason we explain later; see also Remark 5. Now, the following theorem formalizes the marginal coverage guarantee of Algorithm 0.

Theorem 1 (Finite-sample coverage of oracle-weighted algorithm)**.** *Suppose $X_1, \ldots, X_n \in \mathbb{R}^p$ are i.i.d. from P^*_X. Let $\hat{\mathcal{C}}$ be the output of Algorithm 0. Then, for $(X_{n+1}, Y^*_{n+1}) \sim P^*$ independent of $\{X_i\}_{i=1}^n$, we have $\mathbb{P}(\hat{\sigma}^*_o(Y^*_{n+1}) \in \hat{\mathcal{C}}(X_{n+1})) \geq 1 - \alpha$.*

The proof of Theorem 1 is deferred to Appendix B. The idea is to utilize the general weighted conformal inference theory based on permutations, which we review in Appendix A. The crux of this

Algorithm 0 Weighted Split Conformal Clustering (Oracle)

Require: Covariates $X_1, \ldots, X_n \in \mathbb{R}^p$, user-specified level $\alpha \in (0,1)$, test point $X_{n+1} = x$.

1: Split the covariates into a training set $\{X_i\}_{i\in\mathcal{I}_{tr}}$ and a calibration set $\{X_i\}_{i\in\mathcal{I}_{ca}}$.
2: Apply clustering to the training data $\{X_i\}_{i\in\mathcal{I}_{tr}}$ and obtain corresponding labels $\{Y_i\}_{i\in\mathcal{I}_{tr}}$.
3: Fit a soft classifier $\hat{\pi}\colon \mathbb{R}^p \to \Delta_K$ on the cluster-labeled training data $\{(X_i, Y_i)\}_{i\in\mathcal{I}_{tr}}$.
4: Apply clustering to the calibration data $\mathcal{D}_{ca}$ and obtain corresponding labels $\{Y_i\}_{i\in\mathcal{I}_{ca}}$.
5: For each candidate label $y \in [K]$, calculate the calibration weights $w_i(y)$ for $i \in \mathcal{I}_{ca}$ and the test weight $w(y)$ obtained by normalizing the weights in (2)-(3).
6: Define suitable conformity scores $S_i = s((X_i, Y_i); \hat{\pi})$ for $i \in \mathcal{I}_{ca}$ and construct

$$\hat{\mathcal{C}}(x) = \left\{ y \in [K] : s((x,y);\hat{\pi}) \le \mathsf{Q}_{1-\alpha}\left(\sum_{i\in\mathcal{I}_{ca}} w_i(y)\, \delta_{S_i} + w(y)\delta_{\infty} \right) \right\}.$$

7: **return** $\hat{\mathcal{C}}(x)$.

theory is to derive permutation-dependent weights from the joint distribution. We show that these weights can be reduced to (2) and (3) in our clustering setting. Then, from the general theory, we can show the desired coverage. Lastly, we remark that while the aforementioned alignment step in Algorithm 1 of [11] is natural in practice, we have excluded this step in Algorithm 0 to keep the theoretical derivation straightforward and streamlined.

While Theorem 1 provides the desired coverage guarantee, the exact oracle weights reveal two obstacles to implementation. First, they involve the unknown conditional density $p^*_{Y|X}$, which is an oracle quantity and must be estimated in practice. Second, their sample-dependent terms require evaluating the clustering rule on the leave-one-out augmented covariate set $\mathcal{D}^{-i}_{ca,\{x\}}$ for $i \in \mathcal{I}_{ca}$. Hence, for each test covariate x, the exact construction requires one clustering fit on $\mathcal{D}_{ca}$ and $|\mathcal{I}_{ca}|$ additional leave-one-out fits, which is computationally intractable. Also, this inevitably introduces label misalignment from multiple clustering fits. Therefore, beyond the statistical challenge of estimating the oracle label law, the exact oracle-weighted construction is computationally expensive even at the level of the sample-dependent terms, motivating a more tractable approximation.

2.3 A Tractable Estimated-Weight Construction

As the preceding discussion makes clear, a feasible weighted-conformal clustering procedure must not only overcome the computational burden of repeated clustering refits but also come with a tractable estimation of the latent conditional law $P^*_{Y|X}$. To alleviate the computational difficulty, we introduce a novel augmented calibration mechanism that eliminates the leave-one-out refitting burden in the oracle weights. Our key idea is to remove this dependence by modifying the synthetic-label generation mechanism itself. Instead of generating calibration labels from a clustering rule fitted on $\mathcal{D}_{ca}$, we fit the clustering algorithm once on the augmented covariate set for $X_{n+1} = x$, namely,

$$\mathcal{D}_{aug} := \mathcal{D}_{ca} \cup \{x\}.$$

This augmentation is a simple yet novel idea that bridges the full-conformal and split-conformal prediction, utilizing the advantages of both mechanisms. Consequently, the joint synthetic likelihood factors that previously necessitated repeated refits become common multiplicative terms and cancel after normalization. The corresponding oracle weights reduce to the pointwise likelihood-ratio form as follows

$$\tilde{w}_i \propto \frac{p^*_{Y|X}(Y_i \mid X_i)}{\left[\mathcal{A}_{\mathcal{D}_{aug}}(X_i)\right]_{Y_i}} \quad \text{for } i \in \mathcal{I}_{ca}, \qquad \tilde{w}(y) \propto \frac{p^*_{Y|X}(y \mid x)}{\left[\mathcal{A}_{\mathcal{D}_{aug}}(x)\right]_{y}}. \tag{4}$$

This transformation is both statistically and computationally consequential. Statistically, the weights retain the familiar distribution-shift interpretation, with each ratio comparing the target latent-label probability in the numerator to the synthetic label probability induced by the augmented clustering fit in the denominator. Computationally, the augmented construction replaces the $|\mathcal{I}_{ca}| + 1$ clustering fits required by the exact oracle weights in (2)–(3) with a single fit on $\mathcal{D}_{aug}$. Thus, the augmented calibration step is the pivotal mechanism that turns the weighted conformal correction into a tractable framework for cluster-label uncertainty quantification.

The augmented construction removes the main computational obstacle, but the weights in (4) still depend on the unknown conditional density $p^*_{Y|X}$. To obtain an implementable method, let $\hat{p}_{Y|X}$ be

a suitable estimator of the true conditional label density $p^*_{Y|X}$ based on the training split, which is aligned with the training clustering. We then replace the oracle numerator in (4) by $\hat{p}_{Y|X}$, giving the estimated weights

$$\hat{w}_i \propto \frac{\hat{p}_{Y|X}(\hat{\sigma}(Y_i) \mid X_i)}{\left[\mathcal{A}_{\mathcal{D}_{aug}}(X_i)\right]_{Y_i}} \quad \text{for } i \in \mathcal{I}_{ca}, \qquad \hat{w}(y) \propto \frac{\hat{p}_{Y|X}(y \mid x)}{\left[\mathcal{A}_{\mathcal{D}_{aug}}(x)\right]_{\hat{\sigma}^{-1}(y)}}, \tag{5}$$

where $\hat{\sigma}$ is a permutation that aligns the augmented cluster labels to the training clustering and hence to $\hat{p}_{Y|X}$. The weights in (5) are normalized to $\{w_i(y)\}_{i\in\mathcal{I}_{ca}} \cup \{w(y)\}$ so that they sum to one, which are used in the weighted conformal distribution separately for each candidate label y; here, we place y-dependence in the normalized weights because the normalization turns the unnormalized calibration weights $\hat{w}_i$'s in (5) into y-dependent weights. Algorithm 1 summarizes the above procedure.

Algorithm 1 Weighted Conformal Clustering (Augmented)

Require: Covariates $X_1, \ldots, X_n \in \mathbb{R}^p$, user-specified level $\alpha \in (0,1)$, test point $X_{n+1} = x$.

1: Split the covariates into a training set $\{X_i\}_{i\in\mathcal{I}_{tr}}$ and a calibration set $\{X_i\}_{i\in\mathcal{I}_{ca}}$.
2: Apply clustering to the training data $\{X_i\}_{i\in\mathcal{I}_{tr}}$ and obtain corresponding labels $\{Y_i\}_{i\in\mathcal{I}_{tr}}$.
3: Fit a soft classifier $\hat{\pi}\colon \mathbb{R}^p \to \Delta_K$ on the cluster-labeled training data $\{(X_i, Y_i)\}_{i\in\mathcal{I}_{tr}}$.
4: Apply clustering to the augmented data $\mathcal{D}_{aug} := \mathcal{D}_{ca} \cup \{x\}$ and obtain the labels $\{Y_i\}_{i\in\mathcal{I}_{ca}}$ for the calibration data.
5: Align the cluster and classification labels: for a suitable metric ℓ on Δ_K, find

$$\hat{\sigma} \in \arg\min_{\sigma\in\mathcal{S}_K} \textstyle\sum_{i\in\mathcal{I}_{ca}\cup\{n+1\}} \ell(\sigma(\hat{\pi}(X_i)), \mathcal{A}_{\mathcal{D}_{aug}}(X_i)),$$

where $\sigma(w) = (w_{\sigma(1)}, \ldots, w_{\sigma(K)})$ for any $w \in \Delta_K$ and $\sigma \in \mathcal{S}_K$.
6: Estimate a suitable conditional density $\hat{p}_{Y|X}$ based on the training split.
7: For each candidate label $y \in [K]$, calculate the calibration weights $w_i(y)$ for $i \in \mathcal{I}_{ca}$ and the test weight $w(y)$ obtained by normalizing the weights in (5).
8: Define suitable conformity scores $S_i = s((X_i, \hat{\sigma}(Y_i)); \hat{\pi})$ for $i \in \mathcal{I}_{ca}$ and construct

$$\hat{\mathcal{C}}(x) = \left\{ y \in [K] : s((x,y);\hat{\pi}) \le \mathsf{Q}_{1-\alpha}\left(\textstyle\sum_{i\in\mathcal{I}_{ca}} w_i(y)\,\delta_{S_i} + w(y)\delta_\infty\right)\right\}.$$

9: **return** $\hat{\mathcal{C}}(x)$.

