CAKE: Confidence in Assignments via K-partition Ensembles

Aggelos Semoglou[a,b,*], John Pavlopoulos[a,b]

[a] Department of Informatics, Athens University of Economics and Business, Athens, Greece
[b] Archimedes Research Unit, Athena Research Center, Athens, Greece

ARTICLE INFO

Keywords:
Unsupervised learning
Ensemble learning
Clustering ensembles
Confidence estimation
Error detection

Software:
https://github.com/semoglou/cake
https://pypi.org/project/cake-ensemble/

Published in:
Machine Learning with Applications, Vol. 24, 2026
DOI: https://doi.org/10.1016/j.mlwa.2026.100915
License: CC BY-NC-ND 4.0

ABSTRACT

Clustering is widely used for unsupervised structure discovery, yet it offers limited insight into how reliable each individual assignment is. Diagnostics, such as convergence behavior or objective values, may reflect global quality, but they do not indicate whether particular instances are assigned confidently, especially for initialization-sensitive algorithms like *k*-means. This assignment-level instability can undermine both accuracy and robustness. Ensemble approaches improve global consistency by aggregating multiple runs, but they typically lack tools for quantifying pointwise confidence in a way that combines cross-run agreement with geometric support from the learned cluster structure. This work introduces **CAKE** (Confidence in Assignments via K-partition Ensembles), a framework that evaluates each point using two complementary statistics computed over a clustering ensemble: assignment stability and consistency of local geometric fit. These are combined into a single, interpretable score in [0, 1]. The theoretical analysis shows that CAKE remains effective under noise and separates stable from unstable points. Experiments on synthetic and real-world datasets indicate that CAKE effectively highlights ambiguous points and stable core members, providing a confidence ranking over instances that can be used for selection or prioritization in downstream clustering workflows.

1. Introduction

Clustering is a fundamental task in unsupervised machine learning, widely used to uncover structure in unlabeled data (Aggarwal & Reddy, 2013; Jain et al., 1999). It has core applications in pattern discovery, exploratory analysis, and decision-making across both scientific and applied domains (Kaufman & Rousseeuw, 2009). However, unlike supervised learning, where confidence estimation techniques such as conformal prediction (Shafer & Vovk, 2007) or calibrated probabilities (Guo et al., 2017) are common, clustering methods typically do not offer reliable confidence scores for individual data point assignments. This makes it difficult to assess assignment trustworthiness, which is particularly important when downstream decisions rely on clustering results.

In practice, clustering algorithms often exhibit sensitivity to initialization, susceptibility to local optima, and vulnerability to noisy or ambiguous data (Xu & Wunsch, 2005). Consequently, outcomes may vary across executions, even under fixed algorithmic settings. For example, *k*-means (MacQueen, 1967), despite its popularity, can produce divergent results across runs due to random initialization (Bubeck et al., 2012). Practitioners often mitigate this by running the algorithm multiple times with different random seeds and selecting the best result based on internal validation metrics. However, this approach only addresses global variability and offers no insight into the reliability of individual point assignments. Such assignment-level instability is particularly problematic in settings like anomaly detection or scientific discovery, where unreliable assignments can mislead interpretation or obscure important patterns. Moreover, most existing evaluation methods assess clustering quality at a global or cluster level, using objective values or internal validation metrics such as the Silhouette score or Davies–Bouldin index (Arbelaitz et al., 2013; Vendramin et al., 2009).

Ensemble-based techniques like consensus clustering (Aktas et al., 2024; Strehl & Ghosh, 2002; M. Zhang, 2022) have also been explored to improve robustness by aggregating multiple partitions. However, they still lack interpretable, pointwise confidence scores that capture both cross-run agreement in label assignments and geometric fit. In response, many ensemble-style uncertainty heuristics focus primarily on *agreement*, for example, counting aligned "votes" for each point's label across runs or measuring dispersion of assignments, to identify unstable samples (Inkaya, 2023; X. Zhang et al., 2025). While useful, agreement alone does not reveal whether a point is *geometrically* well supported

*Corresponding author
ang.semoglou@aueb.gr (A. Semoglou); annis@aueb.gr (J. Pavlopoulos)
ORCID(s): 0009-0005-3715-530X (A. Semoglou); 0000-0001-9188-7425 (J. Pavlopoulos)

by its assigned cluster, since a point can be consistently assigned due to a systematic bias or an overly rigid decision boundary even when it is isolated or weakly connected to the cluster structure. Conversely, purely geometric signals (such as the Silhouette score) computed within a single run can be overly optimistic in ambiguous regions: a point may appear well placed under that run's geometry, yet still switch clusters across runs because it lies near a boundary or because multiple partitions explain it nearly equally well (Liu et al., 2022). These complementary failure modes (Fig. 1) motivate a confidence signal that fuses stability and geometry at the point level.

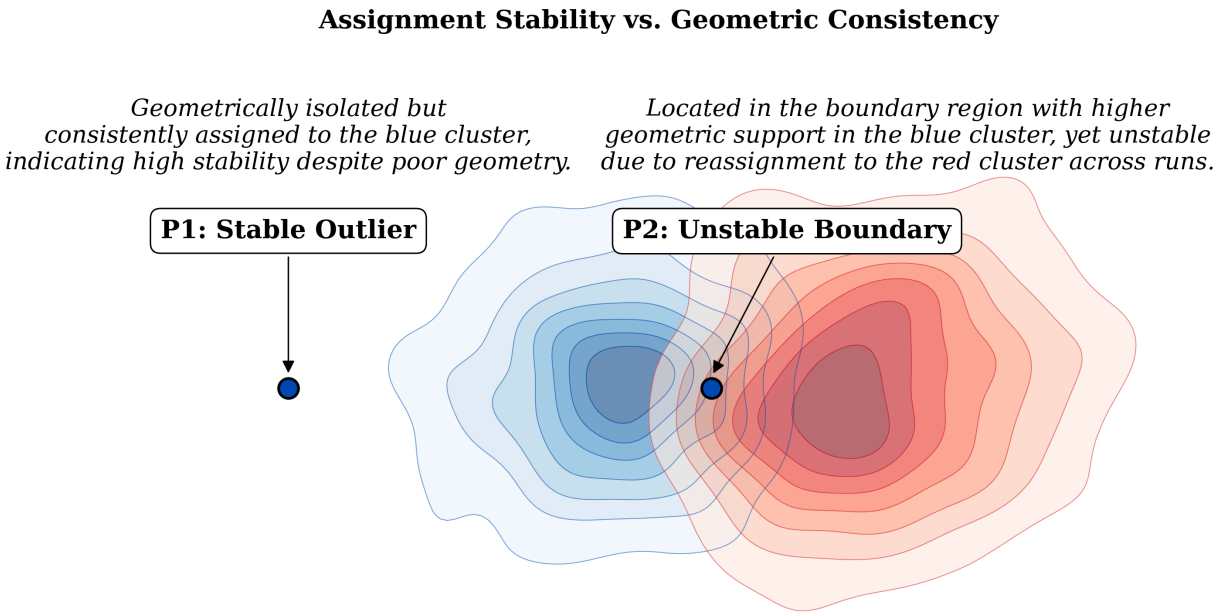


Figure 1: Assignment stability and geometric consistency can fail in complementary ways. P1 (stable outlier; left) is consistently assigned (high stability) despite weak integration into its assigned cluster structure, while P2 (unstable boundary; right) shows higher within-cluster fit to one cluster in a single run yet switches labels across runs near the boundary. These cases suggest that reliable pointwise confidence should account for both signals jointly.

This raises a central question:

How can each data point be assigned a confidence score that, across runs, reflects both the assignment stability and the consistency of its geometric support under the learned clustering structure?

To address this question, **CAKE** (Confidence in Assignments via K-partition Ensembles) is introduced as a framework that quantifies per-point confidence by combining two complementary statistics derived from an ensemble of clustering partitions (Fig. 2): (*i*) pairwise assignment agreement, computed via optimal label alignment using the Hungarian algorithm (Kuhn, 1955; Meila, 2007), and (*ii*) local geometric consistency, measured through aggregated Silhouette statistics. These components are integrated into a single confidence score in $[0, 1]$ for each point, indicating how strongly its cluster membership is supported and providing a fine-grained ranking that highlights ambiguous points and stable core members. The paper provides theoretical guarantees for the statistical reliability of CAKE scores and demonstrates, across synthetic and real-world datasets, that CAKE effectively distinguishes between high- and low-confidence assignments. This enables selective removal or prioritization of points, improving clustering quality and interpretability in a fully label-free setting. Empirically, CAKE stabilizes with a modest number of runs, yielding a reliable confidence ranking without requiring large ensembles. This keeps the method lightweight and practical in iterative, exploratory clustering workflows. CAKE introduces a principled framework for per-instance confidence estimation in clustering, bridging ensemble diversity with pointwise assessment of assignment quality. More broadly, CAKE turns clustering ensembles into a practical, per-point diagnostic that complements global validation.

The CAKE implementation and experimental code are publicly available at https://github.com/semoglou/cake.
The Python package is available on PyPI at https://pypi.org/project/cake-ensemble/.

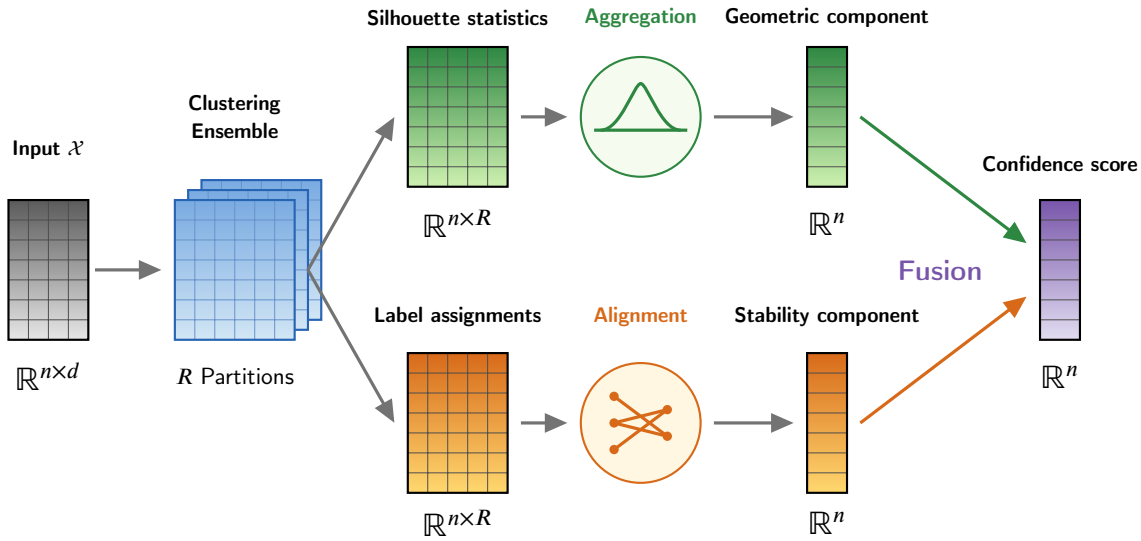


Figure 2: CAKE framework overview. Across an ensemble of R clustering runs (partitions), CAKE aggregates Silhouette statistics into a geometric component and (aligned) label assignments into a stability component, then fuses the two into a confidence score for each data point.

2. Related Work

Assessing the quality of clustering results has been the focus of extensive prior work. Classical validation metrics such as the Silhouette score (Dudek, 2020; Pavlopoulos et al., 2025; Rousseeuw, 1987), Davies–Bouldin index (Davies & Bouldin, 1979), and Calinski–Harabasz criterion (Calinski & Harabasz, 1974) provide global or cluster-level assessments of compactness and separation, but they are not designed to quantify the reliability of individual assignments or to isolate unstable or ambiguous samples within a partition (Arbelaitz et al., 2013).

Clustering ensembles aim to improve robustness by aggregating multiple partitions, using techniques such as co-association matrices or consensus functions (Fred & Jain, 2005; Inkaya, 2023; Strehl & Ghosh, 2002) and more recent ensemble-selection approaches (Golalipour et al., 2021). While effective at stabilizing global structure, these methods typically do not provide per-point confidence scores (Boongoen & Iam-On, 2018; M. Zhang, 2022). Their focus lies in producing a single consensus clustering, rather than quantifying how reliably each point is assigned across runs (Topchy et al., 2005). As a result, the consensus output can obscure localized disagreement in regions near cluster boundaries or in heterogeneous areas of the data, making it difficult to identify unstable points. When point-level information is available (e.g., via co-association counts), it is usually an intermediate artifact used to form the consensus rather than an explicit confidence score for diagnosing assignments. Moreover, ensemble aggregation is typically driven by partition agreement and does not incorporate pointwise geometric evidence of assignment quality into the confidence signal.

Efforts to incorporate uncertainty into clustering include fuzzy clustering (Bezdek, 1981; Bezdek et al., 1984), probabilistic mixture models, and bootstrap-based stability analysis (Ben-Hur et al., 2002; Hennig, 2007; Lange et al., 2004; Liu et al., 2022). In related ensemble-stability work, per-point uncertainty is often summarized by the dispersion of aligned label "votes" across runs (e.g., entropy) (Ayad & Kamel, 2008). Recent deep clustering work also studies calibrated cluster confidence, for example via deep clustering networks explicitly trained for confidence calibration (Jia et al., 2025). These approaches provide soft assignments or probability estimates, but may be less interpretable when applied to hard clustering results; in particular, agreement-focused resampling or label-alignment heuristics quantify per-point consistency but omit local geometric fit, leaving no unified stability–geometry measure. In addition, model-based techniques such as Gaussian Mixture Models (Reynolds, 2009) require distributional assumptions, while bootstrapping methods can be computationally intensive and sensitive to sampling variability.

Another relevant area involves the challenge of label alignment when comparing clustering results, often addressed using the Hungarian algorithm for optimal permutation matching (Kuhn, 1955; Meila, 2007).

While this alignment step is standard in ensemble clustering, it has been less commonly used to systematically quantify per-point assignment stability across an ensemble.

Extending these approaches, CAKE combines optimal label alignment with local geometric analysis to provide a principled, interpretable confidence score for each data point, bridging ensemble-based robustness with pointwise reliability and grounding the ensemble consensus in geometric evidence. CAKE draws inspiration from supervised ensemble techniques, where disagreement between classifiers informs uncertainty (Beluch et al., 2018; Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017). Such methods leverage variation across independently trained models or stochastic passes (e.g., dropout) to estimate predictive confidence. In contrast, CAKE operates in an unsupervised regime, using clustering ensemble diversity and within-cluster geometric coherence to infer confidence scores without labels.

3. Methodology

3.1. Setting & Notation

Assume a dataset $\mathcal{X} = \{x_i\}_{i=1}^{n} \subset \mathbb{R}^d$ consisting of n data points. Let $\mathcal{C} = \{C^{(1)}, C^{(2)}, \dots, C^{(R)}\}$ denote a collection of R clustering results obtained by repeatedly applying the same clustering algorithm to $\mathcal{X}$, each time using the same number of clusters k but with a different random seed.

This collection forms a clustering ensemble by capturing multiple partitions of the dataset under stochastic variation. Although all runs use the same algorithm and parameters, random initialization introduces meaningful differences in the resulting partitions, reflecting uncertainty in how individual points are grouped. More generally, the ensemble could be constructed using resampling strategies (e.g., clustering on bootstrapped subsets of $\mathcal{X}$), or by aggregating partitions from different clustering algorithms, offering a broader perspective on assignment variability (Fig. 4). While the proposed framework is compatible with such (heterogeneous) ensemble construction strategies (Fig. 10), including resampling and multi-algorithm clustering ensembles, here, the focus is on (homogeneous) ensembles formed by repeated applications of a single clustering method, allowing assignment-level confidence to be quantified without interference from other sources of variation.

To use these partitions quantitatively, let $\mathcal{L} = \{L^{(1)}, L^{(2)}, \dots, L^{(R)}\}$ denote the corresponding set of assignments. Each labeling $L^{(r)} \in \mathcal{L}$ assigns a cluster label to every data point $x_i \in \mathcal{X}$, such that $L^{(r)} = \left\{L_1^{(r)}, L_2^{(r)}, \dots, L_n^{(r)}\right\}$, where $L_i^{(r)} \in \{1, 2, \dots, k\}$: the label of x_i in clustering run r.