Remark 2. In Algorithm 1, while $\hat{p}_{Y|X}$ can be any suitable estimator of $p^*_{Y|X}$ as long as it is independent of the calibration split, the most natural choice is to use the soft classifier $\hat{\pi}$ from Steps 2-3, namely, $\hat{p}_{Y|X}(\cdot \mid x) = \hat{\pi}(x)$. Also, note that Algorithm 1 can accommodate hard clustering such as k-means. In this case, the augmented clustering fit produces degenerate assignment probabilities, with $[\mathcal{A}_{\mathcal{D}_{aug}}(X_i)]_{Y_i} = 1$, and $\hat{p}_{Y|X}$ can still be obtained by using the soft classifier $\hat{\pi}$ as explained earlier. Meanwhile, if one uses a generalizable soft clustering algorithm, the soft classifier $\hat{\pi}$ in Step 3 may be skipped since we can use it as $\hat{p}_{Y|X}$ directly. Lastly, note that the alignment in Step 5 is based on the augmented covariates and thus is different from the alignment in Step 5 of Algorithm 1 in [11]; also, we only use the covariates, not the obtained cluster labels. The alignment step can be done by the standard linear assignment algorithms in polynomial time. Lastly, we note that the denominators of the weights in (5) can be small or even zero (when using hard clustering), which can cause numerical instability. To deal with this issue, we clip the denominator, namely, set it as $\epsilon = 10^{-6}$ if it is below ϵ. If both the numerator and the denominator are below ϵ, so that the ratio is effectively $0/0$, we set the weight to 1. The weights are then normalized as before.

The following theorem derives a finite-sample bound on the coverage.

Theorem 2 (Finite-sample coverage with estimated weights)**.** *Suppose $X_1, \ldots, X_n \in \mathbb{R}^p$ are i.i.d. from P^*_X. Let $\hat{\mathcal{C}}$ be the output of Algorithm 1, where the conditional label distribution corresponding to the density $\hat{p}_{Y|X}$ in* (5) *is denoted as $\widehat{P}_{Y|X}$. Then, for $(X_{n+1}, Y^*_{n+1}) \sim P^*$ independent of $\{X_i\}_{i=1}^n$, we have*

$$\mathbb{P}\left(\hat{\sigma}^*_o(Y^*_{n+1}) \in \hat{\mathcal{C}}(X_{n+1})\right) \ge 1 - \alpha - \min_{\sigma\in\mathcal{S}_K} \mathbb{E}_{tr}\mathbb{E}_{X\sim P^*_X}\left[d_{\mathrm{TV}}\left(\widehat{P}_{Y|X}(\cdot \mid X), \sigma \circ P^*_{Y|X}(\cdot \mid X)\right)\right],$$

where d_{TV} is the total variation distance over the simplex, $\mathbb{E}_{tr}$ is the expectation with respect to the training split, and $\mathbb{E}_{X\sim P^*_X}$ is the expectation with respect to $X \sim P^*_X$ independent of everything else. Here, $\sigma \circ P^*_{Y|X}(\cdot \mid X)$ denotes the label distribution $P^*_{Y|X}(\cdot \mid X)$ permuted according to $\sigma \in \mathcal{S}_K$.

The proof of Theorem 2 is deferred to Appendix B. Theorem 2 combines two ideas: the augmentation mechanism of the calibration set and the estimation of the conditional density from the training set. Particularly, if we can perfectly estimate the oracle conditional density, namely, $\hat{p}_{Y|X} = p^*_{Y|X}$, then Algorithm 1 has $1 - \alpha$ coverage. For general $\hat{p}_{Y|X}$, Theorem 2 shows that the coverage loss of the estimated-weight procedure is controlled by the error in estimating the latent conditional label probabilities. The proof again builds on the general weighted conformal inference theory that derives the weights from the joint density of the calibration and test points, which, unlike Theorem 1, leads to the simpler weights in (5) thanks to the augmentation.

Remark 3. Notice that the optimization in Step 5 of Algorithm 1 may have multiple minimizers. In this case, we can use any deterministic, observation-order-invariant tie-breaking rule to define $\hat{\sigma}$. For example, one may select the lexicographically smallest permutation among all minimizers. Even a random tie-breaking rule is acceptable as long as the randomness is independent of everything else. Then, we can represent the permutation $\hat{\sigma}$ as a deterministic symmetric function of the augmented covariates, which allows us to use the general weighted conformal inference theory. We will formally define this in the proof of Theorem 2.

Unlike [11], our framework does not introduce an additional replace-one stability term, since the augmented calibration construction removes the instability from repeated leave-one-out clustering refits. The remaining requirement is consistency of the estimated conditional label law, which is intrinsic when targeting unobserved latent labels in an unsupervised setting. However, the two guarantees are not directly comparable, as our bound concerns out-of-sample conditional estimation error, while [11] studies the in-sample behavior of stochastic split conformal clustering. Nevertheless, under correctly specified smooth parametric mixtures, standard identifiability and regularity conditions yield consistent conditional label estimation, so both methods attain asymptotic validity. Beyond these classical mixture models, our weighted framework can accommodate more flexible conditional probability estimators, enabling broader clustering regimes; the next section illustrates this advantage in nonlinear examples, where our method maintains coverage with more informative prediction sets. Of course, for such nonlinear clustering examples, little is known about the consistency in general. Deriving conditions under which the clustering algorithm is consistent has been a long-standing problem in the literature, which is beyond the scope of this current paper. We leave the study of consistency of clustering that fits within our theoretical framework and [11] as future work.

While Theorem 2 emphasizes consistent estimation, exact model specification is not required: under misspecification, the coverage loss is governed by the discrepancy between the limiting estimated conditional law and the oracle law, which can be offset by a conservative choice of the nominal level. Although the framework can be applied to hard clustering methods such as k-means, their degenerate one-hot assignments cannot consistently recover a smooth latent conditional label distribution (see Remark 2 of [11]). So the undercoverage term in Theorem 2 generally does not necessarily vanish when using strictly deterministic assignments. Therefore, to fully capture assignment uncertainty and fully realize the theoretical guarantees afforded by this framework, we recommend using stochastic soft-clustering algorithms.

3 Empirical Studies

3.1 Simulations

Gaussian Mixture Models We simulate data from a $K = 5$ component Gaussian mixture model (GMM) with equal mixing proportions in low-dimensional ($\mathbb{R}^2$) and high-dimensional ($\mathbb{R}^{50}$) settings. Mixture centers are fixed, sharing the common covariance matrix $\sigma^2 I_p$. For $\alpha = 0.1$, we evaluate the coverage and set size of Algorithm 1 for varying σ^2 with fixed $n = 1200$ ($n = 3000$ for $\mathbb{R}^{50}$) and for varying sample size with fixed $\sigma^2 = 2.6$ ($\sigma^2 = 2.3$ for $\mathbb{R}^{50}$). Algorithm 1 is compared with the split conformal clustering (CC) method [11] and the naive cutoff method, which cuts the estimated soft labels to $1 - \alpha$. For $\mathbb{R}^2$, the stochastic version of GMM clustering is used for both Algorithm 1 and split CC. For $\mathbb{R}^{50}$, we use spectral clustering, which first applies singular value decomposition to the data and then applies stochastic clustering to the top K singular vectors. We use k-nearest neighbor classification (with k $= \sqrt{n_{tr}}$) and generalized inverse quantile [25] as the conformity score.

The top row of Figure 1 shows the results in $\mathbb{R}^2$, plotting the coverage and set size as a function of σ^2 or the sample size n. We can see that both Algorithm 1 and split CC achieve valid coverage

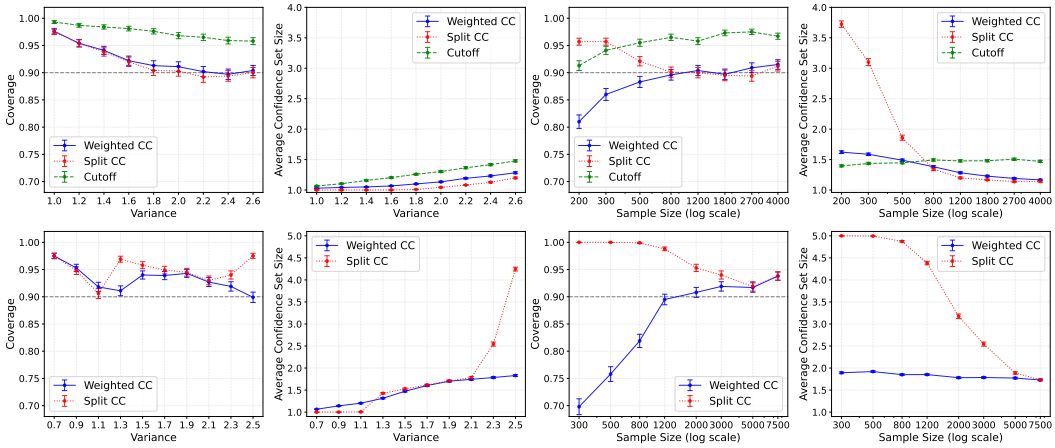


Figure 1: Coverage and confidence set size for GMM simulations with $K = 5$ components for varying noise level σ^2 and sample size n. Top row: GMM in 2D with stochastic GMM clustering. Bottom row: GMM in 50D with spectral clustering that applies stochastic GMM clustering to the top K singular vectors. Algorithm 1 is compared with the split conformal clustering (CC) and naive cutoff methods.

with informative set size. Meanwhile, as noted in [11], the naive cutoff method results in larger sets because, by construction, it has to include less probable labels to achieve the target coverage. The bottom row shows the results in $\mathbb{R}^{50}$ with spectral clustering. The first two plots show that Algorithm 1 remains valid and informative for the largest variances $\sigma^2 \in \{2.3, 2.5\}$, while split CC becomes oversized and hence uninformative. The last two plots show that Algorithm 1 starts to achieve valid coverage with informative set size at $n = 1200$, while the set size of split CC remains large even at $n = 3000$ and starts to match that of Algorithm 1 at $n = 5000$. These results illustrate that Algorithm 1 can provide more informative uncertainty quantification for spectral clustering in high-dimensional settings.

Nonlinear Clusters We consider two nonlinear clustering examples based on three moons and two concentric circles. In both examples, each component lies on a nonlinear manifold, which is not linearly separable. We apply spectral clustering that applies stochastic GMM clustering to the top K singular vectors of the Laplacian of the mutual k-nearest neighbor graph (with k = 5% of n_{tr}). For these examples, the variance σ^2 denotes the spread (thickness) of the manifold around the ideal nonlinear structure. For varying σ^2 (with fixed $n = 600$ for moons, $n = 400$ for circles) or varying n (with fixed $\sigma^2 = 1.8$ for moons, $\sigma^2 = 1.05$ for circles), we compare the coverage and set size of Algorithm 1 and split CC.

The top row of Figure 2 visualizes the confidence sets produced by Algorithm 1 as heatmaps. Simulated points are overlaid with the heatmaps that color each point $x \in \mathbb{R}^2$ by a mixture of the colors corresponding to the labels in the confidence set $\hat{\mathcal{C}}(x)$. In both examples, we see that the sets become larger for larger σ^2 as the gap between the manifolds becomes smaller, making the clustering task more difficult and thus increasing the uncertainty. Meanwhile, the middle and bottom rows of Figure 2 show that both Algorithm 1 and split CC have almost perfect coverage with informative set size for sufficiently small σ^2, while the coverage drops for larger σ^2 as clustering becomes more challenging. Interestingly, Algorithm 1 maintains the informative set size for a wider range of σ^2 while still achieving valid coverage, but split CC fails to produce informative sets for larger σ^2. Also, we can see that both methods achieve asymptotically valid coverage with decreasing set size as n increases, but Algorithm 1 achieves this faster than split CC, as also observed in the spectral clustering simulations for the GMM in $\mathbb{R}^{50}$.

In summary, Algorithm 1 provides valid and more informative confidence sets for nonlinear clustering compared to split CC, confirming that the weighted conformal framework brings benefits to a wide range of clustering problems beyond the standard mixture model setting considered in [11].

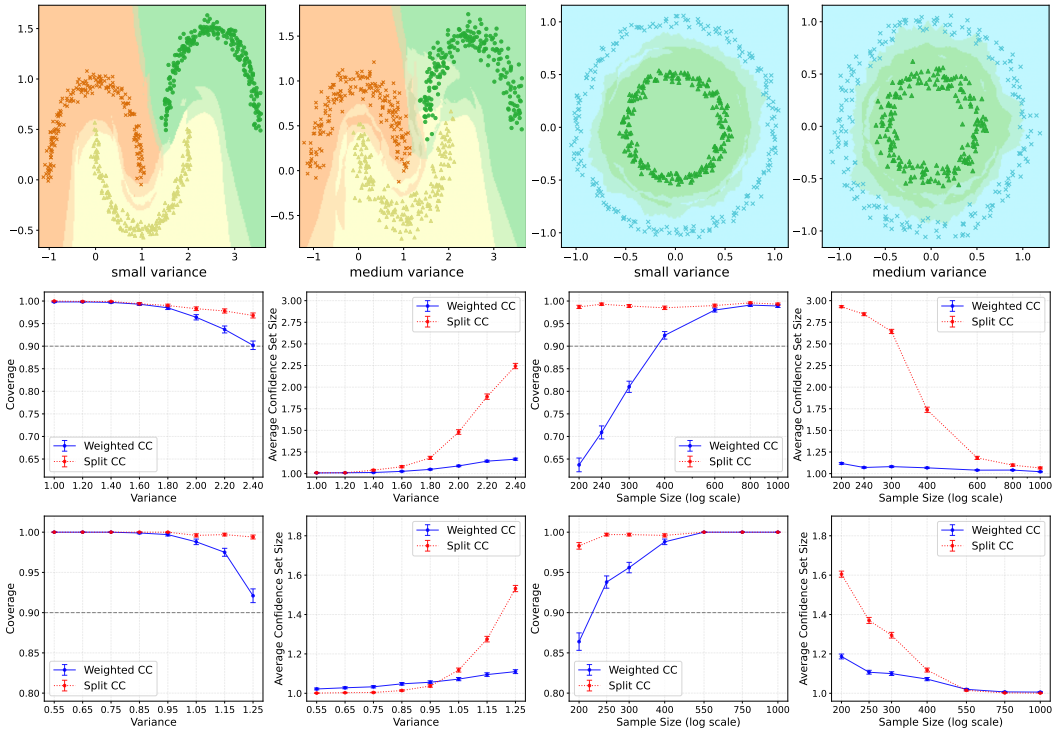


Figure 2: Confidence set heatmaps and coverage/set size for the nonlinear clustering simulations. Top row: confidence sets produced by Algorithm 1 for moons ($\sigma^2 \in \{1.0, 2.2\}$) and circles ($\sigma^2 \in \{0.65, 1.25\}$). Middle row: coverage and set size for the moon example for varying σ^2 and for varying n. Bottom row: coverage and set size for the circle example. Algorithm 1 and split CC are implemented with spectral clustering that applies stochastic GMM clustering to the top K singular vectors of the Laplacian of the mutual nearest neighbor graph.

3.2 MNIST Application with Deep Clustering

We apply Algorithm 1 with deep clustering to the MNIST dataset, which consists of images of handwritten digits (0–9) in $\mathbb{R}^{28\times 28}$. Deep clustering [32, 3, 34], which combines deep representation learning with clustering, has been shown to be effective for clustering high-dimensional data such as images. We first pretrain a convolutional autoencoder to learn a low-dimensional (10D) representation. We focus on 10,000 images not used for pretraining, which we split into training (4,500), calibration (4,500), and test (1,000) sets. For each test image, we produce a confidence set using Algorithm 1 with stochastic GMM clustering applied in the 10D representation space with $\alpha = 0.1$.

The left panel of Figure 3 plots the true digit labels for the training and calibration sets in the t-SNE space. The right panel visualizes the confidence sets for the test set, where each point is colored by its true label and marked by its set size (circle, triangle, square for size 1, 2, 3+). Most of them are singleton confidence sets (45.3%) and tend to be located near the centers of the clusters, while the rest with size 2 (30.4%) or 3+ (24.3%) tend to appear near the boundaries between clusters or in more ambiguous regions. Several ambiguous points with large confidence sets are shown with their corresponding images, which are indeed tricky to classify even for humans.

We can also check the coverage by using the true labels. To do so, we use 10-fold stratified cross-validation, where we calculate the coverage of the true labels of each fold (1,000 points) given the weighted conformal sets produced by the remaining 9 folds. The average coverage across folds is 0.872 with a standard deviation of 0.050, and the average set size across folds is 1.868 with a standard deviation of 0.064. Though slightly below the nominal level of 0.9, this shows that weighted conformal clustering can provide a reasonable coverage.

In summary, this application demonstrates that Algorithm 1 can provide informative uncertainty quantification with deep clustering for high-dimensional data, extending its utility beyond standard

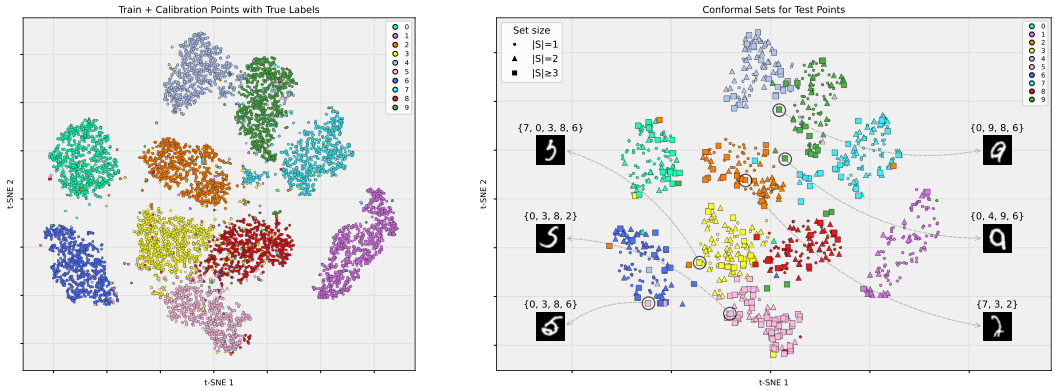


Figure 3: True labels and confidence sets for the MNIST dataset. Left: true digit labels for the training and calibration sets in the t-SNE space. Right: confidence sets produced by Algorithm 1 for the test set, visualized with different markers based on the set size and colored by the true digit labels. Several ambiguous points with large confidence sets are shown with their corresponding images.

mixture models and spectral clustering. All the experiments in this section are reproducible using the code available at `https://github.com/DataSlingers/Weighted_Conformal_Clustering`.

4 Discussion

This paper introduces weighted conformal clustering, an innovative framework that extends the conformal clustering approach of [11] by treating synthetic clustering labels as inducing a conditional label-distribution shift from the latent target labels. Using weighted conformal prediction beyond exchangeability, we develop a tractable algorithm with asymptotic coverage under consistent conditional-density estimation, and show empirically that it yields smaller confidence sets in high-dimensional and nonlinear clustering while matching existing methods in smooth mixture settings. This represents a substantive step forward for the emerging conformal clustering framework, with potential implications for conformal prediction in unsupervised learning beyond clustering. A key ingredient is the novel augmented calibration mechanism, which makes weighted conformal prediction computationally tractable while combining aspects of both full and split conformal procedures. This suggests augmented calibration as a broader methodological principle as a whole for more interesting applications in the future.

Acknowledgments and Disclosure of Funding

The authors acknowledge funding from NSF DMS-2516872.

References

[1] Anastasios N Angelopoulos, Rina Foygel Barber, and Stephen Bates. Theoretical foundations of conformal prediction. *arXiv:2411.11824*, 2024.

[2] Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p-values. *The Annals of Statistics*, 51(1):149–178, 2023.

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision*, 2018.

[4] Yiqun T Chen and Daniela M Witten. Selective inference for k-means clustering. *Journal of Machine Learning Research*, 24(152):1–41, 2023.