3.2. Ensemble Silhouette Statistics

For each point $x_i \in \mathcal{X}$, let $s_i^{(r)}$ denote its Silhouette score under clustering run $r \in \{1, 2, \dots, R\}$,

$$s_i^{(r)} = \frac{b_i^{(r)} - a_i^{(r)}}{\max\left\{a_i^{(r)},\ b_i^{(r)}\right\}} \in [-1, 1], \tag{1}$$

which quantifies the quality of its assignment by comparing its average intra-cluster distance $a_i^{(r)}$ to its minimum average inter-cluster distance $b_i^{(r)}$ under clustering run r.

Computing all $s_i^{(r)}$ is $O(n^2 d)$ per clustering run. When the base method is centroidal (e.g., k-means, k-means++) and n is large, a centroid-based Silhouette approximation (Wang et al., 2017) can be computed instead by replacing $\tilde{a}_i^{(r)} = \|x_i - \mu_{C_i}^{(r)}\|$ and $\tilde{b}_i^{(r)} = \min_{C \neq C_i} \|x_i - \mu_C^{(r)}\|$, where $\mu_C^{(r)}$ is the centroid of cluster C in run r, and C_i denotes the cluster index of x_i in run r (i.e., $C_i = L_i^{(r)}$). This reduces the per–run cost to $O(nkd)$ while tracking the exact score closely in centroidal settings.

Aggregating over the ensemble. The mean Silhouette score and standard deviation (std) are computed for each point x_i across all R clustering runs in the ensemble:

$$\mu_i = \frac{1}{R}\sum_{r=1}^{R} s_i^{(r)}, \quad \sigma_i = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\left(s_i^{(r)} - \mu_i\right)^2} \tag{2}$$

These two quantities are used to define a per-point measure of *Silhouette-based reliability*. A high mean μ_i indicates that x_i tends to be well-placed geometrically across clustering runs, while a low std σ_i implies that this quality is consistent. Subtracting σ_i from μ_i penalizes points with high variability in Silhouette values across runs, emphasizing both quality and stability of geometric fit. The resulting score is thresholded at zero to ensure non-negativity:[1]

$$\tilde{S}_i = \left(\mu_i - \sigma_i\right)_+ = \max\left\{0,\ (\mu_i - \sigma_i)\right\} \in [0, 1]. \tag{3}$$

$\tilde{S}_i$ rewards points that not only have high average Silhouette (good cluster fit) but also low variability across runs (stable geometric placement). However, a high $\tilde{S}_i$ does not guarantee that a point is consistently assigned to the same cluster across runs. It only indicates that the point maintains a consistently strong fit within its assigned cluster. To capture actual assignment stability, a complementary measure is introduced based on cluster agreement across the ensemble.

3.3. Ensemble Assignment Stability

Label alignment. Given two clustering labelings $L^{(r_1)}, L^{(r_2)} \in \mathcal{L}$ from clustering runs r_1, r_2, the labels may not be aligned: that is, cluster label j in $L^{(r_1)}$ and in $L^{(r_2)}$ may correspond to a different group of points. To align $L^{(r_2)}$ onto $L^{(r_1)}$, a contingency matrix $M \in \mathbb{N}^{k\times k}$ is defined, where each entry $M_{i,j}$ counts the number of points assigned to cluster i in $L^{(r_1)}$ and cluster j in $L^{(r_2)}$:

$$M_{i,j} = \sum_{p=1}^{n} \mathbb{1}\left\{\left(L_p^{(r_1)} = i\right) \wedge \left(L_p^{(r_2)} = j\right)\right\} \tag{4}$$

where $\mathbb{1}(\cdot)$ is the indicator function. Finding the optimal alignment corresponds to finding a permutation π^*, with $\pi^* : \{1, \dots, k\} \to \{1, \dots, k\}$, that maximizes agreement:

$$\pi^* = \arg\max_{\pi} \sum_{i=1}^{k} M_{i,\pi(i)}, \tag{5}$$

where $\pi(i)$ denotes the label in $L^{(r_2)}$ that is matched to label i in $L^{(r_1)}$. This optimization is equivalent to solving a *linear sum assignment problem*, which can be solved using the Hungarian algorithm in $O(k^3)$ time. After aligning $L^{(r_2)}$ onto $L^{(r_1)}$, the aligned labeling is denoted by $L^{(r_2\to r_1)}$ and the pointwise agreement between two clustering runs $(r_1,\ r_2)$ for a data point $x_i \in \mathcal{X}$ is defined as:

$$A_i^{(r_1,r_2)} = \mathbb{1}\left\{L_i^{(r_1)} = L_i^{(r_2\to r_1)}\right\} \in \{0, 1\}. \tag{6}$$

Instance pairwise stability. Given R clustering runs, there are $\binom{R}{2} = \frac{R(R-1)}{2}$ distinct unordered pairs of partitions. The pairwise stability score c_i is defined for every point $x_i \in \mathcal{X}$ as the fraction of all unordered run-pairs in which x_i is assigned to the same cluster (after alignment):

$$c_i = \frac{2}{R(R-1)} \sum_{r_1<r_2} A_i^{(r_1,r_2)} \in [0, 1]. \tag{7}$$

A common alternative is to align all runs to a single reference partition (e.g., a medoid run) and compute vote-based uncertainty (such as entropy). This can be attractive in time-sensitive applications because it reduces alignment cost from quadratic to linear in R. However, reference-based scores can depend on the chosen run and may inherit its instability, effectively privileging one partition as canonical. In contrast, Eq. 7 averages agreement over *all* unordered run-pairs, yielding a symmetric stability score that is less sensitive to any single run. Practically, c_i can still be computed efficiently by streaming over run-pairs and accumulating agreements without storing all pairwise indicators.

By definition, c_i rewards points that are consistently assigned to the same cluster across the ensemble. Points with high c_i exhibit minimal assignment variation and can be viewed as stable members of the underlying clustering structure. Moreover, since $A_i^{(r_1,r_2)}$ is a Bernoulli indicator taking values in $\{0, 1\}$, the quantity c_i is an order-2 U-statistic with a binary (Bernoulli) kernel (Lee, 1990). As such, c_i is an *unbiased estimator of the true pairwise assignment stability* $\mathbb{E}[A_i^{(r_1,r_2)}]$ (i.e., the expected agreement over pairs of runs, which reflects how reproducibly x_i is clustered across the ensemble).

[1] To preserve the signal from consistently negative means (e.g., $\mu_i < 0$ with small σ_i; a relatively uncommon regime in practice), $\mu_i - \sigma_i$ can be mapped to $[0, 1]$ via $\left[(\mu_i - \sigma_i) + 1\right]/2$

3.4. CAKE Score

From the clustering ensemble, two complementary per-instance diagnostics are derived: the assignment-stability estimate c_i (Eq. 7) and the Silhouette-based reliability score $\tilde{S}_i$ (Eq. 3). CAKE combines these signals into a single confidence value in $[0, 1]$ (outlined in Fig. 2 & Algorithm 1). It rewards observations that are repeatedly assigned to the same cluster across runs and are strongly supported by their local geometry, while penalizing points with unstable assignments or poor local separation from other clusters. The two formulations are:

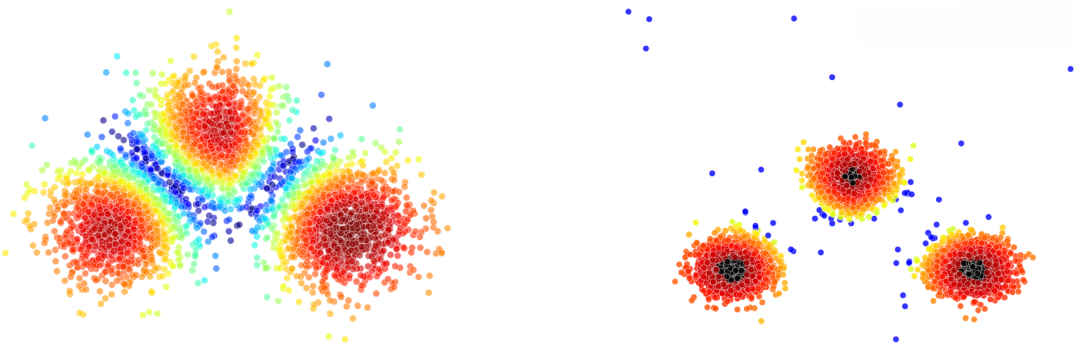
$$\mathrm{CAKE}_i^{(\mathrm{PR})} = c_i \, \tilde{S}_i \tag{8a}$$

$$\mathrm{CAKE}_i^{(\mathrm{HM})} = \frac{2\, c_i \, \tilde{S}_i}{c_i + \tilde{S}_i} \tag{8b}$$

The product (Eq. 8a) attains high values only when both components are large and rapidly downweights points that are weak in either stability or geometric fit. The harmonic mean (Eq. 8b) provides a smoother trade-off, remaining sensitive to low values while being less dominated by a single moderate component, which can make the combined score easier to interpret. This mirrors the role of the F1 score in classification, which balances precision and recall via a harmonic mean (Powers, 2011), and similarly treats stability and geometric fit as complementary requirements for high confidence (see the score distribution and percentiles on synthetic data in Fig. 3).

Algorithm 1 CAKE scores computation

Require: Dataset $\mathcal{X}$, Clustering algorithm(s) $\mathcal{F}$, Ensemble size $R > 1$
Ensure: CAKE scores $\forall x_i \in \mathcal{X}$
  Initialize label assignments matrix $L \in \mathbb{N}^{n \times R}$ ▷ Assigned labels per point/clustering run
  Initialize Silhouette score matrix $S \in \mathbb{R}^{n \times R}$ ▷ Silhouette per point/clustering run
  Initialize Agreement Vector $A \in \mathbb{N}^n$ ▷ Pairwise (clustering) run agreements
  for $r = 1$ to R **do**
    Run clustering algorithm $\mathcal{F}$ on $\mathcal{X}$ with random seed r ▷ Ensemble construction (§3.1)
    Let $L[:, r] \leftarrow$ predicted labels
    Compute Silhouette scores $S[:, r]$ (Eq. 1)
  end for
  $\mu \leftarrow \frac{1}{R} \sum_{r=1}^{R} S[:, r]$, $\sigma \leftarrow \sqrt{\frac{1}{R} \sum_{r=1}^{R} (S[:, r] - \mu)^2}$ (Eq. 2) ▷ Ensemble Silhouette statistics (§3.2)
  Geometric component $\tilde{S} \leftarrow (\mu - \sigma)_+$ (Eq. 3)
  for each clustering pair (r_1, r_2) with $r_1 < r_2$ **do** ▷ Ensemble assignment stability (§3.3)
    Compute optimal label mapping $\pi^{(r_2 \to r_1)}$ (Eq. 5)
    $A \leftarrow A + \mathbb{1}\left\{L[:, r_1] = \pi^{(r_2 \to r_1)}(L[:, r_2])\right\}$ (Eq. 6)
  end for
  Stability component $c \leftarrow 2A[R(R-1)]^{-1}$ (Eq. 7)
  CAKE $\leftarrow$ $\mathrm{CAKE}^{(\mathrm{PR})}$ or $\mathrm{CAKE}^{(\mathrm{HM})}$ (Eq. 8) ▷ CAKE scores computation (§3.4)
  return CAKE



(a) **Low** → **high** CAKE scores.

(b) **Bottom 1%** → **top 90%** CAKE scores

Figure 3: $\mathrm{CAKE}^{(\mathrm{PR})}$ (Eq. 8a) scores distribution (left) and percentiles (right) on synthetic data.

Complexity. With the centroid proxy, computing Silhouette scores over R runs costs $O(Rnkd)$, while exact computations cost $O(Rn^2d)$. Label alignment over all run-pairs builds a $k \times k$ contingency and solves a Hungarian assignment per pair, for $O\big(\binom{R}{2}(n+k^3)\big)$ time; thus CAKE runs in $O(Rnkd + \binom{R}{2}(n+k^3))$ overall, or $O(Rn^2d + \binom{R}{2}(n+k^3))$ using exact Silhouette scores, and uses $O(nR)$ memory (see runtime vs. R, n on synthetic data in Fig. 12 and Tables 5–6).

Non-convex data. The silhouette component (Eqs. 1–3) is typically computed with Euclidean distances and thus favors convex, isotropic clusters. For non-convex structure, Euclidean silhouette can be misleading. CAKE accommodates such settings by replacing Euclidean distances in the silhouette computation (§3.2) with distances induced by a *positive semidefinite (PSD) kernel*. Specifically, a *kernelized Silhouette* is used, replacing Euclidean distances by RKHS distances to cluster means under a PSD kernel κ (see Fig. 4 for an application to non-convex data):

$$d_\kappa^2(x_i, C) = \kappa(x_i, x_i) - \frac{2}{|C|}\sum_{x_j \in C}\kappa(x_i, x_j) + \frac{1}{|C|^2}\sum_{x_p \in C}\sum_{x_q \in C}\kappa(x_p, x_q). \tag{9}$$

Here, $\kappa(\cdot,\cdot)$ is a *positive semidefinite (PSD) kernel*, i.e., a symmetric similarity function such that the Gram matrix $[\kappa(x_i, x_j)]_{i,j}$ is PSD. Intuitively, $\kappa(x, y)$ measures how similar x and y are, and it corresponds to an inner product in some feature space: $\kappa(x, y) = \langle \phi(x), \phi(y)\rangle$.

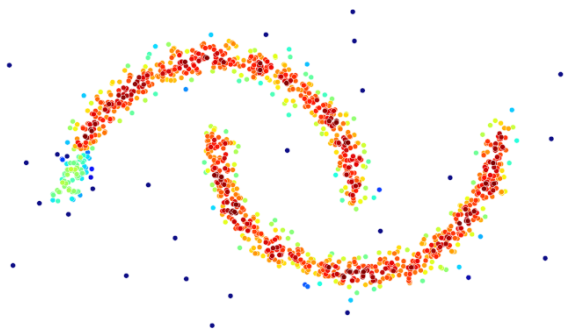


Figure 4: CAKE$^{(PR)}$ distribution on the two-moons dataset (n=1200) with 3% uniform outliers. An ensemble of R=25 partitions is constructed using **spectral clustering** on a k-NN graph, jittering $n_{\text{neighbors}} \in [10, 15]$ per run to induce diversity. For the geometric component, the kernelized Silhouette (§3.2; Eq. 9) is used with a self-tuning RBF kernel: $\kappa(x_i, x_j) = \exp\big[-\|x_i - x_j\|^2/(\sigma_i\sigma_j)\big]$. The Gram matrix K is built once and reused across runs, setting each σ_i to the distance from x_i to its k_{nn}=7th nearest neighbor. Points are colored by their CAKE scores (↑ **red**, ↓ **blue**).

4. Theoretical Analysis

Two non-asymptotic results are provided for the finite-ensemble stability scores c_i (Eq. 7); full derivations and analogous discussion for the geometric component $\tilde{S}_i$ (Eq. 3) are given in Appendix A.

4.1. Ranking-Error Bound for Finite Ensembles

Let $R \geq 2$ be the number of independent clustering runs performed on $\mathcal{X}$. For a point $x_i \in \mathcal{X}$, the true pairwise stability is $\theta_i = \mathbb{E}\left[A_i^{(r_1,r_2)}\right]$, where $A_i^{(r_1,r_2)}$ (Eq. 6) is the Bernoulli kernel used in the order-2 U-statistic estimator c_i (Eq. 7). Intuitively, the empirical scores should rank points in the same order as their true stability θ_i and any reversed ordering should be random noise that fades when more runs are added.[2] The following bound shows how the misordering probability shrinks with R (derivations in Appendix A.1). For any two points $x_i, x_j \in \mathcal{X}$ and margin $\gamma > 0$, it holds:

$$\mathbb{E}\left[A_i^{(r_1,r_2)}\right] \geq \mathbb{E}\left[A_j^{(r_1,r_2)}\right] + \gamma \Rightarrow \Pr\left[c_i < c_j\right] \leq 2\exp\left\{-R\frac{\gamma^2}{8}\right\}. \tag{10}$$

4.2. False-Positive Bound for Uniform-Noise Points

Consider a uniform-noise observation, i.e., a point whose cluster label in each run is drawn independently and uniformly from $\{1, \ldots, k\}$. Then $\theta_i = \frac{1}{k}$ and deviations of c_i from θ_i decay exponentially in R (Hoeffding, 1963). Thus, the

[2] By standard concentration for bounded U-statistics, c_i converges to θ_i exponentially in R (Appendix A.1).

probability that a noise point falsely attains a high score decreases rapidly with ensemble size (Appendix A.2):

$$\Pr[c_i > \tau \mid x_i \text{ uniform-noise}] \leq \exp\left\{-\frac{R}{2}\left(\tau - \frac{1}{k}\right)^2\right\} \quad \forall \text{ threshold } \tau > \frac{1}{k}. \tag{11}$$

More generally, for any label distribution with expected agreement $\theta_i = \mathbb{E}[A_i^{(r_1,r_2)}]$: $\Pr[c_i > \tau] \leq \exp\left\{-\frac{R}{2}(\tau - \theta_i)^2\right\}$, $(\forall \tau > \theta_i)$ so the probability still decays exponentially in R (detailed analysis in Appendix A.2).