[5] Giovanni Cherubin, Ilia Nouretdinov, Alexander Gammerman, Roberto Jordaney, Zhi Wang, Davide Papini, and Lorenzo Cavallaro. Conformal clustering and its application to botnet traffic. In *International Symposium on Statistical Learning and Data Sciences*, 2015.

[6] David B Dahl, Devin J Johnson, and Peter Müller. Search algorithms and loss functions for Bayesian clustering. *Journal of Computational and Graphical Statistics*, 31(4):1189–1201, 2022.

[7] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[8] Lucy L Gao, Jacob Bien, and Daniela Witten. Selective inference for hierarchical clustering. *Journal of the American Statistical Association*, 119(545):332–342, 2024.

[9] Joeri Hofmans, Eva Ceulemans, Douglas Steinley, and Iven Van Mechelen. On the added value of bootstrap analysis for K-means clustering. *Journal of Classification*, 32(2):268–284, 2015.

[10] Hanwen Huang, Yufeng Liu, Ming Yuan, and JS Marron. Statistical significance of clustering using soft thresholding. *Journal of Computational and Graphical Statistics*, 24(4):975–993, 2015.

[11] YoonHaeng Hur, Anirban Nath, and Genevera Allen. Inference for clustering: Conformal sets for cluster labels. *arXiv:2604.03488*, 2026.

[12] M. K. Kerr and G. A. Churchill. Bootstrapping cluster analysis: Assessing the reliability of conclusions from microarray experiments. *Proceedings of the National Academy of Sciences*, 98(16):8961–8965, 2001.

[13] Bahareh Mohammadi Kiani. A conformalized density-based clustering analysis of malicious traffic for botnet detection. In *Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications*, 2020.

[14] Patrick K Kimes, Yufeng Liu, David Neil Hayes, and James Stephen Marron. Statistical significance for hierarchical clustering. *Biometrics*, 73(3):811–821, 2017.

[15] Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282, 2019.

[16] Junu Lee, Ilia Popov, and Zhimei Ren. Full-conformal novelty detection: A powerful and non-random approach. *arXiv:2501.02703*, 2025.

[17] Jing Lei. Classification with confidence. *Biometrika*, 101(4):755–769, 2014.

[18] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

[19] Ziyi Liang, Matteo Sesia, and Wenguang Sun. Integrative conformal p-values for out-of-distribution testing with labelled outliers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(3):671–693, 2024.

[20] Chuen-Ming Liu, Zhi-Ping Niu, and Kuan-Ting Liao. Mechanisms to improve clustering uncertain data with UKmeans. *Data & Knowledge Engineering*, 116:1–18, 2018.

[21] Yufeng Liu, David Neil Hayes, Andrew Nobel, and James Stephen Marron. Statistical significance of clustering for high-dimension, low-sample size data. *Journal of the American Statistical Association*, 103(483):1281–1293, 2008.

[22] Ilia Nouretdinov, James Gammerman, Matteo Fontana, and Daljit Rehal. Multi-level conformal clustering: A distribution-free technique for clustering and anomaly detection. *Neurocomputing*, 397:279–291, 2020.

[23] Aleksandr Podkopaev and Aaditya Ramdas. Distribution-free uncertainty quantification for classification under label shift. In *Uncertainty in Artificial Intelligence*, 2021.

[24] Yaniv Romano, Evan Patterson, and Emmanuel J. Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, 2019.

[25] Yaniv Romano, Matteo Sesia, and Emmanuel Candès. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing Systems*, 2020.

[26] Mauricio Sadinle, Jing Lei, and Larry Wasserman. Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 114(525):223–234, 2019.

[27] Hui Shen, Shankar Bhamidi, and Yufeng Liu. Statistical significance of clustering with multidimensional scaling. *Journal of Computational and Graphical Statistics*, 33(1):219–230, 2024.

[28] Wenwen Si, Sangdon Park, Insup Lee, Edgar Dobriban, and Osbert Bastani. PAC prediction sets under label shift. *arXiv:2310.12964*, 2023.

[29] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems*, 2019.

[30] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

[31] Sara Wade and Zoubin Ghahramani. Bayesian cluster analysis: Point estimation and credible balls (with discussion). *Bayesian Analysis*, 13(2):559–626, 2018.

[32] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, 2016.

[33] Young-Joo Yun and Rina Foygel Barber. Selective inference for clustering with unknown variance. *Electronic Journal of Statistics*, 17(2):1923–1946, 2023.

[34] Sheng Zhou, Hongjia Xu, Zhuonan Zheng, Jiawei Chen, Zhao Li, Jiajun Bu, Jia Wu, Xin Wang, Wenwu Zhu, and Martin Ester. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. *ACM Computing Surveys*, 57(3):1–38, 2024.

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: The abstract and introduction clearly state the formulation of the problem, the theoretical coverage guarantees (Theorems 1 and 2), and the empirical results.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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 limitations, specifically regarding the need for consistent conditional-density estimation and the implications of using hard clustering versus soft clustering, are discussed in Section 2.3.

   Guidelines:

   - The answer [N/A] 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: The assumptions are stated in the theorems in Section 2, and complete proofs are provided in Appendix B.

Guidelines:

- The answer [N/A] 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: Section 3 provides detailed descriptions of the data generating processes, clustering algorithms, sample sizes, dimensions, and hyperparameters.

   Guidelines:

   - The answer [N/A] 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: The code is available at `https://github.com/DataSlingers/Weighted_Conformal_Clustering`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

Answer: [Yes]

Justification: Detailed parameters including training/calibration/test splits, noise levels (σ^2), number of clusters (K), and conformity scores are reported in Section 3.

Guidelines:

- The answer [N/A] 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: The figures show empirical coverage and set sizes averaged over multiple independent trials to capture the variability and performance of the methods.

Guidelines:

- The answer [N/A] 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: [No]

   Justification: The experiments are standard clustering and conformal prediction tasks on modest-sized datasets that can be easily reproduced on a standard desktop computer. We partially address this in the appendix, reporting the wall-clock time and peak memory usage for a specific simulation setup.

   Guidelines:

   - The answer [N/A] 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: The paper proposes a methodological and theoretical framework and uses standard, non-sensitive datasets (like MNIST).

   Guidelines:

   - The answer [N/A] 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: [No]

    Justification: The paper focuses on theoretical and methodological foundations for uncertainty quantification in unsupervised learning, without foreseeable direct negative societal impacts.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release any data or models with a high risk for misuse.

    Guidelines:

    - The answer [N/A] 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: The MNIST dataset and existing methods (e.g., split conformal clustering) are properly cited in the text and bibliography.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not introduce new datasets or large-scale assets.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The research does not involve human subjects or crowdsourcing.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The research does not involve human subjects.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as a component of the core methodology.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.

A Theory of Weighted Conformal Prediction

Definition 2. For any measure μ on $\mathbb{R}$, let $\mathsf{Q}_\tau(\mu)$ be the τ-th quantile of μ defined by

$$\mathsf{Q}_\tau(\mu) = \inf\{t \in \mathbb{R} : \mu((-\infty, t]) \geq \tau\} \quad \forall \tau \in [0,1].$$

For $\{s_1, \ldots, s_n\} \subset \mathbb{R}$, we abbreviate $\mathsf{Q}_\tau\left(\frac{1}{n}\sum_{i=1}^n \delta_{s_i}\right)$ as $\mathsf{Q}_\tau(\{s_i\}_{i=1}^n)$.

A.1 Full Conformal Prediction with Weighting

We state the weighted conformal procedure, where $(X_1, Y_1), \ldots, (X_{n+1}, Y_{n+1})$ are from an arbitrary distribution P_{n+1} on $(\mathcal{X} \times \mathcal{Y})^{n+1}$, which may not be a product distribution. The high-level idea is explained in Section 7.5 of [1]. We review this here for completeness.

Recall that the full conformal prediction set can be rewritten as

$$C_n(x) = \left\{ y \in \mathcal{Y} : s((x,y); \mathcal{D}_n^{(x,y)}) \leq \mathsf{Q}_{1-\alpha}\left(\frac{\delta_{s((x,y);\mathcal{D}_n^{(x,y)})} + \sum_{i=1}^n \delta_{s((X_i,Y_i);\mathcal{D}_n^{(x,y)})}}{n+1} \right) \right\},$$

where $\mathcal{D}_n^{(x,y)} = \{(X_1, Y_1), \ldots, (X_n, Y_n), (x, y)\}$ and $s \colon \mathcal{X} \times \mathcal{Y} \times 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}$ is a conformity score function.[2] The full conformal prediction set above no longer satisfies the coverage guarantee because the exchangeability assumption is violated. Particularly, the score $s((X_{n+1}, Y_{n+1}); \mathcal{D}_n^{(X_{n+1},Y_{n+1})})$ may tend to be larger than the scores of the training data because of distribution shift, and hence the prediction set will tend to under-cover.

Let $Z_1, \ldots, Z_n, Z_{n+1} \in \mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ have joint density f on $\mathcal{Z}^{n+1}$, without assuming exchangeability, and let $\hat{P}_{n+1} = \{Z_1, \ldots, Z_{n+1}\}$ denote the unordered empirical measure. Conditional on $\hat{P}_{n+1}$, the identity of the test point is determined by the relative likelihoods of the different orderings of these $n+1$ observations under f. Now, fix a candidate label y, and form the augmented sample by setting $Z_i^y = Z_i$ for $i \in [n]$ and $Z_{n+1}^y = (X_{n+1}, y)$. The following theorem states the general weighted conformal procedure and its marginal validity.