5. Empirical Validation

5.1. Synthetic & Real-World Datasets

CAKE is evaluated on synthetic and real-world datasets. The **synthetic datasets** are described next (visualizations in Fig. 5). **S1** has 4,000 points sampled from three Gaussian clusters with equal size and std of 2.0; **S2** has 3,000 points from three clusters with std of 2.0, 2.5, and 1.5. **S3** has 4,500 points, where 3,000 belong to three Gaussian clusters with unit std, and 1,500 are uniformly distributed noise points. **S4** has 3,000 points arranged in four wide Gaussian clusters, located at the corners of a square. **S5** has 4,000 points from three clusters with strong density contrast: one tight cluster with std of 0.2, one wide with std of 3.0, and one intermediate with unit std. **S6** has 4,000 points from three clusters with std of 0.4, 2.5, and 0.4, where a sparse central cluster overlaps denser ones on each side. **S7** has 4,000 points, distributed across three clusters of size 500, 1,000, and 2,500, with std of 0.3, 1.5, and 2.5 respectively.

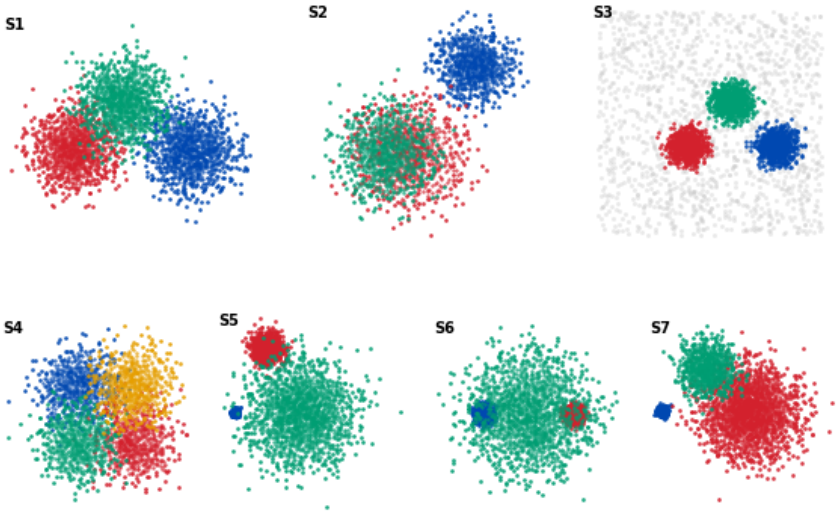


Figure 5: Synthetic datasets (**S1**-**S7**), colored by ground-truth cluster labels; noise in light gray.

The **real-world datasets**, which are publicly available via OpenML (Vanschoren et al., 2014), SCIKIT-LEARN (Pedregosa et al., 2011), or TensorFlow (Abadi et al., 2015), span a wide range of domains and structures (sizes, dimensionality, and preprocessing are summarized in Table 1): **Iris** (IR), **Breast Cancer** (BC), **Pendigits** (PD), **Letter** (LT), **Digits** (DG), **Fashion MNIST** (FM), **Satimage** (SA), and **20 Newsgroups** (NG).

Dataset (abbrev.)	Modality	n	d	k	Preprocessing / Representation
Iris (IR)	Tabular	150	4	3	None
Breast Cancer (BC)	Tabular	569	10	2	Standardize; PCA
Pendigits (PD)	Tabular	10,992	16	10	None
Letter (LT)	Tabular	20,000	16	26	Standardize
Digits (DG)	Image	1,797	64	10	Flatten 8×8 grayscale images
Fashion MNIST (FM)	Image	60,000	784	10	Flatten 28×28 grayscale images
Satimage (SA)	Remote sensing	6,435	30	6	PCA
20 Newsgroups (NG)	Text	18,846	100	20	all-MiniLM-L6-v2 embeddings; PCA

Table 1: Real datasets. n: number of samples; d: post-preprocessing dimensionality; k: number of classes (clusters).

5.2. Evaluation Setup

Instance removal. Instances are ranked according to their CAKE scores (§3.4), and the highest-confidence subset is then selected in order to assess the effect of retaining the most reliable points on downstream clustering quality. For each dataset, CAKE scores are computed using an ensemble of 20 independent k-means runs at the ground-truth number of clusters k (random initialization for synthetic datasets; k-means++ for real datasets). k-means is used because it is widely used, easy to interpret, and known to be sensitive to initialization, making it a natural candidate for assessing assignment-level confidence; CAKE itself is model-agnostic and applies to any hard-assignment clustering ensemble (see Figs. 4, 10 and Appendix C). In this setting, CAKE is treated primarily as a pointwise confidence score that induces a ranking over instances, while filtering is considered an application of that ranking under a specified retention fraction. Accordingly, the experiments are designed to assess whether the induced ranking distinguishes more reliable from less reliable instances under a fixed clustering resolution. The subsequent subset-selection analysis then provides a downstream test of whether high-confidence instances are also those whose assignments are more consistently supported by the local cluster structure.

For each dataset, six equal-size subsets are formed, each retaining 70% of the points, $m = \lfloor 0.7n \rfloor$, according to different criteria: **Random** (sample m points uniformly), **Consensus** (top m by across-run label agreement after aligning all runs to a reference–medoid partition, i.e., the most representative run), $\tilde{\mathbf{S}}$ (top m by the Silhouette component; Eq. 3), **C** (top m by the stability component; Eq. 7), **CAKE**$^{(\mathbf{PR})}$ (top m by the product; Eq. 8a), and **CAKE**$^{(\mathbf{HM})}$ (top m by the harmonic mean; Eq. 8b). While this specific threshold is not necessarily optimal, it offers a reasonable coverage–confidence trade-off (see Fig. 7; Appendix Fig. 13). An adaptive thresholding variant, in which the retained fraction is selected automatically from the data, is reported separately in Appendix D.

Clustering performance is compared on the full datasets and on their filtered subsets using new k-means runs (separate from the CAKE ensemble) at the ground-truth k. Performance is measured via Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI), and clustering accuracy (ACC), after Hungarian alignment to ground-truth labels. To assess variability, the evaluation is repeated with multiple independent runs and means are reported with (Student's t) 95% confidence intervals (Table 2; see Appendix B Tables 11–12 for MiniBatchKMeans and GMM evaluation). Additional results for Silhouette, Normalized Mutual Information (NMI), and the correlation between CAKE score percentiles and clustering accuracy are reported in Appendix B (Table 7).

Convergence. The stabilization of CAKE scores with increasing ensemble size R is studied. For each $R \in \{5, \ldots, 70\}$, B=10 sub-ensembles of size R are drawn, CAKE scores are recomputed using the centroid-based Silhouette proxy (§3.2), and the per-point standard deviation across the B estimates is reported (Fig. 6). Lower variability suggests that CAKE scores are stabilizing (converging) as the clustering ensemble grows.

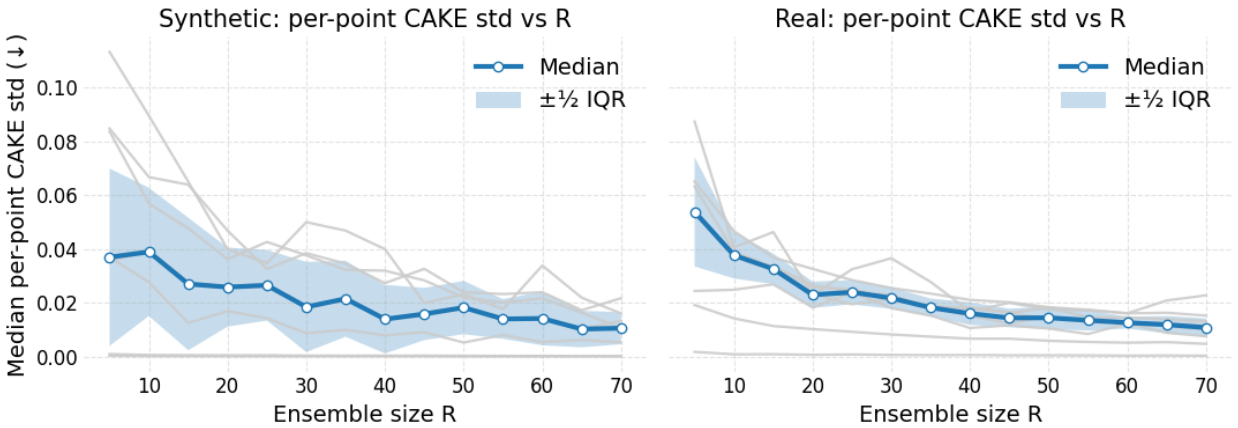


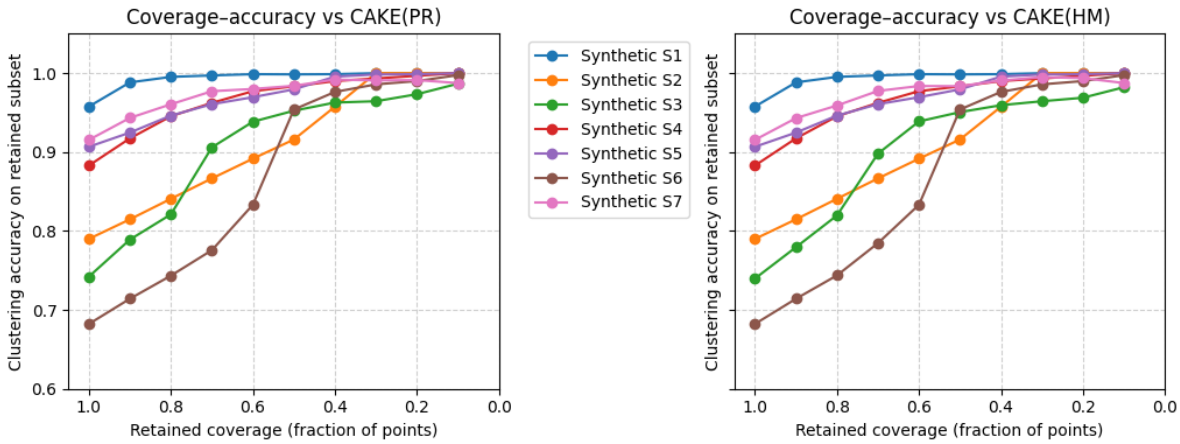
Figure 6: Convergence vs R. Median per-point CAKE$^{(\mathrm{HM})}$ scores (Eq. 8b) std vs. ensemble size R for synthetic (left) and real (right) datasets. **Lines:** light gray: individual datasets; blue: across-dataset median. **Band:** $\pm\frac{1}{2}$ IQR. Variability drops and stabilizes by $R \approx 30$–40 across datasets.

Dataset	Subset	avg ARI	avg AMI	avg ACC
S1	Full	0.879 [0.8785, 0.8791]	0.833 [0.8331, 0.8338]	0.958 [0.9578, 0.9580]
	Random	0.879 [0.8790, 0.8790]	0.834 [0.8339, 0.8339]	0.958 [0.9579, 0.9579]
	Consensus	0.884 [0.8842, 0.8842]	0.839 [0.8394, 0.8394]	0.960 [0.9596, 0.9596]
	C component	0.884 [0.8842, 0.8842]	0.839 [0.8394, 0.8394]	0.960 [0.9596, 0.9596]
	$\tilde{S}$ component	**0.992** [0.9921, 0.9921]	**0.983** [0.9827, 0.9827]	**0.997** [0.9971, 0.9971]
	CAKE$^{(PR)}$	**0.992** [0.9921, 0.9921]	**0.983** [0.9827, 0.9827]	**0.997** [0.9971, 0.9971]
	CAKE$^{(HM)}$	**0.992** [0.9921, 0.9921]	**0.983** [0.9827, 0.9827]	**0.997** [0.9971, 0.9971]
S2	Full	0.552 [0.5225, 0.5812]	0.606 [0.6056, 0.6068]	0.762 [0.6993, 0.8244]
	Random	0.529 [0.4901, 0.5684]	0.601 [0.5923, 0.6102]	0.724 [0.6441, 0.8029]
	Consensus	0.528 [0.5070, 0.5492]	0.628 [0.6031, 0.6530]	**0.797** [0.7838, 0.8094]
	C component	0.528 [0.5070, 0.5492]	0.628 [0.6031, 0.6530]	**0.797** [0.7838, 0.8094]
	$\tilde{S}$ component	0.679 [0.5731, 0.7840]	0.698 [0.6789, 0.7162]	0.759 [0.6359, 0.8814]
	CAKE$^{(PR)}$	**0.680** [0.5741, 0.7860]	**0.700** [0.6800, 0.7192]	0.761 [0.6393, 0.8827]
	CAKE$^{(HM)}$	**0.680** [0.5738, 0.7858]	0.699 [0.6796, 0.7182]	0.760 [0.6367, 0.8829]
S3	Full	0.498 [0.4941, 0.5016]	0.614 [0.6098, 0.6178]	0.733 [0.7268, 0.7384]
	Random	0.449 [0.3792, 0.5188]	0.580 [0.5352, 0.6255]	0.678 [0.6076, 0.7479]
	Consensus	0.753 [0.7090, 0.7973]	0.757 [0.7301, 0.7836]	0.842 [0.8013, 0.8828]
	C component	0.719 [0.6957, 0.7432]	0.721 [0.7041, 0.7379]	0.830 [0.8128, 0.8473]
	$\tilde{S}$ component	**0.795** [0.7667, 0.8241]	**0.781** [0.7591, 0.8022]	**0.862** [0.8348, 0.8886]
	CAKE$^{(PR)}$	0.734 [0.6444, 0.8232]	0.751 [0.6982, 0.8031]	0.784 [0.7085, 0.8591]
	CAKE$^{(HM)}$	0.784 [0.7538, 0.8137]	0.776 [0.7597, 0.7932]	0.837 [0.8022, 0.8724]
S4	Full	0.716 [0.7158, 0.7167]	0.678 [0.6775, 0.6786]	0.884 [0.8833, 0.8837]
	Random	0.727 [0.7268, 0.7276]	0.693 [0.6926, 0.6933]	0.888 [0.8883, 0.8886]
	Consensus	0.606 [0.5946, 0.6179]	0.598 [0.5939, 0.6029]	0.755 [0.7432, 0.7665]
	C component	0.606 [0.5946, 0.6179]	0.598 [0.5939, 0.6029]	0.755 [0.7432, 0.7665]
	$\tilde{S}$ component	**0.902** [0.9022, 0.9022]	**0.867** [0.8667, 0.8667]	**0.962** [0.9624, 0.9624]
	CAKE$^{(PR)}$	**0.902** [0.9022, 0.9022]	**0.867** [0.8667, 0.8667]	**0.962** [0.9624, 0.9624]
	CAKE$^{(HM)}$	**0.902** [0.9022, 0.9022]	**0.867** [0.8667, 0.8667]	**0.962** [0.9624, 0.9624]
S5	Full	0.593 [0.4434, 0.7435]	0.660 [0.5739, 0.7462]	0.782 [0.6384, 0.9254]
	Random	0.549 [0.3891, 0.7089]	0.634 [0.5415, 0.7267]	0.743 [0.5933, 0.8933]
	Consensus	0.479 [0.4780, 0.4807]	0.570 [0.5689, 0.5710]	0.708 [0.7046, 0.7116]
	C component	0.688 [0.5929, 0.7831]	0.697 [0.6399, 0.7540]	0.867 [0.7988, 0.9355]
	$\tilde{S}$ component	0.840 [0.6821, 0.9980]	0.855 [0.7557, 0.9535]	0.894 [0.7637, 1.0000]
	CAKE$^{(PR)}$	0.908 [0.9075, 0.9075]	0.879 [0.8792, 0.8792]	0.966 [0.9664, 0.9664]
	CAKE$^{(HM)}$	**0.925** [0.9245, 0.9245]	**0.898** [0.8979, 0.8979]	**0.973** [0.9729, 0.9729]
S6	Full	0.303 [0.3006, 0.3045]	0.473 [0.4720, 0.4748]	0.681 [0.6794, 0.6834]
	Random	0.315 [0.2978, 0.3325]	0.480 [0.4690, 0.4920]	0.689 [0.6748, 0.7032]
	Consensus	0.561 [0.5537, 0.5684]	0.586 [0.5780, 0.5943]	0.779 [0.7712, 0.7859]
	C component	0.565 [0.5640, 0.5654]	0.590 [0.5896, 0.5907]	0.782 [0.7815, 0.7828]
	$\tilde{S}$ component	0.566 [0.5506, 0.5811]	0.590 [0.5770, 0.6036]	0.781 [0.7654, 0.7960]
	CAKE$^{(PR)}$	0.558 [0.5302, 0.5860]	0.583 [0.5573, 0.6095]	0.777 [0.7524, 0.8006]
	CAKE$^{(HM)}$	**0.573** [0.5662, 0.5798]	**0.596** [0.5893, 0.6034]	**0.789** [0.7817, 0.7962]
S7	Full	0.615 [0.4786, 0.7523]	0.649 [0.5642, 0.7333]	0.802 [0.6727, 0.9317]
	Random	0.602 [0.4685, 0.7363]	0.636 [0.5464, 0.7253]	0.797 [0.6691, 0.9257]
	Consensus	0.460 [0.4371, 0.4820]	0.545 [0.5385, 0.5519]	0.679 [0.6411, 0.7165]
	C component	0.450 [0.4267, 0.4725]	0.540 [0.5326, 0.5480]	0.668 [0.6329, 0.7035]
	$\tilde{S}$ component	0.713 [0.5401, 0.8869]	0.762 [0.6529, 0.8713]	0.820 [0.6817, 0.9582]
	CAKE$^{(PR)}$	**0.861** [0.7566, 0.9662]	**0.861** [0.7908, 0.9309]	**0.939** [0.8647, 1.0000]
	CAKE$^{(HM)}$	0.678 [0.4982, 0.8581]	0.741 [0.6239, 0.8579]	0.797 [0.6623, 0.9308]
Ir	Full	0.692 [0.6255, 0.7580]	0.728 [0.6907, 0.7651]	0.857 [0.7859, 0.9288]
	Random	0.643 [0.5586, 0.7277]	0.697 [0.6525, 0.7425]	0.823 [0.7323, 0.9134]
	Consensus	0.658 [0.5951, 0.7206]	0.733 [0.7047, 0.7620]	0.835 [0.7681, 0.9024]
	C component	0.658 [0.5951, 0.7206]	0.733 [0.7047, 0.7620]	0.835 [0.7681, 0.9024]
	$\tilde{S}$ component	0.824 [0.6491, 0.9985]	0.871 [0.7671, 0.9741]	0.862 [0.7138, 1.0000]
	CAKE$^{(PR)}$	0.855 [0.7552, 0.9541]	0.865 [0.8118, 0.9190]	**0.923** [0.8345, 1.0000]
	CAKE$^{(HM)}$	**0.874** [0.7222, 1.0000]	**0.900** [0.8097, 0.9908]	0.909 [0.7849, 1.0000]
Bc	Full	0.663 [0.6558, 0.6698]	0.543 [0.5343, 0.5518]	0.908 [0.9058, 0.9101]
	Random	0.660 [0.6598, 0.6598]	0.534 [0.5337, 0.5337]	0.907 [0.9070, 0.9070]
	Consensus	**0.779** [0.7788, 0.7789]	**0.679** [0.6773, 0.6815]	0.942 [0.9422, 0.9422]

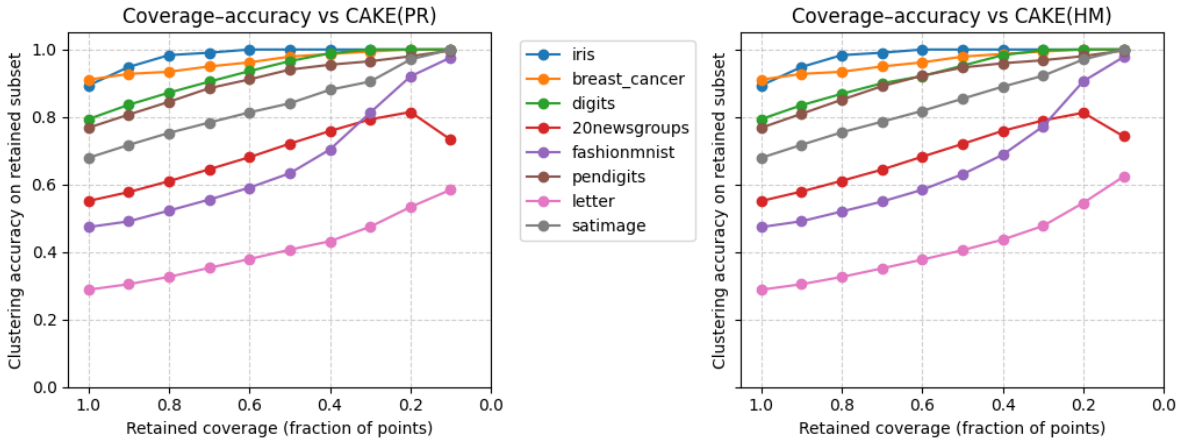
	C component	**0.779** [0.7788, 0.7789]	**0.679** [0.6773, 0.6815]	0.942 [0.9422, 0.9422]
	$\tilde{S}$ component	0.754 [0.7543, 0.7543]	0.652 [0.6516, 0.6516]	**0.950** [0.9497, 0.9497]
	CAKE$^{(PR)}$	0.754 [0.7543, 0.7543]	0.652 [0.6516, 0.6516]	**0.950** [0.9497, 0.9497]
	CAKE$^{(HM)}$	0.754 [0.7543, 0.7543]	0.652 [0.6516, 0.6516]	**0.950** [0.9497, 0.9497]
	Full	0.630 [0.6032, 0.6568]	0.731 [0.7204, 0.7421]	0.747 [0.7153, 0.7785]
	Random	0.594 [0.5726, 0.6145]	0.712 [0.7009, 0.7240]	0.713 [0.6918, 0.7351]
	Consensus	0.777 [0.7201, 0.8344]	0.835 [0.8094, 0.8614]	0.799 [0.7503, 0.8484]
DG	C component	0.774 [0.7250, 0.8239]	0.834 [0.8111, 0.8569]	0.784 [0.7296, 0.8376]
	$\tilde{S}$ component	0.763 [0.7175, 0.8092]	0.852 [0.8316, 0.8725]	0.779 [0.7312, 0.8268]
	CAKE$^{(PR)}$	0.741 [0.6807, 0.8013]	0.842 [0.8126, 0.8711]	0.744 [0.6829, 0.8055]
	CAKE$^{(HM)}$	**0.793** [0.7466, 0.8392]	**0.866** [0.8452, 0.8875]	**0.815** [0.7730, 0.8575]
	Full	0.352 [0.3421, 0.3610]	0.509 [0.5000, 0.5179]	0.511 [0.4957, 0.5259]
	Random	0.359 [0.3505, 0.3675]	0.516 [0.5080, 0.5232]	0.515 [0.4975, 0.5332]
	Consensus	0.462 [0.4500, 0.4735]	0.586 [0.5819, 0.5895]	0.575 [0.5581, 0.5922]
NG	C component	0.473 [0.4607, 0.4863]	0.590 [0.5831, 0.5977]	0.578 [0.5586, 0.5973]
	$\tilde{S}$ component	0.467 [0.4437, 0.4903]	0.609 [0.5998, 0.6183]	0.561 [0.5322, 0.5892]
	CAKE$^{(PR)}$	**0.494** [0.4766, 0.5123]	**0.618** [0.6118, 0.6234]	**0.603** [0.5820, 0.6246]
	CAKE$^{(HM)}$	0.467 [0.4437, 0.4903]	0.609 [0.5998, 0.6183]	0.561 [0.5322, 0.5892]
	Full	0.353 [0.3416, 0.3641]	0.506 [0.4966, 0.5163]	0.514 [0.4880, 0.5391]
	Random	0.362 [0.3495, 0.3746]	0.509 [0.4994, 0.5193]	0.537 [0.5127, 0.5611]
	Consensus	0.433 [0.4199, 0.4458]	0.560 [0.5533, 0.5666]	0.596 [0.5825, 0.6097]
FM	C component	0.427 [0.4111, 0.4421]	0.560 [0.5517, 0.5686]	0.596 [0.5742, 0.6177]
	$\tilde{S}$ component	0.447 [0.4295, 0.4637]	0.585 [0.5740, 0.5961]	0.568 [0.5357, 0.5999]
	CAKE$^{(PR)}$	0.445 [0.4149, 0.4744]	0.588 [0.5748, 0.6012]	0.566 [0.5274, 0.6046]
	CAKE$^{(HM)}$	**0.474** [0.4467, 0.5007]	**0.595** [0.5791, 0.6111]	**0.599** [0.5586, 0.6402]
	Full	0.541 [0.5121, 0.5690]	0.676 [0.6667, 0.6846]	0.669 [0.6326, 0.7057]
	Random	0.566 [0.5336, 0.5976]	0.682 [0.6725, 0.6917]	0.712 [0.6633, 0.7611]
	Consensus	0.653 [0.6190, 0.6874]	0.734 [0.7210, 0.7467]	0.718 [0.6904, 0.7451]
PD	C component	0.626 [0.5952, 0.6563]	0.721 [0.7078, 0.7345]	0.704 [0.6768, 0.7316]
	$\tilde{S}$ component	0.754 [0.6854, 0.8226]	**0.842** [0.8137, 0.8695]	0.803 [0.7354, 0.8708]
	CAKE$^{(PR)}$	0.724 [0.6749, 0.7740]	0.819 [0.7981, 0.8406]	0.784 [0.7372, 0.8309]
	CAKE$^{(HM)}$	**0.759** [0.7166, 0.8010]	**0.842** [0.8242, 0.8597]	**0.810** [0.7665, 0.8536]
	Full	0.152 [0.1456, 0.1577]	0.368 [0.3634, 0.3727]	0.281 [0.2751, 0.2869]
	Random	0.152 [0.1476, 0.1557]	0.373 [0.3675, 0.3794]	0.284 [0.2790, 0.2888]
	Consensus	0.199 [0.1939, 0.2031]	0.430 [0.4264, 0.4333]	0.324 [0.3177, 0.3306]
LT	C component	0.210 [0.2031, 0.2169]	0.440 [0.4348, 0.4461]	0.333 [0.3230, 0.3434]
	$\tilde{S}$ component	0.213 [0.2034, 0.2234]	0.458 [0.4492, 0.4665]	0.334 [0.3203, 0.3472]
	CAKE$^{(PR)}$	0.219 [0.2096, 0.2277]	0.458 [0.4495, 0.4659]	0.328 [0.3176, 0.3380]
	CAKE$^{(HM)}$	**0.222** [0.2163, 0.2267]	**0.460** [0.4555, 0.4643]	**0.335** [0.3277, 0.3422]
	Full	0.510 [0.4657, 0.5550]	0.595 [0.5556, 0.6346]	0.663 [0.6298, 0.6972]
	Random	0.520 [0.4927, 0.5472]	0.608 [0.5895, 0.6273]	0.672 [0.6542, 0.6894]
	Consensus	0.523 [0.4794, 0.5669]	0.575 [0.5549, 0.5945]	0.706 [0.6551, 0.7569]
SA	C component	0.523 [0.4794, 0.5669]	0.575 [0.5549, 0.5945]	0.706 [0.6551, 0.7569]
	$\tilde{S}$ component	**0.710** [0.7080, 0.7125]	**0.764** [0.7584, 0.7703]	**0.799** [0.7842, 0.8138]
	CAKE$^{(PR)}$	0.682 [0.6383, 0.7249]	0.749 [0.7268, 0.7704]	0.772 [0.7317, 0.8129]
	CAKE$^{(HM)}$	0.653 [0.5997, 0.7067]	0.734 [0.7087, 0.7594]	0.750 [0.6964, 0.8037]

Table 2: ARI, AMI, ACC (clustering accuracy) averages and t-95% confidence intervals, computed over multiple independent clustering runs with varying random seeds. Results are reported for the full datasets (Full) and their filtered subsets mentioned in §5.2. Highest values in **bold**.

Coverage-accuracy evaluation. CAKE is evaluated as a pointwise reliability score for ensemble clustering by measuring clustering accuracy as a function of retained coverage. For each dataset, per-instance CAKE$^{(PR)}$ and CAKE$^{(HM)}$ scores are computed from an ensemble of R=20 k-means partitions. A reference k-means solution is fitted on the full dataset, and its accuracy is evaluated against ground truth after label alignment. For a grid of coverage levels $c \in [0.1, 1.0]$, instances are ranked by CAKE and the reference accuracy restricted to the top-c fraction of instances is measured. Plotting accuracy as a function of coverage yields coverage–accuracy curves, which show how well each CAKE variant orders instances from easier to harder and enables a direct comparison of CAKE$^{(PR)}$ and CAKE$^{(HM)}$ (Fig. 7). Sharper curves (higher accuracy at lower coverage) indicate that CAKE concentrates correctly assigned points among the highest scores (see also Appendix B Fig. 13 for a percentile-based clustering accuracy view).



(a) Coverage–accuracy on **synthetic** datasets: S1, S2, S3, S4, S5, S6, S7.



(b) Coverage–accuracy on **real** datasets: IR, BC, DG, NG, FM, PD, LT, SA.

Figure 7: Clustering accuracy on the retained subset as a function of retained coverage, using CAKE$^{(PR)}$ (left) and CAKE$^{(HM)}$ (right). Top (7a): synthetic datasets. Bottom (7b): real datasets.

AUROC/AUPRC on consensus correctness. CAKE is evaluated by its ability to distinguish *correctly* clustered instances from *incorrect* ones. For each dataset, a consensus partition $z^\star$ is first computed from the clustering ensemble and a binary correctness label $\mathbb{1}\left[\pi(z_i^\star) = y_i\right]$ is defined, where π denotes the Hungarian alignment mapping from consensus labels to ground-truth classes. Given a confidence score (higher indicates higher reliability), discriminative power is measured using AUROC (area under the ROC curve) and AUPRC (area under the precision-recall curve, i.e., average precision) when predicting $\mathbb{1}\left[\pi(z_i^\star) = y_i\right]=1$. Four signals are compared: (*i*) CAKE$^{(PR)}$ and (*ii*) CAKE$^{(HM)}$, which are computed from the same ensemble. (*iii*) *Entropy-based agreement*: for each run r, its labels are aligned to $z^\star$, yielding aligned labels $\tilde{z}^{(r)}$. This induces, for each point, an empirical distribution $p_{i\ell} = \frac{1}{R}\sum_{r=1}^{R} \mathbb{1}\left[\tilde{z}_i^{(r)} = \ell\right]$, with normalized entropy $H_i = -\sum_{\ell=1}^{k} p_{i\ell} \log\big(p_{i\ell} + \varepsilon\big) \Rightarrow \hat{H}_i = 1 - (H_i/\log k)$, so that $\hat{H}_i \in [0,1]$ is high when aligned votes concentrate on a single label and low when the ensemble is uncertain. (*iv*) *Bootstrap-based stability*: B subsamples of size $\lfloor 0.8n \rfloor$ are drawn and k-means is run on each subsample, obtaining partitions $\{z^{(b)}\}_{b=1}^{B}$. For each point i, all pairs of subsamples (b, b') in which i appears are considered. For each such pair, $z^{(b')}$ is aligned to $z^{(b)}$ on the points shared by the two subsamples via Hungarian matching, and agreement is counted according to whether i receives the same label in the aligned pair. The score is the average pairwise agreement: $\text{boot}_i = \frac{1}{|\mathcal{P}_i|}\sum_{(b,b')\in\mathcal{P}_i} \mathbb{1}\left[z_i^{(b)} = \tilde{z}_i^{(b')}\right]$, where $\mathcal{P}_i$ is the set of subsample-pairs containing i, and $\tilde{z}^{(b')}$ denotes the labels of run b' after alignment to b.

B=20 is used to match the default ensemble size R=20, yielding a comparable computational budget between baselines. This evaluation (Table 3) quantifies discriminative power at the point level (see Appendix B Tables 8–10 for the same evaluation using MiniBatchKMeans/K-Medoids/GMM). As a complement to Table 3, it is also tested whether each signal yields a monotonic reliability ordering. Points are binned by score percentiles and Spearman's correlation ($\rho_{\text{Spear.}}$) between percentile rank and within-bin clustering accuracy on representative datasets is reported (Fig. 8).