Theorem 3. *Suppose $(X_1, Y_1), \ldots, (X_{n+1}, Y_{n+1})$ are drawn from some distribution P_{n+1} on $(\mathcal{X} \times \mathcal{Y})^{n+1}$ whose probability density function is $f \colon (\mathcal{X} \times \mathcal{Y})^{n+1} \to \mathbb{R}_+$. For $(x, y) \in \mathcal{X} \times \mathcal{Y}$, let $Z_i^{(x,y)} := (X_i, Y_i)$ for $i \in [n]$ and $Z_{n+1}^{(x,y)} := (x, y)$, and for any permutation σ of $[n+1]$, define*

$$w_\sigma^{(x,y)} := \frac{f(Z_{\sigma(1)}^{(x,y)}, \ldots, Z_{\sigma(n+1)}^{(x,y)})}{\sum_{\sigma' \in \mathcal{S}_{n+1}} f(Z_{\sigma'(1)}^{(x,y)}, \ldots, Z_{\sigma'(n+1)}^{(x,y)})}.$$

Let $s \colon \mathcal{X} \times \mathcal{Y} \times 2^{\mathcal{X}\times\mathcal{Y}} \to \mathbb{R}$ be a fixed function, where the second argument of s is an unordered subset of $\mathcal{X} \times \mathcal{Y}$. Define

$$C_n(x) = \left\{ y \in \mathcal{Y} : s((x,y); \mathcal{D}_n^{(x,y)}) \leq \mathsf{Q}_{1-\alpha}\left(\sum_{\sigma \in \mathcal{S}_{n+1}} w_\sigma^{(x,y)} \delta_{s(Z_{\sigma(n+1)}^{(x,y)}; \mathcal{D}_n^{(x,y)})} \right) \right\},$$

where $\mathcal{D}_n^{(x,y)} = \{(X_1, Y_1), \ldots, (X_n, Y_n), (x, y)\}$. Then, we have

$$\mathbb{P}(Y_{n+1} \in C_n(X_{n+1})) \geq 1 - \alpha.$$

Proof. Let $\mathcal{D}_{n+1} = \mathcal{D}_n^{(X_{n+1},Y_{n+1})}$. Let $S_i = s((X_i, Y_i); \mathcal{D}_{n+1})$ for $i = 1, \ldots, n+1$. Then, by definition, we have

$$Y_{n+1} \in C_n(X_{n+1}) \Leftrightarrow S_{n+1} \leq \mathsf{Q}_{1-\alpha}\left(\sum_{\sigma \in \mathcal{S}_{n+1}} w_\sigma \delta_{S_{\sigma(n+1)}} \right) =: q,$$

where $w_\sigma = w_\sigma^{(X_{n+1},Y_{n+1})}$. By Lemma 2, we have

$$S_{n+1} \,\Big|\, \frac{1}{n+1}\sum_{i=1}^{n+1} \delta_{(X_i,Y_i)} \sim \sum_{\sigma \in \mathcal{S}_{n+1}} w_\sigma \delta_{S_{\sigma(n+1)}}.$$

[2] The second argument of s is an unordered subset of $\mathcal{X} \times \mathcal{Y}$, meaning that s ignores the order of its arguments.

Hence,

$$\mathbb{P}\left(S_{n+1} \le q \,|\, \frac{1}{n+1}\sum_{i=1}^{n+1}\delta_{(X_i,Y_i)}\right) = \sum_{\sigma\in\mathcal{S}_{n+1}} w_\sigma 1\{S_{\sigma(n+1)} \le q\} \ge 1-\alpha,$$

where the last inequality is due to Lemma 1. By the law of total probability, we have

$$\mathbb{P}(Y_{n+1} \in C_n(X_{n+1})) \ge 1-\alpha.$$

□

A.2 Technical Lemmas

Lemma 1. *For any $z_1,\dots,z_n \in \mathbb{R}$, weights $w_1,\dots,w_n \ge 0$ with $\sum_{i=1}^n w_i = 1$, and $\tau \in [0,1]$, we have*

$$\sum_{i=1}^n w_i 1\left\{z_i \le \mathsf{Q}_\tau\left(\sum_{j=1}^n w_j\delta_{z_j}\right)\right\} \ge \tau.$$

Suppose we swap z_n with any $z \in \mathbb{R}\cup\{\infty\}$ such that $z \ge \max_{i\in[n]} z_i$. Then, we have

$$z_n \le \mathsf{Q}_\tau\left(\sum_{i=1}^n w_i\delta_{z_i}\right) \quad\Leftrightarrow\quad z_n \le \mathsf{Q}_\tau\left(\sum_{i=1}^{n-1} w_i\delta_{z_i} + w_n\delta_z\right).$$

Proof. Let F be the CDF of the measure $\sum_{i=1}^n w_i\delta_{z_i}$, i.e., $F(t) = \sum_{i=1}^n w_i 1\{z_i \le t\}$ for $t\in\mathbb{R}$. By definition, we have $\mathsf{Q}_\tau\left(\sum_{i=1}^n w_i\delta_{z_i}\right) = \inf\{t\in\mathbb{R} : F(t) \ge \tau\}$. By the right-continuity of F, we can deduce that $F\left(\mathsf{Q}_\tau\left(\sum_{i=1}^n w_i\delta_{z_i}\right)\right) \ge \tau$, which is exactly the desired result. Next, for $z \ge \max_{i\in[n]} z_i$, we have

$$q := \mathsf{Q}_\tau\left(\sum_{i=1}^n w_i\delta_{z_i}\right) \le \mathsf{Q}_\tau\left(\sum_{i=1}^{n-1} w_i\delta_{z_i} + w_n\delta_z\right) =: q^z.$$

Hence, $z_n \le q$ implies $z_n \le q^z$. Conversely, suppose $z_n \le q^z$. Let F^z be the CDF of the measure $\sum_{i=1}^{n-1} w_i\delta_{z_i} + w_n\delta_z$. Since $z_n \le q^z$, we must have $F^z(t) < \tau$ for any $t < z_n$; also, we have $F(t) = F^z(t)$ for any $t < z_n$ because $z_n \le z$ implies

$$F(t) = \sum_{i=1}^{n-1} w_i 1\{z_i \le t\} + w_n 1\{z_n \le t\} = \sum_{i=1}^{n-1} w_i 1\{z_i \le t\},$$
$$F^z(t) = \sum_{i=1}^{n-1} w_i 1\{z_i \le t\} + w_n 1\{z \le t\} = \sum_{i=1}^{n-1} w_i 1\{z_i \le t\}.$$

Hence, for any $t < z_n$, we have $F(t) = F^z(t) < \tau$, which implies $z_n \le q$. □

Lemma 2. *Suppose $\mathcal{Z}$ is a finite set. Let P_n be a distribution on $\mathcal{Z}^n$ with the probability mass function $f\colon \mathcal{Z}^n \to \mathbb{R}_+$. For $(Z_1,\dots,Z_n) \sim P_n$, we have*

$$Z_i \,|\, \{Z_1,\dots,Z_n\} = \{z_1,\dots,z_n\} \sim \frac{1}{\sum_{\sigma\in\mathcal{S}_n} f(z_{\sigma(1)},\dots,z_{\sigma(n)})}\sum_{\sigma\in\mathcal{S}_n} f(z_{\sigma(1)},\dots,z_{\sigma(n)})\delta_{z_{\sigma(i)}}$$

for any $i\in[n]$ and $z_1,\dots,z_n\in\mathcal{Z}$. If we have a function $s\colon \mathcal{Z}\times 2^{\mathcal{Z}} \to S$, where S is some set, then we also have

$$s(Z_i;\{Z_1,\dots,Z_n\}) \,|\, \{Z_1,\dots,Z_n\} = \{z_1,\dots,z_n\}$$
$$\sim \frac{1}{\sum_{\sigma\in\mathcal{S}_n} f(z_{\sigma(1)},\dots,z_{\sigma(n)})}\sum_{\sigma\in\mathcal{S}_n} f(z_{\sigma(1)},\dots,z_{\sigma(n)})\delta_{s(z_{\sigma(i)};\{z_1,\dots,z_n\})}.$$

If $P_n = P\otimes\cdots\otimes P\otimes Q$ for some distributions P, Q on $\mathcal{Z}$ whose probability mass functions are $p, q\colon \mathcal{Z}\to\mathbb{R}_+$, respectively, then we have

$$Z_n \,|\, \{Z_1,\dots,Z_n\} = \{z_1,\dots,z_n\} \sim \frac{1}{\sum_{i=1}^n \frac{q\{z_i\}}{p\{z_i\}}}\sum_{i=1}^n \frac{q\{z_i\}}{p\{z_i\}}\delta_{z_i}$$

and

$$s(Z_i; \{Z_1, \ldots, Z_n\}) \,|\, \{Z_1, \ldots, Z_n\} = \{z_1, \ldots, z_n\} \sim \frac{1}{\sum_{i=1}^n \frac{q\{z_i\}}{p\{z_i\}}} \sum_{i=1}^n \frac{q\{z_i\}}{p\{z_i\}} \delta_{s(z_{\sigma(i)}; \{z_1, \ldots, z_n\})}.$$

Proof. Without loss of generality, we can assume $i = n$. It suffices to show that for any $A \subset \mathcal{Z}$, we have

$$\mathbb{P}(Z_n \in A \,|\, \{Z_1, \ldots, Z_n\} = \{z_1, \ldots, z_n\}) = \frac{\sum_{\sigma \in \mathcal{S}_n} f(z_{\sigma(1)}, \ldots, z_{\sigma(n)}) 1\{z_{\sigma(n)} \in A\}}{\sum_{\sigma \in \mathcal{S}_n} f(z_{\sigma(1)}, \ldots, z_{\sigma(n)})}.$$

Clearly,

$$\begin{aligned}\mathbb{P}(\{Z_1, \ldots, Z_n\} = \{z_1, \ldots, z_n\}) &= \sum_{\sigma \in \mathcal{S}_n} \mathbb{P}((Z_1, \ldots, Z_n) = (z_{\sigma(1)}, \ldots, z_{\sigma(n)})) \\ &= \sum_{\sigma \in \mathcal{S}_n} f(z_{\sigma(1)}, \ldots, z_{\sigma(n)}).\end{aligned}$$

Similarly,

$$\begin{aligned}&\mathbb{P}(Z_n \in A, \{Z_1, \ldots, Z_n\} = \{z_1, \ldots, z_n\}) \\ &\quad = \sum_{\sigma \in \mathcal{S}_n} \mathbb{P}((Z_1, \ldots, Z_n) = (z_{\sigma(1)}, \ldots, z_{\sigma(n)}), z_{\sigma(n)} \in A) \\ &\quad = \sum_{\sigma \in \mathcal{S}_n} f(z_{\sigma(1)}, \ldots, z_{\sigma(n)}) 1\{z_{\sigma(n)} \in A\}.\end{aligned}$$

Hence, we prove the desired claim. Meanwhile, for a function $s \colon \mathcal{Z} \times 2^{\mathcal{Z}} \to S$, where S is some set, we have for any $B \subset S$,

$$\begin{aligned}&\mathbb{P}(s(Z_n; \{Z_1, \ldots, Z_n\}) \in B, \{Z_1, \ldots, Z_n\} = \{z_1, \ldots, z_n\}) \\ &\quad = \sum_{\sigma \in \mathcal{S}_n} \mathbb{P}((Z_1, \ldots, Z_n) = (z_{\sigma(1)}, \ldots, z_{\sigma(n)}), s(z_{\sigma(n)}; \{z_1, \ldots, z_n\}) \in B) \\ &\quad = \sum_{\sigma \in \mathcal{S}_n} f(z_{\sigma(1)}, \ldots, z_{\sigma(n)}) 1\{s(z_{\sigma(n)}; \{z_1, \ldots, z_n\}) \in B\}.\end{aligned}$$

Hence, we prove the second claim. Lastly, when $P_n = P \otimes \cdots \otimes P \otimes Q$, we have

$$f(z_1, \ldots, z_n) = \prod_{i=1}^{n-1} p\{z_i\} \cdot q\{z_n\} = \left(\prod_{i=1}^n p\{z_i\} \right) \cdot \frac{q\{z_n\}}{p\{z_n\}}.$$

Hence, for any $\sigma \in \mathcal{S}_n$, we have

$$f(z_{\sigma(1)}, \ldots, z_{\sigma(n)}) = \left(\prod_{i=1}^n p\{z_i\} \right) \cdot \frac{q\{z_{\sigma(n)}\}}{p\{z_{\sigma(n)}\}}.$$

Therefore,

$$\begin{aligned}\sum_{\sigma \in \mathcal{S}_n} f(z_{\sigma(1)}, \ldots, z_{\sigma(n)}) 1\{z_{\sigma(n)} \in A\} &= \left(\prod_{i=1}^n p\{z_i\} \right) \cdot \sum_{\sigma \in \mathcal{S}_n} \frac{q\{z_{\sigma(n)}\}}{p\{z_{\sigma(n)}\}} 1\{z_{\sigma(n)} \in A\} \\ &= \left(\prod_{i=1}^n p\{z_i\} \right) \cdot \sum_{i=1}^n \sum_{\sigma \in \mathcal{S}_n, \sigma(n) = i} \frac{q\{z_i\}}{p\{z_i\}} 1\{z_i \in A\} \\ &= (n-1)! \left(\prod_{i=1}^n p\{z_i\} \right) \cdot \sum_{i=1}^n \frac{q\{z_i\}}{p\{z_i\}} 1\{z_i \in A\}.\end{aligned}$$

Similarly,

$$\sum_{\sigma \in \mathcal{S}_n} f(z_{\sigma(1)}, \ldots, z_{\sigma(n)}) = (n-1)! \left(\prod_{i=1}^n p\{z_i\} \right) \cdot \sum_{i=1}^n \frac{q\{z_i\}}{p\{z_i\}}.$$

Hence,

$$\mathbb{P}(Z_n \in A \,|\, \{Z_1, \ldots, Z_n\} = \{z_1, \ldots, z_n\}) = \frac{\sum_{i=1}^n \frac{q\{z_i\}}{p\{z_i\}} 1\{z_i \in A\}}{\sum_{i=1}^n \frac{q\{z_i\}}{p\{z_i\}}},$$

which proves

$$Z_n \,|\, \{Z_1, \ldots, Z_n\} = \{z_1, \ldots, z_n\} \sim \frac{1}{\sum_{i=1}^n \frac{q\{z_i\}}{p\{z_i\}}} \sum_{i=1}^n \frac{q\{z_i\}}{p\{z_i\}} \delta_{z_i}.$$

□

Remark 4. Lemma 2 can be extended to the case where $\mathcal{Z}$ is an infinite set with f being a probability density function. In this case, instead of conditioning on the event $\{Z_1, \ldots, Z_n\} = \{z_1, \ldots, z_n\}$, we should think of the regular conditional distribution of Z_i given a random measure $\frac{1}{n}\sum_{j=1}^n \delta_{Z_j}$ on $\mathcal{Z}$. Also, the last part of Lemma 2 still holds with $\frac{\mathrm{d}Q}{\mathrm{d}P}(z_i)$ in the place of $\frac{q\{z_i\}}{p\{z_i\}}$. This is formally stated as Proposition 7.6 of [1].

B Proofs

B.1 Proof of Theorem 1

Proof. By the definition of the alignment $\hat{\sigma}_o^*$, we have

$$\mathbb{P}\left(\hat{\sigma}_o^*(Y_{n+1}^*) \in \hat{\mathcal{C}}(X_{n+1}) \mid \hat{\mathcal{C}}\right) \geq \mathbb{P}\left(Y_{n+1}^* \in \hat{\mathcal{C}}(X_{n+1}) \mid \hat{\mathcal{C}}\right).$$

By taking the expectation, we have $\mathbb{P}(\hat{\sigma}_o^*(Y_{n+1}^*) \in \hat{\mathcal{C}}(X_{n+1})) \geq \mathbb{P}(Y_{n+1}^* \in \hat{\mathcal{C}}(X_{n+1}))$. Hence, it suffices to show $\mathbb{P}(Y_{n+1}^* \in \hat{\mathcal{C}}(X_{n+1})) \geq 1 - \alpha$.

For notational convenience, we prove this for a pretrained version of Algorithm 0, where we assume that $\mathcal{I}_{ca} = \{1, \ldots, n\}$, namely, take the whole input as the calibration, while treating the training split and the soft classifier trained on it as fixed independent quantities. Accordingly, from now on, we will condition on the training split.

Now, let

$$Z_i := (X_i, Y_i), \;\; i = 1, \ldots, n, \qquad \text{and} \qquad Z_{n+1} := (X_{n+1}, Y_{n+1}^*).$$

We will derive the oracle weights appearing in Algorithm 0 from the general weighted conformal prediction theory presented in Appendix A. Note that $X_1, \ldots, X_{n+1}$ are i.i.d. from P_X^*, and conditional on the calibration covariates, the synthetic calibration labels are independent with

$$Y_j \mid X_1, \ldots, X_n \sim \mathrm{Cat}\big(\mathcal{A}_{\mathcal{D}_{ca}}(X_j)\big), \qquad j = 1, \ldots, n,$$

while the test label Y_{n+1}^* is drawn from the true conditional law $P_{Y|X}^*(\cdot \mid X_{n+1})$. Then, the joint density of the augmented sample $(Z_1, \ldots, Z_{n+1})$ at $(z_1, \ldots, z_{n+1}) = ((x_1, y_1), \ldots, (x_{n+1}, y_{n+1}))$ is

$$f(z_1, \ldots, z_{n+1}) = \left(\prod_{k=1}^{n+1} p_X^*(x_k)\right) \left(\prod_{j=1}^{n} \left[\mathcal{A}_{\{x_1, \ldots, x_n\}}(x_j)\right]_{y_j}\right) p_{Y|X}^*(y_{n+1} \mid x_{n+1}). \tag{6}$$

To compute the conformal weights, consider the permutations such that the i-th observation occupies the test position, namely, $\sigma \in \mathcal{S}_{n+1}$ such that $\sigma(n+1) = i$. Under this permutation, the point (x_i, y_i) is treated as the test point, while the remaining n covariates are $\{x_{\sigma(1)}, \ldots, x_{\sigma(n)}\}$. Thus, substituting the permuted indices into the joint density gives

$$f(z_{\sigma(1)}, \ldots, z_{\sigma(n+1)}) = \left(\prod_{k=1}^{n+1} p_X^*(x_{\sigma(k)})\right) \left(\prod_{j=1}^{n} \left[\mathcal{A}_{\{x_{\sigma(1)}, \ldots, x_{\sigma(n)}\}}(x_{\sigma(j)})\right]_{y_{\sigma(j)}}\right) p_{Y|X}^*(y_i \mid x_i).$$

Let $\{x_{\sigma(1)}, \ldots, x_{\sigma(n)}\} = \{x_1, \ldots, x_{n+1}\} \setminus \{x_i\} =: \mathcal{D}_{n+1}^{-i}$. Then,

$$\prod_{j=1}^{n} \left[\mathcal{A}_{\{x_{\sigma(1)}, \ldots, x_{\sigma(n)}\}}(x_{\sigma(j)})\right]_{y_{\sigma(j)}} = \prod_{j \in [n+1] \setminus \{i\}} \left[\mathcal{A}_{\mathcal{D}_{n+1}^{-i}}(x_j)\right]_{y_j}.$$

Also, note that $\prod_{k=1}^{n+1} p_X^*(x_{\sigma(k)}) = \prod_{k=1}^{n+1} p_X^*(x_k)$. Thus, we have

$$f(z_{\sigma(1)}, \ldots, z_{\sigma(n+1)}) = \left(\prod_{k=1}^{n+1} p_X^*(x_k) \right) p_{Y|X}^*(y_i \mid x_i) \prod_{j \in [n+1] \setminus \{i\}} \left[\mathcal{A}_{\mathcal{D}_{n+1}^{-i}}(x_j) \right]_{y_j}. \tag{7}$$

Observe that the right-hand side of (7) depends only on the index i, and not on the particular choice of permutation σ satisfying $\sigma(n+1) = i$. Since there are exactly $n!$ such permutations, this yields the unnormalized weights as

$$\begin{aligned} \tilde{w}_i &:= \sum_{\sigma \in \mathcal{S}_{n+1} : \sigma(n+1) = i} f(z_{\sigma(1)}, \ldots, z_{\sigma(n+1)}) \\ &= n! \left(\prod_{k=1}^{n+1} p_X^*(x_k) \right) p_{Y|X}^*(y_i \mid x_i) \prod_{j \in [n+1] \setminus \{i\}} \left[\mathcal{A}_{\mathcal{D}_{n+1}^{-i}}(x_j) \right]_{y_j}, \end{aligned} \tag{8}$$

for $i = 1, \ldots, n+1$. Now, replacing (x_i, y_i)'s with (X_i, Y_i)'s and (x_{n+1}, y_{n+1}) with $(X_{n+1}, y) = (x, y)$, we obtain the unnormalized weights in (2)-(3). Now, we can apply Theorem 3 to show that the probability of $Y_{n+1}^* \in \hat{\mathcal{C}}(X_{n+1})$ conditional on the training split is at least $1 - \alpha$. By the tower property, we complete the proof. □

Remark 5. As we can see from the proof, the key is to apply Theorem 3 after calculating the weights. This is possible because of the scores defined in Algorithm 0. If we included the alignment step as in Algorithm 1 of [11], we would end up with alignment-dependent scores, which we cannot apply Theorem 3 to.