Dataset	Metric	CAKE$^{(PR)}$	CAKE$^{(HM)}$	Entropy agreement	Bootstrap stability
S1	AUPRC	**0.997**	**0.997**	0.958	0.960
	AUROC	**0.932**	**0.932**	0.506	0.519
S2	AUPRC	0.930	**0.931**	0.785	0.807
	AUROC	0.761	**0.764**	0.488	0.553
S3	AUPRC	**0.932**	**0.932**	0.852	0.819
	AUROC	0.842	**0.844**	0.723	0.665
S4	AUPRC	**0.976**	**0.976**	0.887	0.889
	AUROC	**0.843**	**0.843**	0.514	0.528
S5	AUPRC	0.978	**0.980**	0.904	0.908
	AUROC	0.823	**0.834**	0.499	0.509
S6	AUPRC	0.923	**0.924**	0.763	0.754
	AUROC	0.821	**0.823**	0.657	0.621
S7	AUPRC	0.975	**0.977**	0.933	0.919
	AUROC	0.817	**0.826**	0.602	0.575
IR	AUPRC	0.977	**0.984**	0.859	0.923
	AUROC	0.840	**0.888**	0.328	0.675
BC	AUPRC	**0.971**	**0.971**	0.915	0.929
	AUROC	**0.773**	**0.773**	0.555	0.636
DG	AUPRC	0.909	**0.912**	0.734	0.869
	AUROC	0.767	**0.777**	0.526	0.717
NG	AUPRC	**0.638**	0.624	0.623	0.637
	AUROC	**0.678**	0.670	0.636	0.637
FM	AUPRC	**0.758**	0.755	0.631	0.751
	AUROC	0.707	0.703	0.667	**0.742**
PD	AUPRC	0.951	**0.954**	0.900	0.864
	AUROC	0.857	**0.866**	0.769	0.718
LT	AUPRC	**0.465**	**0.465**	0.402	0.409
	AUROC	0.661	**0.664**	0.617	0.602
SA	AUPRC	0.890	**0.894**	0.706	0.725
	AUROC	0.784	**0.788**	0.560	0.592

Table 3: Per-point *consensus correctness* prediction (AUPRC and AUROC) with CAKE$^{(PR)}$, CAKE$^{(HM)}$, entropy-based agreement, and bootstrap-based stability. The highest value for each dataset and metric is highlighted in **bold**.

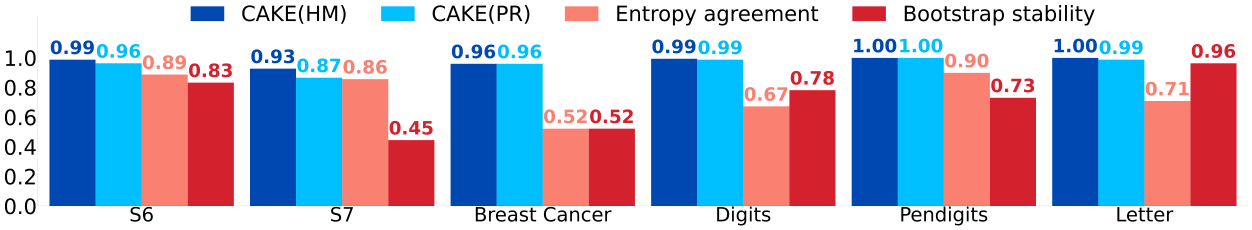


Figure 8: Correlation ($\rho_{\text{Spear.}}$) between confidence-score percentiles and per-bin clustering accuracy.

Beyond the ensemble-derived baselines above, additional comparisons were conducted against native soft and density-based confidence signals. In particular, two Fuzzy C-Means (FCM) confidence scores were considered: (*i*) maximum membership, defined as the largest membership weight assigned to a point $\mathrm{FCM}_i^{(1)} = \max_\ell u_{i\ell}$, where $u_{i\ell}$ is the FCM membership of point x_i to cluster ℓ, and (*ii*) entropy-based confidence, defined as one minus the normalized entropy of the membership vector $\mathrm{FCM}_i^{(2)} = 1 - \frac{H_i}{\log k}$, where $H_i = -\sum_{\ell=1}^{k} u_{i\ell} \log\big(u_{i\ell} + \varepsilon\big)$. In addition, two density-based baselines were examined. The first is a standard global Local Outlier Factor (LOF) (Smiti, 2020) inlierness score $\mathrm{LOF}_i^{(1)}$, obtained from the negative outlier factor so that larger values indicate more typical (less outlier-like) observations. The second is an ensemble cluster-local LOF baseline $\mathrm{LOF}_i^{(2)}$, in which a LOF-based score is computed within the assigned cluster of each run in the same ensemble used by CAKE, and then averaged across runs. This yields a cluster-aware, ensemble-averaged density-based confidence score. Comparisons of these baselines against $\mathrm{CAKE}^{(\mathrm{HM})}$ are reported in Table 4. Supplementary comparisons against native GMM-based confidence signals, with $\mathrm{CAKE}^{(\mathrm{HM})}$ computed from GMM ensembles, are provided in Appendix E.

Dataset	Metric	$\mathrm{CAKE}^{(\mathrm{HM})}$	$\mathrm{FCM}^{(1)}$	$\mathrm{FCM}^{(2)}$	$\mathrm{LOF}^{(1)}$	$\mathrm{LOF}^{(2)}$
S1	AUPRC	**0.997**	0.995	0.994	0.966	0.970
	AUROC	**0.932**	0.911	0.895	0.541	0.619
S2	AUPRC	**0.931**	0.878	0.869	0.769	0.774
	AUROC	**0.764**	0.682	0.655	0.442	0.459
S3	AUPRC	0.932	**0.961**	**0.961**	0.680	0.716
	AUROC	0.844	0.942	**0.943**	0.550	0.587
S4	AUPRC	**0.976**	0.970	0.969	0.891	0.910
	AUROC	**0.843**	0.826	0.819	0.515	0.598
S5	AUPRC	0.980	**0.982**	**0.982**	0.954	0.943
	AUROC	0.834	0.855	**0.857**	0.682	0.631
S6	AUPRC	0.924	**0.931**	0.929	0.742	0.782
	AUROC	0.823	**0.832**	0.827	0.617	0.643
S7	AUPRC	**0.977**	0.970	0.970	0.941	0.944
	AUROC	**0.826**	0.767	0.768	0.635	0.630
Ir	AUPRC	0.984	**0.993**	0.992	0.890	0.902
	AUROC	0.888	**0.940**	0.926	0.500	0.559
Bc	AUPRC	0.971	**0.985**	**0.985**	0.948	0.960
	AUROC	0.773	**0.889**	**0.889**	0.639	0.704
Dg	AUPRC	**0.912**	0.686	0.698	0.775	0.789
	AUROC	**0.777**	0.432	0.431	0.599	0.634
Ng	AUPRC	**0.624**	0.489	0.489	0.598	0.598
	AUROC	**0.670**	0.459	0.456	0.589	0.608
Fm	AUPRC	**0.755**	0.578	0.575	0.604	0.613
	AUROC	**0.703**	0.498	0.504	0.583	0.587
Pd	AUPRC	**0.954**	0.939	0.930	0.783	0.822
	AUROC	**0.866**	0.823	0.807	0.537	0.598
Lt	AUPRC	**0.465**	0.227	0.229	0.301	0.333
	AUROC	**0.664**	0.381	0.385	0.514	0.546
Sa	AUPRC	0.894	**0.898**	0.896	0.707	0.713
	AUROC	0.788	**0.795**	0.788	0.555	0.561

Table 4: Per-point *consensus correctness* prediction (AUPRC and AUROC) with $\mathrm{CAKE}^{(\mathrm{HM})}$, $\mathrm{FCM}^{(1)}$ (max membership), $\mathrm{FCM}^{(2)}$ (entropy confidence), $\mathrm{LOF}^{(1)}$ (global LOF), and $\mathrm{LOF}^{(2)}$ (ensemble-averaged cluster-local LOF). The highest value for each dataset and metric is shown in **bold**.

Error discovery via confidence ranking. CAKE is tested for flagging clustering errors. For each dataset, a k-means++ model is fitted at the ground-truth k, predicted labels are aligned to ground truth, and misclustered points are marked as positives ("errors"). Three scores are compared: CAKE$^{(HM)}$ (Eq. 8b; R=20 random k-means runs with centroid-based Silhouette approximation), *GMM* $p_{\max}$ (max posterior under a k-component GMM), and their rank-average fusion. Ranking points from low to high confidence, AUPRC on the error class is reported (Fig. 9).

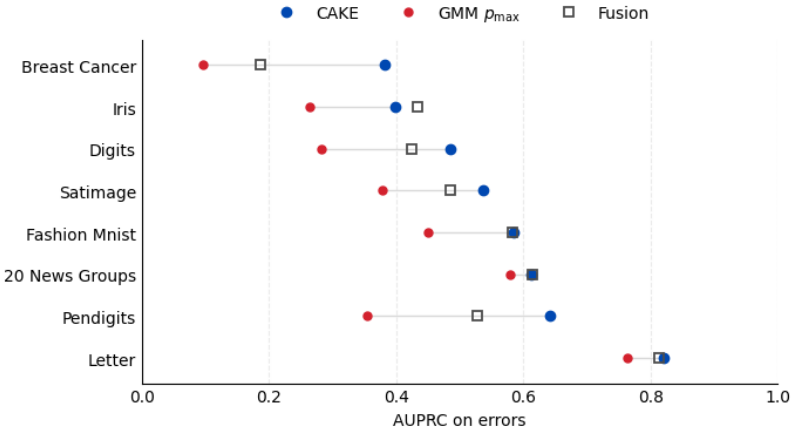
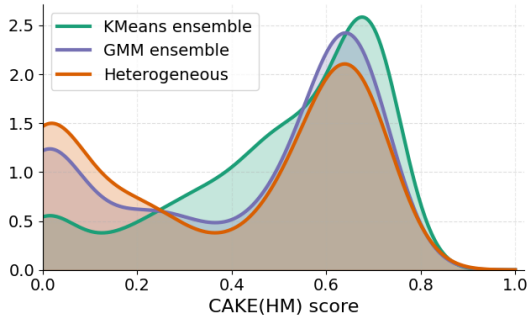
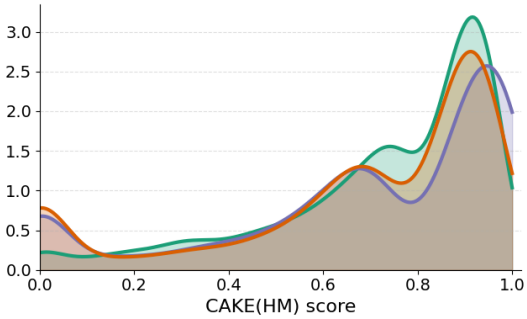


Figure 9: AUPRC on the error class. Markers show per-dataset AUPRC; light gray connectors join GMM $p_{\max}$ to CAKE to highlight the per-dataset CAKE score improvement. A simple fusion of these two is marked by □.

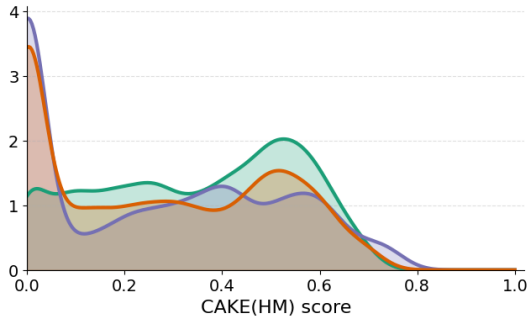
Beyond *k*-means clustering. CAKE is model-agnostic: it can be applied to ensembles produced by a range of base algorithms. To illustrate this, CAKE$^{(HM)}$ is computed under three ensemble constructions: (*i*) homogeneous k-means, (*ii*) homogeneous GMM, and (*iii*) heterogeneous k-means+GMM, on representative datasets (Fig. 10). A broader ablation on clustering ensemble design and its effect on CAKE score (ranking) quality is provided in Appendix C.



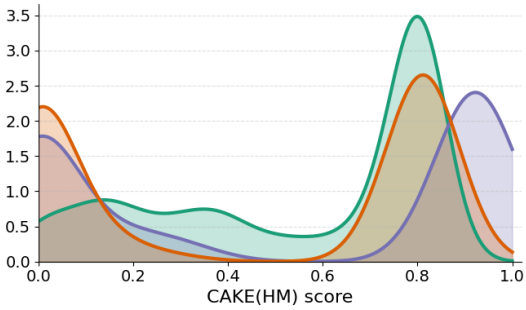
(a) Breast Cancer



(b) Synthetic S5



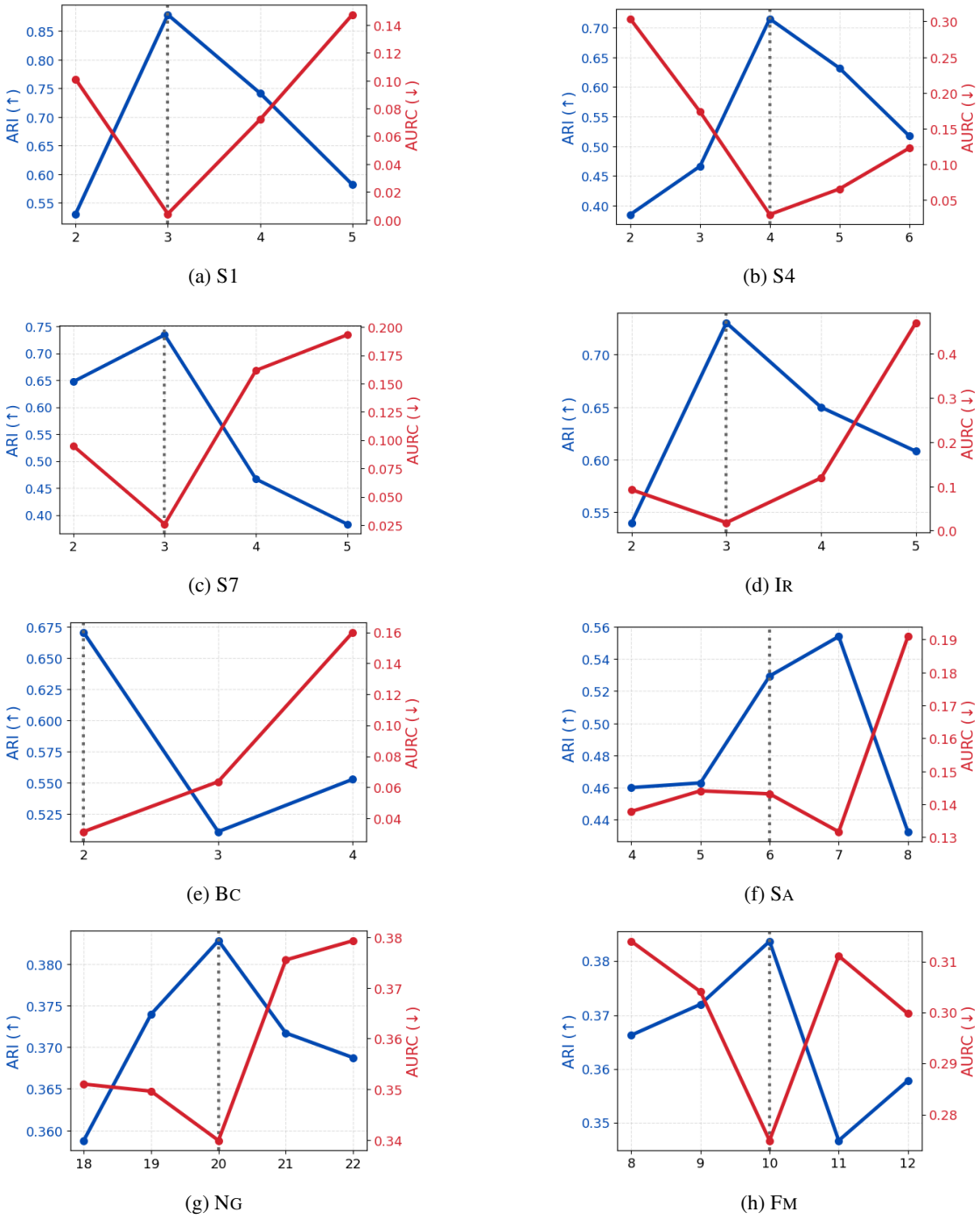
(c) Pendigits



(d) Synthetic S6

Figure 10: Distributions (KDEs) of CAKE$^{(HM)}$ (Eq. 8b) under: homogeneous k-means ensembles, homogeneous GMM ensembles, and heterogeneous k-means+GMM ensembles.