B.2 Proof of Theorem 2

Proof. As in the proof of Theorem 1, we have $\mathbb{P}(\hat{\sigma}_o^*(Y_{n+1}^*) \in \hat{\mathcal{C}}(X_{n+1})) \geq \mathbb{P}(Y_{n+1}^* \in \hat{\mathcal{C}}(X_{n+1}))$. Hence, we derive a lower bound on $\mathbb{P}(Y_{n+1}^* \in \hat{\mathcal{C}}(X_{n+1}))$. Also, as in the proof of Theorem 1, we assume that $\mathcal{I}_{ca} = \{1, \ldots, n\}$, treat the training split-related objects as fixed independent quantities, and consider the conditional probability given the training split.

We first consider an auxiliary data-generating process in which the test label is drawn from the estimated conditional law $\widehat{P}_{Y|X}$ having the density $\hat{p}_{Y|X}$, namely,

$$(X_{n+1}, \tilde{Y}_{n+1}) \sim P_X^* \times \widehat{P}_{Y|X},$$

independently of the calibration data. The calibration labels are still generated according to the augmented clustering mechanism. Thus, conditional on $\mathcal{D}_{aug}$, we have

$$Y_j \mid X_1, \ldots, X_n, X_{n+1} \sim \mathrm{Cat}\big(\mathcal{A}_{\mathcal{D}_{aug}}(X_j)\big), \qquad j = 1, \ldots, n.$$

Under this auxiliary setup, let us derive the joint density of the augmented sample $(Z_1, \ldots, Z_{n+1}) := ((X_1, Y_1), \ldots, (X_{n+1}, \hat{\sigma}^{-1}(\tilde{Y}_{n+1}))$. Here, notice that $\hat{\sigma}$ in Step 5 of Algorithm 1 is symmetric with respect to the ordering of $X_1, \ldots, X_{n+1}$. In other words, $\hat{\sigma} = \rho(X_1, \ldots, X_{n+1})$ for a deterministic symmetric function ρ. The density of $(Z_1, \ldots, Z_{n+1})$ at $(z_1, \ldots, z_{n+1}) = ((x_1, y_1), \ldots, (x_{n+1}, y_{n+1}))$ is

$$f(z_1, \ldots, z_{n+1}) := \left(\prod_{k=1}^{n+1} p_X^*(x_k) \right) \left(\prod_{j=1}^{n} [\mathcal{A}_{\mathcal{D}}(x_j)]_{y_j} \right) \hat{p}_{Y|X}(\hat{\sigma}(y_{n+1}) \mid x_{n+1}),$$

where $\mathcal{D} := \{x_1, \ldots, x_{n+1}\}$.

To compute the conformal weights, consider the permutations such that the i-th observation occupies the test position i.e. $\sigma \in \mathcal{S}_{n+1}$ such that $\sigma(n+1) = i$. Under this permutation, the point (x_i, y_i) is treated as the test point, while the remaining n covariates form the calibration set. Since $\{x_{\sigma(1)}, \ldots, x_{\sigma(n+1)}\} = \mathcal{D}$ and $\prod_{k=1}^{n+1} p_X^*(x_{\sigma(k)}) = \prod_{k=1}^{n+1} p_X^*(x_k)$, we have for every σ satisfying $\sigma(n+1) = i$,

$$f(z_{\sigma(1)}, \ldots, z_{\sigma(n+1)}) = \left(\prod_{k=1}^{n+1} p_X^*(x_k) \right) \left(\prod_{j \in [n+1] \setminus \{i\}} [\mathcal{A}_{\mathcal{D}}(x_j)]_{y_j} \right) \hat{p}_{Y|X}(\hat{\sigma}(y_i) \mid x_i). \tag{9}$$

Now observe that the right-hand side of (9) depends only on the index i, and not on the particular permutation σ. Since there are exactly $n!$ permutations in $\mathcal{S}_{n+1}$ satisfying $\sigma(n+1) = i$, we have

$$\begin{aligned}\tilde{w}_i &:= \sum_{\sigma \in \mathcal{S}_{n+1}: \sigma(n+1)=i} f(z_{\sigma(1)}, \ldots, z_{\sigma(n+1)}) \\ &= n! \left(\prod_{k=1}^{n+1} p_X^*(x_k) \right) \hat{p}_{Y|X}(\hat{\sigma}(y_i) \mid x_i) \prod_{j \in [n+1] \setminus \{i\}} \left[\mathcal{A}_{\mathcal{D}}(x_j) \right]_{y_j} \\ &= n! \left(\prod_{k=1}^{n+1} p_X^*(x_k) \right) \prod_{j=1}^{n+1} \left[\mathcal{A}_{\mathcal{D}}(x_j) \right]_{y_j} \frac{\hat{p}_{Y|X}(\hat{\sigma}(y_i) \mid x_i)}{\left[\mathcal{A}_{\mathcal{D}}(x_i) \right]_{y_i}}\end{aligned}$$

Now, replacing (x_i, y_i)'s with (X_i, Y_i)'s and (x_{n+1}, y_{n+1}) with (X_{n+1}, y), which replaces $\mathcal{D}$ with $\mathcal{D}_{aug}$, we have

$$\hat{w}_i \propto \frac{\hat{p}_{Y|X}(\hat{\sigma}(Y_i) \mid X_i)}{\left[\mathcal{A}_{\mathcal{D}_{aug}}(X_i) \right]_{Y_i}} \quad \text{for } i \in \mathcal{I}_{ca}, \qquad \tilde{w}(y) \propto \frac{\hat{p}_{Y|X}(\hat{\sigma}(y) \mid x)}{\left[\mathcal{A}_{\mathcal{D}_{aug}}(x) \right]_{y}}. \tag{10}$$

Note that the weight $\tilde{w}(y)$ for the candidate label differs from $\hat{w}(y)$ in (5); in fact, we have $\tilde{w}(y) = \hat{w}(\hat{\sigma}(y))$. Therefore, once we normalize the weights in (10), we have $\{w_i(\hat{\sigma}(y))\}_{i \in \mathcal{I}_{ca}} \cup \{w(\hat{\sigma}(y))\}$. Now, we can apply Theorem 3 to $(Z_1, \ldots, Z_{n+1}) := ((X_1, Y_1), \ldots, (X_{n+1}, \hat{\sigma}^{-1}(\tilde{Y}_{n+1}))$ with the weights $\{w_i(\hat{\sigma}(y))\}_{i \in \mathcal{I}_{ca}} \cup \{w(\hat{\sigma}(y))\}$ and the scores

$$s((X_1, \hat{\sigma}(Y_1)); \hat{\pi}), \ldots, s((X_n, \hat{\sigma}(Y_n)); \hat{\pi}), s((X_{n+1}, \hat{\sigma}(\hat{\sigma}^{-1}(\tilde{Y}_{n+1}))); \hat{\pi}),$$

which yield

$$\mathbb{P}\left(\hat{\sigma}^{-1}(\tilde{Y}_{n+1}) \in \tilde{\mathcal{C}}(X_{n+1}) \right) \geq 1 - \alpha, \tag{11}$$

where the set $\tilde{\mathcal{C}}$ is defined based on Theorem 3 as follows:

$$\tilde{\mathcal{C}}(x) = \left\{ y \in [K] : s((x, \hat{\sigma}(y)); \hat{\pi}) \leq \mathsf{Q}_{1-\alpha}\left(\textstyle\sum_{i \in \mathcal{I}_{ca}} w_i(\hat{\sigma}(y))\, \delta_{S_i} + w(\hat{\sigma}(y)) \delta_{\infty} \right) \right\}.$$

Note that the set $\hat{\mathcal{C}}(x)$ in Algorithm 1 satisfies $\hat{\mathcal{C}}(x) = \{y \in [K] : \hat{\sigma}^{-1}(y) \in \tilde{\mathcal{C}}(x)\}$. Hence, (11) is equivalent to

$$\mathbb{P}\left(\tilde{Y}_{n+1} \in \hat{\mathcal{C}}(X_{n+1}) \right) \geq 1 - \alpha. \tag{12}$$

Now, by the definition of total variation distance,

$$\left| \mathbb{P}\left(\tilde{Y}_{n+1} \in \hat{\mathcal{C}}(X_{n+1}) \right) - \mathbb{P}\left(Y_{n+1}^* \in \hat{\mathcal{C}}(X_{n+1}) \right) \right| \leq d_{\mathrm{TV}}\left(P_X^* \times \widehat{P}_{Y|X}, P_X^* \times P_{Y|X}^* \right). \tag{13}$$

Combining (12) and (13) gives

$$\begin{aligned}\mathbb{P}\left(Y_{n+1}^* \in \hat{\mathcal{C}}(X_{n+1}) \right) &\geq 1 - \alpha - d_{\mathrm{TV}}\left(P_X^* \times \widehat{P}_{Y|X}, P_X^* \times P_{Y|X}^* \right) \\ &\geq 1 - \alpha - \int d_{\mathrm{TV}}\left(\widehat{P}_{Y|X}(\cdot \mid x), P_{Y|X}^*(\cdot \mid x) \right) dP_X^*(x) \\ &= 1 - \alpha - \mathbb{E}_{X \sim P_X^*}\left[d_{\mathrm{TV}}\left(\widehat{P}_{Y|X}(\cdot \mid X), P_{Y|X}^*(\cdot \mid X) \right) \right]\end{aligned} \tag{14}$$

where the second inequality is from Lemma 2 of [11]. Note that we have considered the conditional probability given the training split so far. Now, by taking expectation in (14) over the randomness in the training split, we obtain

$$\mathbb{P}\left(Y_{n+1}^* \in \hat{\mathcal{C}}(X_{n+1}) \right) \geq 1 - \alpha - \mathbb{E}_{tr}\mathbb{E}_{X \sim P_X^*}\left[d_{\mathrm{TV}}\left(\widehat{P}_{Y|X}(\cdot \mid X), P_{Y|X}^*(\cdot \mid X) \right) \right].$$