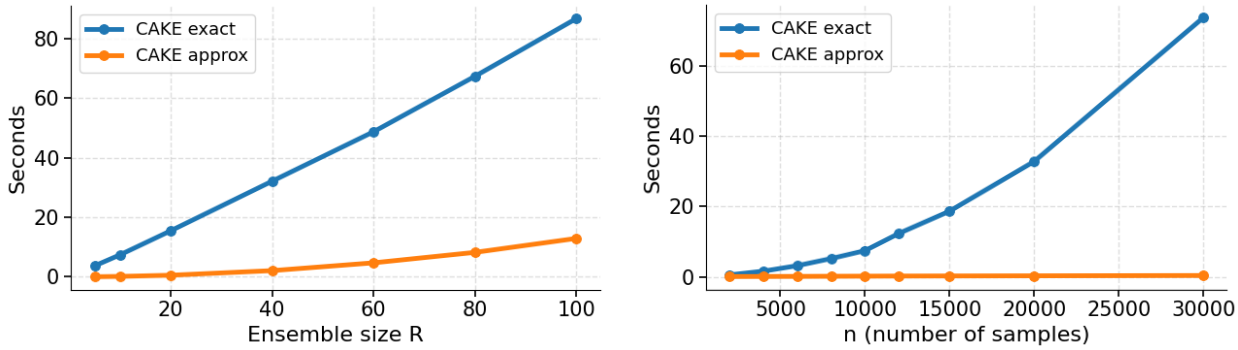
Misspecified number of clusters. CAKE is tested under cluster-count misspecification, i.e., when the clustering uses a suboptimal number of clusters. For each $k' \in \{k-2, \ldots, k+2\}$, where k is the ground-truth number of clusters (Tab. 1), (*i*) an ensemble of R=30 k-means runs with k' clusters is constructed and CAKE$^{(HM)}$ scores are computed, and (*ii*) an independent k-means++ model with the same k' is fitted for evaluation. The following are reported: (a) clustering quality via ARI with respect to ground truth and (b) CAKE ranking quality via AURC (area under the risk–coverage curve), where risk is the misclustering rate after Hungarian alignment. Higher ARI indicates better partitions, while lower AURC indicates that CAKE concentrates correct assignments among high-confidence points (Fig. 11).



(a) S1 (b) S4 (c) S7 (d) IR (e) BC (f) SA (g) NG (h) FM

Figure 11: **Misspecified k.** ARI and AURC vs. k'. Vertical dotted line marks the ground-truth k.

Runtime. The computational cost of CAKE (Eq. 8; Alg. 1) and the benefit of the centroid-based Silhouette proxy (§3.2) are quantified by measuring runtime as a function of ensemble size R and dataset size n on synthetic Gaussian data (Fig. 12). As shown in Fig. 12a (Table 5), runtime grows approximately linearly with R. Figure 12b (Table 6) shows that, when using the centroid-based proxy, runtime scales approximately linearly with n, whereas the exact Silhouette variant increases much more rapidly. Overall, these trends align with the complexity analysis in §3.4 and confirm that the proxy substantially reduces runtime while preserving high agreement with the exact scores (high Pearson correlation and low Mean Absolute Error; Tables 5–6).



(a) Runtime (s) vs. R (ensemble size)

(b) Runtime (s) vs. n (number of samples)

Figure 12: Runtime in seconds (s) for CAKE$^{(HM)}$ (Eq. 8b) using exact pairwise Silhouette scores vs. the centroid-based Silhouette proxy on synthetic data. **Left** (a): runtime as a function of ensemble size R (n=10,000, d=20, k=10; exact runtimes are reported in Table 5). **Right** (b): runtime as a function of the number of samples n (d=20, k=10, R=20; exact runtimes are reported in Table 6). Experiments were run on a cloud-hosted notebook with ~51 GB RAM.

R	Ensemble construction (s)	CAKE exact (s)	CAKE approx. (s)	Pearson corr.	MAE
5	0.04	3.73	0.05	0.974	0.072
10	0.06	7.42	0.16	0.972	0.069
20	0.13	15.49	0.56	0.975	0.062
40	0.25	32.13	2.10	0.955	0.066
60	0.37	48.71	4.71	0.944	0.061
80	0.49	67.32	8.22	0.936	0.062
100	0.63	86.69	12.91	0.914	0.063

Table 5: Runtime in seconds (s) vs. ensemble size R (Fig. 12a); Pearson correlation and Mean Absolute Error between exact and centroid-based Silhouette-proxy CAKE scores.

n	Ensemble construction (s)	CAKE exact (s)	CAKE approx. (s)	Pearson corr.	MAE
2,000	0.02	0.53	0.06	0.977	0.065
4,000	0.05	1.59	0.08	0.989	0.068
8,000	0.06	5.18	0.14	0.974	0.077
10,000	0.07	7.44	0.17	0.972	0.069
15,000	0.08	18.69	0.22	0.994	0.054
20,000	0.10	32.84	0.26	0.971	0.063
30,000	0.14	73.84	0.35	0.982	0.056

Table 6: Runtime in seconds (s) vs. of number of samples n (Fig. 12b); Pearson correlation and Mean Absolute Error between exact and centroid-based Silhouette-proxy CAKE scores.

5.3. Empirical Results

Across synthetic and real-world datasets, clustering on the subset retained by CAKE consistently improves ARI, AMI, and ACC relative to using all points (top 70%; Table 2, with an adaptive-thresholding approach reported in

Appendix D Table 13), with similar gains in Silhouette and NMI (Appendix Table 7). These improvements are typically accompanied by tighter confidence intervals, indicating reduced run-to-run variability.

Although the strongest filtering criterion can be dataset-dependent, particularly when geometry (Eq. 3) or stability (Eq. 7) dominates, CAKE (Eq. 8) provides the best overall trade-off as a confidence-based ranking criterion: it is best or tied-best for most dataset–metric combinations and remains near the top even when a baseline criterion is optimal on a given dataset (Tables 2, 7).

The component baselines ($\tilde{S}$, C) can be competitive when one signal is highly informative. For instance, the geometric component alone is strongest on datasets where local separation is the main determinant of cluster assignments (e.g., S3 and SA, where $\tilde{S}$ attains ARI 0.795 and 0.710 respectively), while agreement-focused signals can lead when cross-run consistency is most decisive (e.g., BC, where Consensus/C perform best in ARI/AMI). However, these signals can fail in complementary ways: a point may be geometrically well-placed in each run but switch clusters across runs, or be consistently assigned yet weakly supported geometrically (Fig. 1). CAKE reduces these failure modes by requiring both stable assignments and consistent geometric support under the learned cluster structure, yielding a more robust confidence signal than either component alone. CAKE outperforms both components on several synthetic datasets (S4–S6) and is also the top-performing filtering choice on most real datasets across ARI/AMI/ACC (e.g., CAKE$^{(HM)}$ on DG, FM, PD, and LT; CAKE$^{(PR)}$ on NG). Overall, these trends further support the central claim that combining stability and geometry yields a more reliable confidence ranking than either component alone.

CAKE produces an informative pointwise ranking. Coverage–accuracy curves show that clustering accuracy improves as a function of retained coverage when points are ranked by CAKE scores (Fig. 7), indicating that CAKE concentrates correctly clustered instances among the highest scores. This behavior is also visible in the percentile/decile analysis: accuracy rises nearly monotonically with CAKE score percentiles on both synthetic and real datasets (Appendix Fig. 13). Overall, CAKE behaves as a reliability ordering from ambiguous/borderline points to stable core members.

CAKE enables pointwise correctness prediction and effective error discovery. When predicting whether a consensus assignment is correct (after Hungarian alignment to ground truth), CAKE achieves strong discriminative performance and typically outperforms entropy-based agreement and bootstrap-based stability baselines in both AUPRC and AUROC (Table 3). A similar picture emerges when the comparison is extended to native soft and density-based confidence signals (Table 4). CAKE$^{(HM)}$ remains among the strongest overall methods, attaining the highest value on $4/7$ synthetic and $5/8$ real datasets across both metrics. The Fuzzy C-Means scores FCM$^{(1)}$ and FCM$^{(2)}$ are competitive, showing that soft memberships can already provide informative pointwise confidence when ambiguity is well captured by a single fuzzy partition. The density-based baselines are weaker overall. In particular, the ensemble cluster-local score LOF$^{(2)}$ is consistently stronger than the global score LOF$^{(1)}$, indicating that (ensemble-averaged) cluster-aware local density is more informative than dataset-level outlierness alone, but both remain below CAKE$^{(HM)}$ in overall ranking quality. Related trends are also observed in the additional GMM-based comparison reported in Appendix E, where CAKE$^{(HM)}$ outperforms the native GMM-based confidence signals on most of the datasets considered (Appendix Table 14). CAKE also yields a more monotonic reliability ordering than the ensemble-derived baselines: score percentiles correlate strongly with within-bin clustering accuracy (Fig. 8). In error discovery experiments, ranking by CAKE concentrates misclustered points more effectively than GMM posterior confidence $p_{\max}$ on most real datasets, while a simple fusion of the two signals is often competitive (Fig. 9).

CAKE scores stabilize with moderate ensemble sizes. As the ensemble size R increases, per-point score variability steadily decreases and the induced ranking becomes essentially stable, with diminishing returns beyond about $R \approx 30$–40 (Fig. 6). This indicates that a relatively small ensemble is sufficient to obtain a reliable confidence ordering.

Under misspecified cluster counts, CAKE's ranking quality also closely tracks clustering quality. Across datasets, Adjusted Rand Index (ARI) typically peaks at (or very near) the ground-truth number of clusters k, and the AURC is minimized at the same (or neighboring) k; equivalently, the k values that yield the most accurate clustering also yield the sharpest CAKE ranking (Fig. 11).

CAKE is model-agnostic. Across homogeneous k-means, GMM, and heterogeneous k-means & GMM clustering ensembles (as well as spectral clustering ensembles for non-convex geometries; Fig. 4), CAKE score distributions are qualitatively similar and remain informative and well-shaped (Fig. 10), supporting that CAKE behaves consistently beyond purely k-means ensembles. A more detailed analysis of ensemble diversity is reported in Appendix C, showing that CAKE remains informative across a broad range of homogeneous and heterogeneous ensemble constructions

(Appendix Fig. 14), while the effect of additional diversity depends on whether it contributes useful complementary variation or simply introduces less stable clustering behaviour. As expected for any ensemble-derived confidence score, CAKE reflects the inductive biases of the base clustering algorithm, since it is intended to measure instance reliability with respect to that procedure and therefore remains comparable across different ensemble constructions.

CAKE is computationally practical. Its runtime follows the expected scaling trends (§3.4). It grows roughly linearly with the ensemble size R and, when using the centroid-based Silhouette proxy, approximately linearly with the dataset size (Fig. 12). The approximation closely matches the exact computation (high correlation and low MAE; Tables 5–6) while substantially reducing cost, making CAKE feasible for large datasets.

6. Conclusion

CAKE provides an interpretable, label-free per-point confidence score by fusing cross-run assignment stability with local geometric fit. It is simple to compute from a modest ensemble of standard clustering runs and yields a scalar in $[0, 1]$ for every instance. More broadly, CAKE shows that ensemble diversity, long used for uncertainty estimation in supervised learning, transfers to unsupervised clustering: variability across runs provides a practical per-point confidence signal.

Promising directions include calibrated confidence (e.g., with a small labeled set), integration into semi- and self-supervised pipelines (e.g., uncertainty-aware pseudo-labeling/per-sample weighting), guiding the selection of k via score distributions, and extending CAKE to multi-k clustering ensembles where k varies across runs. By making instance-level uncertainty explicit, CAKE provides an interpretable pointwise confidence score for clustering assignments, enabling more reliable ranking, prioritization, and downstream use of clustering in labeling, anomaly detection, and decision pipelines.

CRediT authorship statement

Aggelos Semoglou: Conceptualization, Methodology, Software, Formal analysis, Writing - original draft.
John Pavlopoulos: Conceptualization, Methodology, Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Synthetic datasets and code are available at https://github.com/semoglou/cake and https://pypi.org/project/cake-ensemble/. The real-world datasets used in this study are publicly accessible via OpenML, SCIKIT-LEARN, and TensorFlow.

Acknowledgments

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems [Computer software]. *TensorFlow*. https://www.tensorflow.org/

Aggarwal, C. C., & Reddy, C. K. (2013). *Data clustering: Algorithms and applications*. CRC Press.

Aktas, D., Lokman, B., Inkaya, T., & Dejaegere, G. (2024). Cluster ensemble selection and consensus clustering: A multi-objective optimization approach. *European Journal of Operational Research*, *314*(3), 1065–1077. https://doi.org/10.1016/j.ejor.2023.10.029

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46*(1), 243–256.

Ayad, H. G., & Kamel, M. S. (2008). Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(1), 160–173. https://doi.org/10.1109/TPAMI.2007.1138

Beluch, W. H., Genewein, T., Nürnberger, A., & Köhler, J. M. (2018). The power of ensembles for active learning in image classification. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9368–9377. https://doi.org/10.1109/CVPR.2018.00976

Ben-Hur, A., Elisseeff, A., & Guyon, I. (2002). A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing*, *2002*, 6–17.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Springer.

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, *10*(2-3), 191–203.

Boongoen, T., & Iam-On, N. (2018). Cluster ensembles: A survey of approaches with recent extensions and applications. *Computer Science Review*, *28*, 1–25.

Bubeck, S., Meila, M., & von Luxburg, U. (2012). How the initialization affects the stability of the k-means algorithm. *ESAIM: Probability and Statistics*, *16*, 436–452. https://doi.org/10.1051/ps/2012013

Calinski, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, *3*(1), 1–27.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2), 224–227.

Dudek, A. (2020). Silhouette index as clustering evaluation tool. In K. Jajuga, J. Batóg, & M. Walesiak (Eds.), *Classification and Data Analysis* (pp. 19–33). Springer International Publishing. https://doi.org/10.1007/978-3-030-52348-0_2

Fred, A. L. N., & Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(6), 835–850. https://doi.org/10.1109/TPAMI.2005.113

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In M. F. Balcan & K. Q. Weinberger (Eds.), *Proceedings of the 33rd International Conference on Machine Learning* (pp. 1050–1059, Vol. 48). PMLR. https://proceedings.mlr.press/v48/gal16.html

Golalipour, K., Akbari, E., Hamidi, S. S., Lee, M., & Enayatifar, R. (2021). From clustering to clustering ensemble selection: A review. *Engineering Applications of Artificial Intelligence*, *104*, 104388. https://doi.org/10.1016/j.engappai.2021.104388

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning* (pp. 1321–1330, Vol. 70). PMLR. https://proceedings.mlr.press/v70/guo17a.html

Hennig, C. (2007). Cluster-wise assessment of cluster stability. *Computational Statistics & Data Analysis*, *52*(1), 258–271. https://doi.org/10.1016/j.csda.2006.11.025

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, *58*(301), 13–30.

Inkaya, T. (2023). Consensus similarity graph construction for clustering. *Pattern Analysis and Applications*, *26*, 703–733. https://doi.org/10.1007/s10044-022-01116-w

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys (CSUR)*, *31*(3), 264–323.

Jia, Y., Cheng, J., Liu, H., & Hou, J. (2025). Towards calibrated deep clustering network [preprint arXiv:2403.02998]. https://arxiv.org/abs/2403.02998

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: An introduction to cluster analysis*. Wiley-Interscience.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*(1-2), 83–97.

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, *30*, 6405–6416. https://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles

Lange, T., Roth, V., Braun, M., & Buhmann, J. (2004). Stability-based validation of clustering solutions. *Neural Computation*, *16*, 1299–1323.

Lee, A. J. (1990). *U-statistics: Theory and practice*. Marcel Dekker.

Liu, T., Yu, H., & Blair, R. H. (2022). Stability estimation for unsupervised clustering: A review. *WIREs Computational Statistics*, *14*(6), e1575. https://doi.org/10.1002/wics.1575

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. Le Cam & J. Neyman (Eds.), *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability, volume 1: Statistics* (pp. 281–297). University of California Press.

Meila, M. (2007). Comparing clusterings: An information based distance. *Journal of Multivariate Analysis*, *98*(5), 873–895. https://doi.org/10.1016/j.jmva.2006.11.013

Pavlopoulos, J., Vardakas, G., & Likas, A. (2025). Revisiting Silhouette aggregation. In D. Pedreschi, A. Monreale, R. Guidotti, R. Pellungrini, & F. Naretto (Eds.), *Discovery Science* (pp. 354–368). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-78977-9_23

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

Powers, D. M. W. (2011). Evaluation: From precision, recall and f-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, *2*, 37–63.

Reynolds, D. (2009). Gaussian mixture models. In *Encyclopedia of Biometrics* (pp. 659–663). Springer US. https://doi.org/10.1007/978-0-387-73003-5_196

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

Shafer, G., & Vovk, V. (2007). A tutorial on conformal prediction. *Journal of Machine Learning Research*, *9*, 371–421.

Smiti, A. (2020). A critical overview of outlier detection methods. *Computer Science Review*, *38*, 100306. https://doi.org/10.1016/j.cosrev.2020.100306

Strehl, A., & Ghosh, J. (2002). Cluster ensembles: A knowledge reuse framework for combining partitionings. *Journal of Machine Learning Research*, *3*, 583–617.

Topchy, A., Jain, A. K., & Punch, W. (2005). Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(12), 1866–1881. https://doi.org/10.1109/TPAMI.2005.237

Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2014). OpenML: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, *15*(2), 49–60.

Vendramin, L., Campello, R. J. G. B., & Hruschka, E. R. (2009). On the comparison of relative clustering validity criteria. *Proceedings of the 2009 SIAM International Conference on Data Mining (SDM)*, 733–744. https://doi.org/10.1137/1.9781611972795.63

Wang, F., Franco-Penya, H.-H., Kelleher, J. D., Pugh, J., & Ross, R. (2017). An analysis of the application of simplified silhouette to the evaluation of k-means clustering validity. In P. Perner (Ed.), *Machine learning and data mining in pattern recognition – 13th international conference, MLDM 2017, proceedings* (pp. 291–305, Vol. 10358 LNAI). Springer Verlag. https://doi.org/10.1007/978-3-319-62416-7_21

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*(3), 645–678.

Zhang, M. (2022). Weighted clustering ensemble: A review. *Pattern Recognition*, *124*, 108428. https://doi.org/10.1016/j.patcog.2021.108428

Zhang, X., Jia, Y., Song, M., & Wang, R. (2025). Similarity and dissimilarity guided co-association matrix construction for ensemble clustering. *IEEE Transactions on Knowledge and Data Engineering*, *37*(11), 6694–6707.

A. Appendix Theoretical Analysis

A.1. Mis-ranking bound

Fix two points $x_i, x_j \in \mathcal{X}$ such that $\theta_i \geq \theta_j + \gamma$ for some margin $\gamma > 0$. Recall that the empirical score c_i (Eq. 7) is an order-2 U-statistic whose (Bernoulli) kernel $A_i^{(r_1,r_2)} \in [0,1]$ (Eq. 6) has mean $\mathbb{E}\left[A_i^{(r_1,r_2)}\right] = \theta_i$ (§4.1). For each point, the estimation errors are defined as: $e_i = c_i - \theta_i, \;\; e_j = c_j - \theta_j$. Using Hoeffding's inequality for order-2 U-statistics with a kernel in $[0,1]$ for e_i, e_j:

$$\Pr\left(e_i \geq \epsilon\right) \leq \exp\left(-\frac{R\epsilon^2}{2}\right), \quad \Pr\left(e_i \leq -\epsilon\right) \leq \exp\left(-\frac{R\epsilon^2}{2}\right) \quad \forall \epsilon > 0 \text{ (same for } e_j\text{)}.$$

Choosing $\epsilon = \frac{\gamma}{2}$, the events are defined as: $E_1 = \left\{e_i \leq -\frac{\gamma}{2}\right\}, \; E_2 = \left\{e_j \geq \frac{\gamma}{2}\right\}$, with probabilities:

$$\Pr\left(E_1\right), \Pr\left(E_2\right), \text{ which are both bounded above by } \exp\left\{-R\left(\frac{(\gamma/2)^2}{2}\right)\right\} = \exp\left\{-R\frac{\gamma^2}{8}\right\}.$$

Assume, for contradiction, that the event $\left\{c_i < c_j\right\}$ occurs while neither of E_1 nor E_2 takes place.
Then $e_i > -\gamma/2$ and $e_j < \gamma/2$, so that:

$$c_i = \theta_i + e_i \geq \theta_i - \frac{\gamma}{2} \geq \theta_j + \gamma - \frac{\gamma}{2} \geq \theta_j + \frac{\gamma}{2} \geq \theta_j + e_j = c_j \Rightarrow c_i \geq c_j, \text{ which contradicts } c_i < c_j.$$

Therefore, $\left\{c_i < c_j\right\} \subseteq E_1 \cup E_2$. Using the bounds on E_1 and E_2 obtained above:

$$\Pr\left[c_i < c_j\right] \leq \Pr\left[E_1\right] + \Pr\left[E_2\right] \leq 2\exp\left\{-R \cdot \frac{\gamma^2}{8}\right\}. \tag{12}$$

Hence, the probability of a mis-ranking satisfies: $\theta_i - \theta_j \geq \gamma \;\Rightarrow\; \Pr[c_i < c_j] \leq 2\,\exp\left\{-R\frac{\gamma^2}{8}\right\}$.

A.2. False-stability bound

Consider a uniform-noise observation $x_i \in \mathcal{X}$, then for each $L^{(r)} \in \mathcal{L}, \; r \in \{1, \dots, R\}$, its label $L_i^{(r)} \in L^{(r)}$ is drawn independently and uniformly from $\{1, \dots, k\}$. For such a point the Bernoulli kernel in Eq. 6 has expectation:

$$\mathbb{E}\left[A_i^{(r_1,r_2)}\right] = \theta = \frac{1}{k}.$$

Applying Hoeffding's inequality to c_i again gives: $\Pr\left[c_i - \mathbb{E}\left[A_i^{(r_1,r_2)}\right] \geq \epsilon\right] \leq \exp\left\{-R\frac{\epsilon^2}{2}\right\}$. Fix a stability threshold $\tau \in \left(\frac{1}{k}, 1\right)$ and select $\epsilon = (\tau - \theta)$. Then:

$$\Pr\left[c_i > \tau\right] = \Pr\left[c_i - \theta > \tau - \theta\right] \leq \exp\left\{-R\frac{(\tau-\theta)^2}{2}\right\}.$$

Since x_i is a uniform-noise point: $\theta = 1/k$, and thus:

$$\Pr\left[c_i > \tau \mid x_i \text{ uniform-noise}\right] \leq \exp\left\{-\frac{R}{2}\left(\tau - \frac{1}{k}\right)^2\right\}. \tag{13}$$

Additionally, if a fraction $\phi \in (0,1)$ of the n points of the dataset are uniform-noise, linearity of expectation turns the single-point bound into a dataset bound immediately:

$$\mathbb{E}\left[\#\left\{\, x_i \text{ uniform-noise}, c_i > \tau\right\}\right] \leq n \cdot \phi \cdot \exp\left\{-\frac{R}{2}\left(\tau - \frac{1}{k}\right)^2\right\}. \tag{14}$$

In more general scenarios, a noisy observation may be stochastically biased, due to proximity to dominant clusters or algorithmic bias, resulting in a non-uniform label distribution across runs. In such cases, the same concentration bound applies with this generalized $\theta \geq 1/k$:

$$\Pr\left[c_i > \tau\right] \leq \exp\left\{-R\frac{(\tau-\theta)^2}{2}\right\} \tag{15}$$

and extends the false-positive bound beyond uniform noise to any setting where labelings follow a known or estimated marginal distribution.

Geometric component. An analogous mis-ranking guarantee holds for the Silhouette-based geometric component. For each point x_i and run $r \in \{1, \dots, R\}$, let $s_i^{(r)} \in [-1, 1]$ denote its (possibly approximated) Silhouette score in run r. The randomness here comes from the clustering procedure (e.g., different random initializations), so for fixed i, $s_i^{(1)}, \dots, s_i^{(R)}$ are viewed as i.i.d. bounded random variables. Let

$$\bar{s}_i := \frac{1}{R}\sum_{r=1}^{R} s_i^{(r)}, \qquad \hat{\sigma}_i^2 := \frac{1}{R}\sum_{r=1}^{R}(s_i^{(r)} - \bar{s}_i)^2,$$

and recall that the ensemble geometric component used in CAKE is

$$\tilde{S}_i := \max\big(0,\ \bar{s}_i - \hat{\sigma}_i\big)$$

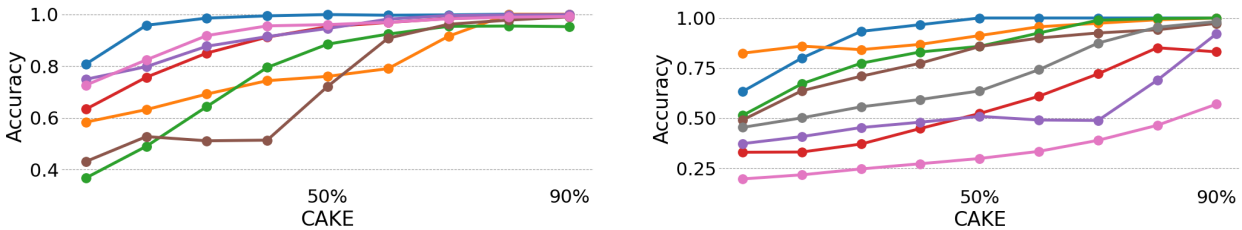
(up to the affine rescaling into $[0, 1]$ described in §3.2). The map

$$(s_i^{(1)}, \dots, s_i^{(R)}) \mapsto \tilde{S}_i$$

is a bounded-differences (Lipschitz) functional of the $s_i^{(r)}$'s, so by standard concentration inequalities for bounded independent variables (e.g., McDiarmid's inequality) $\tilde{S}_i$ concentrates around its population counterpart $g_i := \mathbb{E}[\tilde{S}_i]$ at the usual $1/\sqrt{R}$ rate, with sub-Gaussian tails. Repeating the mis-ranking argument of Appendix A.1, but with c_i and θ_i replaced by $\tilde{S}_i$ and g_i, shows that whenever the true geometric scores of two points differ by a margin $\gamma > 0$ (i.e., $g_i \geq g_j + \gamma$), the probability that CAKE mis-ranks them (i.e., $\tilde{S}_i < \tilde{S}_j$) decays exponentially in $R\gamma^2$. A false-positive bound for $\tilde{S}_i$ can also be derived under additional assumptions on the distribution of Silhouette scores for noise points, again yielding exponentially small probabilities in R.

B. Appendix Results

Clustering accuracy vs. CAKE percentiles (decile analysis). To complement the coverage–accuracy curves in Fig. 7, clustering accuracy is also examined across CAKE score quantiles. For each dataset, k-means is applied at the ground-truth k, the predicted labels are aligned to ground truth via Hungarian matching, points are partitioned into CAKE deciles, and accuracy is computed within each decile. This yields an accuracy–percentile trend that summarizes how accuracy changes from the lowest-scored (most ambiguous) instances to the highest-scored (most reliable) instances, providing a decile-level view of CAKE's ranking behavior over the full dataset (Figs. 13a,13b).



(a) Synthetic datasets: S1, S2, S3, S4, S5, S6, S7.

(b) Real datasets: Ir, Bc, Dg, Ng, Fm, Pd, Lt, Sa.

Figure 13: **Accuracy vs. CAKE score percentile.** Clustering accuracy within CAKE deciles (higher percentiles correspond to higher CAKE scores), after Hungarian alignment to ground-truth labels. This complements Fig. 7 by showing accuracy trends across score quantiles.

Dataset	Subset	avg Silhouette	avg NMI	CORR (CAKE ∼ ACC)
	Full	0.539 [0.5390, 0.5391]	0.834 [0.8332, 0.8339]	
	Random	0.546 [0.5460, 0.5460]	0.834 [0.8340, 0.8340]	
	Consensus	0.539 [0.5389, 0.5389]	0.839 [0.8395, 0.8395]	
S1	C component	0.539 [0.5389, 0.5389]	0.839 [0.8395, 0.8395]	0.881
	$\tilde{S}$ component	**0.692** [0.6921, 0.6921]	**0.983** [0.9828, 0.9828]	
	CAKE$^{(PR)}$	**0.692** [0.6921, 0.6921]	**0.983** [0.9828, 0.9828]	
	CAKE$^{(HM)}$	**0.692** [0.6921, 0.6921]	**0.983** [0.9828, 0.9828]	
	Full	0.462 [0.4499, 0.4736]	0.606 [0.6059, 0.6070]	
	Random	0.466 [0.4487, 0.4828]	0.602 [0.5927, 0.6105]	
	Consensus	0.459 [0.4499, 0.4673]	0.628 [0.6034, 0.6533]	
S2	C component	0.459 [0.4499, 0.4673]	0.628 [0.6034, 0.6533]	0.975
	$\tilde{S}$ component	**0.582** [0.5254, 0.6394]	0.698 [0.6792, 0.7165]	
	CAKE$^{(PR)}$	**0.582** [0.5253, 0.6384]	**0.700** [0.6803, 0.7195]	
	CAKE$^{(HM)}$	**0.582** [0.5252, 0.6392]	0.699 [0.6799, 0.7185]	
	Full	0.537 [0.5270, 0.5472]	0.614 [0.6101, 0.6181]	
	Random	0.503 [0.4694, 0.5357]	0.581 [0.5357, 0.6259]	
	Consensus	0.613 [0.5678, 0.6578]	0.757 [0.7304, 0.7838]	
S3	C component	0.599 [0.5657, 0.6327]	0.721 [0.7044, 0.7382]	0.975
	$\tilde{S}$ component	**0.639** [0.5904, 0.6867]	**0.781** [0.7593, 0.8024]	
	CAKE$^{(PR)}$	0.589 [0.5355, 0.6416]	0.751 [0.6986, 0.8033]	
	CAKE$^{(HM)}$	0.613 [0.5858, 0.6407]	0.777 [0.7600, 0.7934]	
	Full	0.434 [0.4339, 0.4339]	0.678 [0.6778, 0.6790]	
	Random	0.441 [0.4410, 0.4410]	0.693 [0.6931, 0.6938]	
	Consensus	0.391 [0.3845, 0.3981]	0.599 [0.5946, 0.6036]	
S4	C component	0.391 [0.3845, 0.3981]	0.599 [0.5946, 0.6036]	0.999
	$\tilde{S}$ component	**0.602** [0.6021, 0.6021]	**0.867** [0.8669, 0.8669]	
	CAKE$^{(PR)}$	**0.602** [0.6021, 0.6021]	**0.867** [0.8669, 0.8669]	
	CAKE$^{(HM)}$	**0.602** [0.6021, 0.6021]	**0.867** [0.8669, 0.8669]	
	Full	0.537 [0.4467, 0.6267]	0.660 [0.5741, 0.7463]	
	Random	0.512 [0.4115, 0.6127]	0.634 [0.5419, 0.7268]	
	Consensus	0.495 [0.4943, 0.4951]	0.570 [0.5693, 0.5714]	
S5	C component	0.582 [0.5298, 0.6349]	0.697 [0.6401, 0.7542]	0.996
	$\tilde{S}$ component	0.724 [0.6001, 0.8486]	0.855 [0.7559, 0.9536]	
	CAKE$^{(PR)}$	**0.805** [0.8049, 0.8049]	0.879 [0.8793, 0.8793]	
	CAKE$^{(HM)}$	**0.805** [0.8051, 0.8051]	**0.898** [0.8980, 0.8980]	
	Full	0.511 [0.5106, 0.5116]	0.474 [0.4722, 0.4750]	
	Random	0.512 [0.5075, 0.5157]	0.481 [0.4693, 0.4924]	
	Consensus	0.669 [0.6571, 0.6807]	0.586 [0.5783, 0.5946]	
S6	C component	0.675 [0.6745, 0.6748]	0.590 [0.5898, 0.5910]	0.963
	$\tilde{S}$ component	0.691 [0.6545, 0.7281]	0.591 [0.5773, 0.6039]	
	CAKE$^{(PR)}$	0.697 [0.6491, 0.7454]	0.584 [0.5576, 0.6098]	
	CAKE$^{(HM)}$	**0.717** [0.6936, 0.7396]	**0.597** [0.5896, 0.6037]	
	Full	0.467 [0.4112, 0.5232]	0.649 [0.5645, 0.7334]	
	Random	0.466 [0.4091, 0.5222]	0.636 [0.5467, 0.7255]	
	Consensus	0.420 [0.4160, 0.4246]	0.546 [0.5389, 0.5523]	
S7	C component	0.419 [0.4156, 0.4231]	0.541 [0.5330, 0.5484]	0.984
	$\tilde{S}$ component	0.581 [0.5068, 0.6557]	0.762 [0.6531, 0.8714]	
	CAKE$^{(PR)}$	**0.643** [0.5941, 0.6913]	**0.861** [0.7910, 0.9309]	
	CAKE$^{(HM)}$	0.562 [0.4828, 0.6410]	0.741 [0.6242, 0.8580]	
	Full	0.548 [0.5409, 0.5559]	0.731 [0.6947, 0.7679]	
	Random	0.527 [0.5200, 0.5335]	0.703 [0.6592, 0.7470]	
	Consensus	0.537 [0.5191, 0.5550]	0.739 [0.7106, 0.7664]	
Ir	C component	0.537 [0.5191, 0.5550]	0.739 [0.7106, 0.7664]	0.884
	$\tilde{S}$ component	0.637 [0.5766, 0.6971]	0.873 [0.7716, 0.9745]	
	CAKE$^{(PR)}$	**0.655** [0.6133, 0.6964]	0.868 [0.8154, 0.9204]	
	CAKE$^{(HM)}$	0.648 [0.5942, 0.7015]	**0.902** [0.8134, 0.9909]	
	Full	0.357 [0.3560, 0.3574]	0.544 [0.5349, 0.5524]	
	Random	0.360 [0.3599, 0.3599]	0.535 [0.5346, 0.5346]	
	Consensus	0.390 [0.3897, 0.3901]	**0.680** [0.6779, 0.6822]	
Bc				0.963