Note that we have started the proof by observing $\mathbb{P}(\hat{\sigma}_o^*(Y_{n+1}^*) \in \hat{\mathcal{C}}(X_{n+1})) \geq \mathbb{P}(Y_{n+1}^* \in \hat{\mathcal{C}}(X_{n+1}))$. In fact, we have $\mathbb{P}(\hat{\sigma}_o^*(Y_{n+1}^*) \in \hat{\mathcal{C}}(X_{n+1})) \geq \mathbb{P}(\sigma(Y_{n+1}^*) \in \hat{\mathcal{C}}(X_{n+1}))$ for any permutation σ. Let $\tilde{Y}_{n+1}^* = \sigma(Y_{n+1}^*)$ so that the conditional distribution $\tilde{Y}_{n+1}^* \mid X_{n+1}$ follows $\sigma \circ P_{Y|X}^*(\cdot \mid X_{n+1})$. Then, by repeating the proof above, one can deduce from (13) that

$$\left| \mathbb{P}\left(\tilde{Y}_{n+1} \in \hat{\mathcal{C}}(X_{n+1}) \right) - \mathbb{P}\left(\tilde{Y}_{n+1}^* \in \hat{\mathcal{C}}(X_{n+1}) \right) \right| \leq d_{\mathrm{TV}}\left(P_X^* \times \widehat{P}_{Y|X}, P_X^* \times (\sigma \circ P_{Y|X}^*) \right).$$

Hence, we have

$$\mathbb{P}\left(\tilde{Y}^*_{n+1} \in \hat{\mathcal{C}}(X_{n+1})\right) \geq 1 - \alpha - \mathbb{E}_{tr}\mathbb{E}_{X\sim P^*_X}\left[d_{\mathrm{TV}}\left(\widehat{P}_{Y|X}(\cdot \mid X), \sigma \circ P^*_{Y|X}(\cdot \mid X)\right)\right].$$

Therefore,

$$\begin{aligned}\mathbb{P}(\hat{\sigma}^*_o(Y^*_{n+1}) \in \hat{\mathcal{C}}(X_{n+1})) &\geq \max_{\sigma\in\mathcal{S}_K} \mathbb{P}(\sigma(Y^*_{n+1}) \in \hat{\mathcal{C}}(X_{n+1})) \\ &\geq 1 - \alpha - \min_{\sigma\in\mathcal{S}_K} \mathbb{E}_{tr}\mathbb{E}_{X\sim P^*_X}\left[d_{\mathrm{TV}}\left(\widehat{P}_{Y|X}(\cdot \mid X), \sigma \circ P^*_{Y|X}(\cdot \mid X)\right)\right].\end{aligned}$$

□

C Additional Experiments

We provide several additional experiments to illustrate properties of the proposed method.

Ablation Our method (Algorithm 1) introduces a new scheme based on weighted conformal inference with augmentation, where the weights (5) are rigorously derived from theory. The augmentation was the key to making the infeasible oracle method (Algorithm 0) tractable. However, one may wonder if there is another way to design weights without the augmentation, questioning whether the augmentation is necessary for our weighted conformal method. Currently, we believe Algorithm 1 with the weights (5) is the most sensible approach that is computationally tractable and theoretically guaranteed. To demonstrate that the augmentation is a necessary component of our method, we compare our method to a version of Algorithm 1 that uses the weights without the augmentation. In other words, we replace the denominator of the calibration weight $\hat{w}_i$ in (5) with $[\mathcal{A}_{\mathcal{D}_{ca}}(X_i)]_{Y_i}$, namely, clustering with the calibration points only. Then, for $\hat{w}(y)$ in (5), since we cannot define the denominator, we replace it with the one-hot assignment of y based on calibration clustering, which is possible if calibration clustering is generalizable. We test this using the 2D GMM case in Figure 1 (top row). Figure 4 compares the weighted CC (blue, same as Figure 1) to the method above without the augmentation (orange, dashed). We clearly see that without the augmentation, we end up with uninformative sets that are way larger than necessary, even when the sample size is sufficiently large. This demonstrates that the augmentation is an indispensable part of our algorithm.

Misspecified Number of Clusters Currently, our framework (and that of [11]) focuses on the case where we know K, the number of true clusters. However, K is unknown and needs to be estimated in practice. Given the early stage of this line of research on label uncertainty quantification, estimating K, which is a well-known challenge in the literature, is out of scope for this paper. That said, we provide a simple study illustrating the behavior of our algorithm with misspecified K and discuss its potential usage for diagnosing the clustering fit. Figure 5 shows the heatmaps for the 2D moon example in Figure 2. The middle panel corresponds to the second panel of the top row in Figure 2, and the other two panels are produced by the same procedure with $K = 2, 4$ instead of the true $K = 3$, where the points are shown with different markers indicating their true clusters. For $K = 2$, notice that we have learned one cluster (left, orange region) correctly, while the other cluster combines two of the three true clusters. For $K = 4$, we see that there is a region (purple) that passes through the two clusters (right and bottom), leading to the undesirable clustering fit.

Computational Cost Weighted conformal prediction is inherently more computationally expensive than split conformal prediction, as it requires refitting the model and recomputing the weights for each test point. In Tables 1 and 2, we compare the computational costs of Algorithm 1 and Split-CC [11] in terms of wall-clock time and peak memory usage, respectively. We consider the 2D Gaussian mixture model experiment (top row of Figure 1, varying sample size n), where both methods are evaluated on the same task of constructing prediction sets for $m = 500$ test points at a nominal coverage level of $1 - \alpha = 0.9$. All computations are performed on a single core of an Apple M2 processor with 16 GB of memory. The reported results are means (with standard deviations) across five random seeds, with peak memory usage measured using Python's `tracemalloc` module.

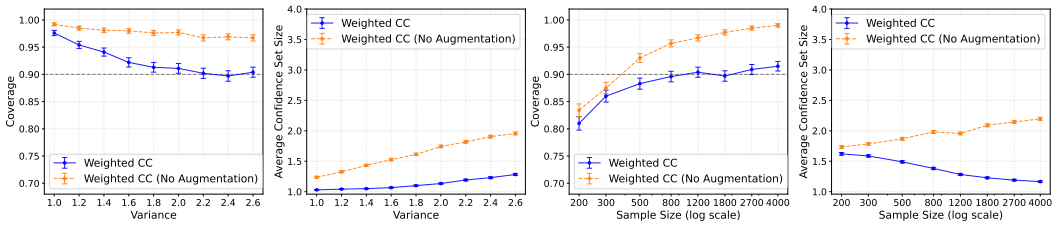


Figure 4: Coverage and confidence set size for GMM simulations with $K = 5$ components for varying noise level σ^2 and sample size n in $\mathbb{R}^2$ as in the top row of Figure 1. Algorithm 1 is compared with the version without the augmentation.

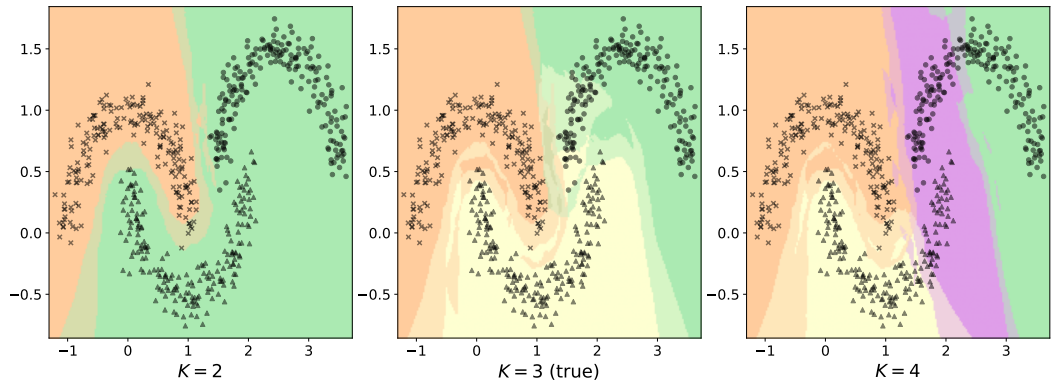


Figure 5: Confidence set heatmaps for the nonlinear clustering simulation (same as the top row of Figure 2) with $K = 2, 3, 4$.

Table 1: Wall-clock time (in seconds) for computing prediction sets of the same $m = 500$ test points; mean (standard deviation) over 5 data seeds, single core.

n	Training (shared)	Split CC	Weighted CC
200	0.0200 (0.0097)	0.0180 (0.0061)	8.37 (2.40)
300	0.0142 (0.0025)	0.0143 (0.0029)	7.77 (1.10)
500	0.0114 (0.0026)	0.0120 (0.0016)	6.42 (1.40)
800	0.0144 (0.0068)	0.0144 (0.0004)	6.56 (0.22)
1200	0.0128 (0.0011)	0.0177 (0.0016)	8.61 (0.77)
1800	0.0147 (0.0013)	0.0219 (0.0017)	10.4 (0.51)
2700	0.0196 (0.0015)	0.0309 (0.0020)	14.5 (0.50)
4000	0.0259 (0.0019)	0.0461 (0.0034)	22.0 (0.52)

Table 2: Peak memory (in MB) for computing prediction sets of the same $m = 500$ test points; one data seed, single core, traced with Python's `tracemalloc`.

n	Training (shared)	Split CC	Weighted CC
200	0.06	0.21	0.49
300	0.05	0.23	0.40
500	0.07	0.27	0.59
800	0.11	0.43	0.66
1200	0.16	0.67	1.01
1800	0.24	1.20	1.61
2700	0.36	2.11	2.37
4000	0.52	3.74	4.17