	C component	0.390 [0.3897, 0.3901]	**0.680** [0.6779, 0.6822]	
	$\tilde{S}$ component	**0.557** [0.5568, 0.5568]	0.653 [0.6526, 0.6526]	
	CAKE$^{(PR)}$	**0.557** [0.5568, 0.5568]	0.653 [0.6526, 0.6526]	
	CAKE$^{(HM)}$	**0.557** [0.5568, 0.5568]	0.653 [0.6526, 0.6526]	
	Full	0.178 [0.1711, 0.1857]	0.734 [0.7233, 0.7447]	
	Random	0.181 [0.1737, 0.1892]	0.717 [0.7053, 0.7280]	
	Consensus	0.213 [0.1953, 0.2311]	0.838 [0.8123, 0.8635]	
DG	C component	0.208 [0.1904, 0.2254]	0.837 [0.8140, 0.8591]	0.996
	$\tilde{S}$ component	0.236 [0.2156, 0.2557]	0.854 [0.8341, 0.8744]	
	CAKE$^{(PR)}$	0.230 [0.2110, 0.2482]	0.844 [0.8155, 0.8731]	
	CAKE$^{(HM)}$	**0.237** [0.2181, 0.2558]	**0.868** [0.8475, 0.8891]	
	Full	0.076 [0.0699, 0.0812]	0.511 [0.5016, 0.5194]	
	Random	0.082 [0.0800, 0.0834]	0.518 [0.5103, 0.5254]	
	Consensus	0.105 [0.1021, 0.1072]	0.588 [0.5838, 0.5914]	
NG	C component	0.106 [0.1028, 0.1094]	0.592 [0.5850, 0.5996]	0.960
	$\tilde{S}$ component	**0.108** [0.1047, 0.1118]	0.611 [0.6017, 0.6201]	
	CAKE$^{(PR)}$	**0.108** [0.0992, 0.1167]	**0.619** [0.6136, 0.6252]	
	CAKE$^{(HM)}$	**0.108** [0.1047, 0.1118]	0.611 [0.6017, 0.6201]	
	Full	0.142 [0.1373, 0.1476]	0.507 [0.4967, 0.5164]	
	Random	0.140 [0.1341, 0.1458]	0.510 [0.4996, 0.5195]	
	Consensus	0.173 [0.1662, 0.1796]	0.560 [0.5535, 0.5668]	
FM	C component	0.173 [0.1677, 0.1786]	0.560 [0.5519, 0.5688]	0.890
	$\tilde{S}$ component	0.184 [0.1716, 0.1961]	0.585 [0.5741, 0.5963]	
	CAKE$^{(PR)}$	0.185 [0.1732, 0.1967]	0.588 [0.5750, 0.6014]	
	CAKE$^{(HM)}$	**0.192** [0.1812, 0.2023]	**0.595** [0.5793, 0.6113]	
	Full	0.303 [0.2904, 0.3161]	0.676 [0.6673, 0.6851]	
	Random	0.291 [0.2822, 0.3005]	0.683 [0.6733, 0.6925]	
	Consensus	0.355 [0.3425, 0.3684]	0.734 [0.7217, 0.7473]	
PD	C component	0.358 [0.3427, 0.3728]	0.722 [0.7085, 0.7351]	0.999
	$\tilde{S}$ component	**0.395** [0.3818, 0.4087]	**0.842** [0.8141, 0.8698]	
	CAKE$^{(PR)}$	0.379 [0.3548, 0.4028]	0.820 [0.7985, 0.8410]	
	CAKE$^{(HM)}$	0.394 [0.3777, 0.4095]	**0.842** [0.8246, 0.8601]	
	Full	0.141 [0.1383, 0.1433]	0.371 [0.3665, 0.3757]	
	Random	0.142 [0.1401, 0.1444]	0.378 [0.3719, 0.3838]	
	Consensus	0.185 [0.1816, 0.1880]	0.434 [0.4304, 0.4373]	
LT	C component	0.190 [0.1864, 0.1938]	0.444 [0.4387, 0.4500]	0.987
	$\tilde{S}$ component	**0.193** [0.1892, 0.1976]	0.462 [0.4531, 0.4703]	
	CAKE$^{(PR)}$	0.190 [0.1849, 0.1956]	0.461 [0.4533, 0.4697]	
	CAKE$^{(HM)}$	0.192 [0.1880, 0.1954]	**0.464** [0.4593, 0.4681]	
	Full	0.349 [0.3378, 0.3605]	0.596 [0.5561, 0.6350]	
	Random	0.348 [0.3350, 0.3602]	0.609 [0.5901, 0.6279]	
	Consensus	0.336 [0.3107, 0.3617]	0.575 [0.5556, 0.5952]	
SA	C component	0.336 [0.3107, 0.3617]	0.575 [0.5556, 0.5952]	0.999
	$\tilde{S}$ component	**0.483** [0.4688, 0.4972]	**0.765** [0.7588, 0.7707]	
	CAKE$^{(PR)}$	0.468 [0.4393, 0.4959]	0.749 [0.7272, 0.7708]	
	CAKE$^{(HM)}$	0.443 [0.4104, 0.4765]	0.735 [0.7092, 0.7598]	

Table 7: Average Silhouette and NMI (Normalized Mutual Information) with t-95% confidence intervals and Spearman's rank correlation of CAKE percentiles with clustering accuracy percentiles. Results are reported for the full datasets (Full) and their filtered subsets mentioned in §5.2. Highest values in **bold**.

Dataset	Metric	CAKE$^{(PR)}$	CAKE$^{(HM)}$	Entropy agreement	Bootstrap stability
S1	AUPRC	0.992	**0.993**	0.976	0.967
	AUROC	0.884	**0.894**	0.733	0.619
S2	AUPRC	**0.881**	**0.881**	0.801	0.872
	AUROC	**0.695**	0.692	0.598	0.688
S3	AUPRC	**0.943**	0.940	0.915	0.868
	AUROC	**0.896**	0.890	0.879	0.820
S4	AUPRC	0.973	**0.974**	0.948	0.929
	AUROC	0.835	**0.836**	0.773	0.697
S5	AUPRC	0.982	**0.983**	0.917	0.909
	AUROC	0.855	**0.856**	0.546	0.530
S6	AUPRC	**0.919**	**0.919**	0.812	0.766
	AUROC	**0.809**	0.808	0.720	0.648
S7	AUPRC	0.975	**0.977**	0.950	0.944
	AUROC	**0.836**	0.834	0.735	0.650
Ir	AUPRC	0.991	**0.992**	0.962	0.980
	AUROC	0.940	**0.941**	0.843	0.907
Bc	AUPRC	0.961	**0.963**	0.920	0.936
	AUROC	0.733	**0.743**	0.584	0.670
Dg	AUPRC	**0.934**	0.931	0.917	0.916
	AUROC	**0.815**	0.809	0.785	0.771
Ng	AUPRC	0.661	0.658	**0.682**	0.660
	AUROC	**0.695**	0.693	0.677	0.656
Fm	AUPRC	0.743	**0.745**	0.652	0.713
	AUROC	0.676	**0.679**	0.643	0.648
Pd	AUPRC	**0.929**	**0.929**	0.903	0.853
	AUROC	**0.844**	0.841	0.823	0.769
Lt	AUPRC	**0.428**	0.423	0.415	0.350
	AUROC	**0.659**	0.656	0.652	0.606
Sa	AUPRC	0.876	**0.879**	0.825	0.870
	AUROC	0.779	0.783	0.716	**0.792**

Table 8: Per-point consensus correctness prediction (equivalent to Table 3, which uses k-means), using **MiniBatchK-Means**. CAKE scores are computed from an R=20 MiniBatchKMeans ensemble; entropy-based agreement (R=20) and bootstrap-based stability (B=20) are both obtained using MiniBatchKMeans runs. Highest per row in **bold**.

Dataset	Metric	CAKE$^{(PR)}$	CAKE$^{(HM)}$	Entropy agreement	Bootstrap stability
S1	AUPRC	**0.997**	**0.997**	0.964	0.965
	AUROC	**0.933**	**0.933**	0.576	0.589
S2	AUPRC	**0.928**	**0.928**	0.888	0.794
	AUROC	0.752	**0.753**	0.720	0.544
S3	AUPRC	**0.936**	0.935	0.848	0.817
	AUROC	**0.853**	**0.853**	0.731	0.673
S4	AUPRC	0.970	**0.971**	0.906	0.928
	AUROC	0.834	**0.837**	0.613	0.676
S5	AUPRC	**0.979**	**0.979**	0.917	0.908
	AUROC	0.797	**0.799**	0.504	0.533
S6	AUPRC	**0.912**	**0.912**	0.799	0.794
	AUROC	**0.686**	0.685	0.546	0.532
S7	AUPRC	0.975	**0.977**	0.939	0.908
	AUROC	0.793	**0.805**	0.575	0.458
NG	AUPRC	0.316	0.316	**0.384**	0.335
	AUROC	0.514	0.514	**0.591**	0.556
BC	AUPRC	**0.969**	0.967	0.937	0.930
	AUROC	**0.726**	0.718	0.626	0.601
DG	AUPRC	0.925	0.920	**0.934**	0.885
	AUROC	0.807	0.798	**0.832**	0.741
FM	AUPRC	**0.770**	0.765	0.755	0.693
	AUROC	0.737	0.733	**0.764**	0.702
IR	AUPRC	0.985	**0.986**	0.930	0.900
	AUROC	0.895	**0.899**	0.570	0.466
LT	AUPRC	0.368	0.372	**0.373**	0.340
	AUROC	0.604	0.605	**0.629**	0.575
PD	AUPRC	0.942	**0.945**	0.845	0.897
	AUROC	0.862	**0.866**	0.725	0.764
SA	AUPRC	**0.915**	0.913	0.882	0.842
	AUROC	**0.815**	0.811	0.760	0.680

Table 9: Per-point consensus correctness prediction (equivalent to Table 3, which uses k-means), using **K-Medoids**. CAKE scores are computed from an R=20 K-Medoids ensemble; entropy-based agreement (R=20) and bootstrap-based stability (B=20) are both obtained using K-Medoids runs. Highest per row in **bold**.

Dataset	Metric	CAKE$^{(PR)}$	CAKE$^{(HM)}$	Entropy agreement	Bootstrap stability
S1	AUPRC	**0.997**	**0.997**	0.958	0.960
	AUROC	**0.932**	**0.932**	0.503	0.525
S2	AUPRC	**0.932**	**0.932**	0.791	0.794
	AUROC	**0.766**	**0.766**	0.507	0.515
S3	AUPRC	0.853	0.845	**0.945**	0.555
	AUROC	0.756	0.747	**0.956**	0.446
S4	AUPRC	**0.976**	**0.976**	0.882	0.895
	AUROC	**0.852**	**0.852**	0.510	0.570
S5	AUPRC	**0.995**	**0.995**	0.992	0.992
	AUROC	**0.568**	**0.568**	0.500	0.531
S6	AUPRC	**0.976**	**0.976**	0.967	0.969
	AUROC	0.506	0.506	0.503	**0.529**
S7	AUPRC	**0.982**	**0.982**	0.967	0.953
	AUROC	**0.675**	**0.675**	0.542	0.437
IR	AUPRC	**0.998**	**0.998**	0.967	0.968
	AUROC	**0.949**	**0.949**	0.500	0.537
BC	AUPRC	**0.909**	**0.909**	0.692	0.727
	AUROC	**0.800**	**0.800**	0.499	0.567
DG	AUPRC	**0.947**	0.945	0.943	0.926
	AUROC	0.797	0.791	**0.799**	0.734
NG	AUPRC	**0.691**	0.684	0.685	0.651
	AUROC	**0.711**	0.704	0.676	0.634
FM	AUPRC	0.684	**0.688**	0.572	0.613
	AUROC	0.679	**0.683**	0.596	0.608
PD	AUPRC	0.898	**0.899**	0.850	0.891
	AUROC	**0.833**	0.832	**0.833**	**0.833**
LT	AUPRC	0.457	0.452	**0.490**	0.399
	AUROC	0.628	0.627	**0.673**	0.586
SA	AUPRC	**0.912**	**0.912**	0.596	0.584
	AUROC	**0.879**	**0.879**	0.507	0.483

Table 10: Per-point consensus correctness prediction (equivalent to Table 3, which uses k-means), using **GMM**. CAKE scores are computed from an R=20 GMM ensemble; entropy-based agreement (R=20) and bootstrap-based stability (B=20) are both obtained using GMM runs. The highest value for each dataset and metric is highlighted in **bold**.

Table 11: Average Silhouette, NMI, ARI, AMI, and ACC over multiple independent MiniBatchKMeans runs (different random seeds). Results are reported for the full dataset and for subsets selected by the different filtering strategies (random, consensus-agree, C component, $\tilde{S}$ component, CAKE$^{(PR)}$, CAKE$^{(HM)}$). Best performance per dataset and metric is shown in **bold**.

		Clustering results on filtered datasets				
Data	Subset	avg Silhouette	avg NMI	avg ARI	avg AMI	avg ACC
S1	Full	0.486	0.742	0.732	0.742	0.848
	Random	0.546	0.833	0.878	0.833	0.958
	Consensus	0.581	0.858	0.889	0.858	0.937
	C component	0.590	0.864	0.910	0.864	0.959
	$\tilde{S}$ component	0.632	0.928	0.928	0.928	0.952
	CAKE$^{(PR)}$	0.636	0.950	0.973	0.950	0.990
	CAKE$^{(HM)}$	**0.646**	**0.955**	**0.975**	**0.955**	**0.991**
S2	Full	0.470	0.600	0.499	0.600	0.666
	Random	0.448	0.593	0.537	0.592	0.753
	Consensus	0.553	0.652	0.657	0.652	0.718
	C component	0.509	0.638	0.604	0.638	0.715
	$\tilde{S}$ component	0.562	0.677	0.608	0.676	0.723
	CAKE$^{(PR)}$	**0.580**	**0.687**	**0.667**	**0.686**	**0.779**
	CAKE$^{(HM)}$	0.578	0.681	0.656	0.681	0.775
S3	Full	0.501	0.588	0.464	0.587	0.700
	Random	0.510	0.604	0.490	0.604	0.719
	Consensus	0.581	0.757	0.763	0.757	0.821
	C component	0.593	0.770	**0.782**	0.769	**0.836**
	$\tilde{S}$ component	0.601	0.742	0.716	0.742	0.799
	CAKE$^{(PR)}$	**0.628**	**0.771**	0.774	**0.771**	0.828
	CAKE$^{(HM)}$	0.587	0.743	0.729	0.742	0.802
S4	Full	0.420	0.644	0.664	0.644	0.856
	Random	0.435	0.674	0.702	0.674	0.876
	Consensus	0.457	0.728	0.699	0.727	0.789
	C component	**0.535**	**0.796**	**0.812**	**0.795**	**0.900**
	$\tilde{S}$ component	0.516	0.785	0.779	0.785	0.859
	CAKE$^{(PR)}$	0.511	0.777	0.775	0.776	0.854
	CAKE$^{(HM)}$	0.496	0.765	0.749	0.765	0.840
S5	Full	0.504	0.615	0.532	0.615	0.746
	Random	0.536	0.651	0.579	0.651	0.774
	Consensus	0.614	0.689	0.655	0.689	0.836
	C component	0.620	0.710	0.681	0.710	0.846
	$\tilde{S}$ component	0.745	0.856	0.839	0.855	0.892
	CAKE$^{(PR)}$	**0.815**	**0.923**	**0.948**	**0.923**	**0.981**
	CAKE$^{(HM)}$	0.779	0.892	0.896	0.892	0.938
S6	Full	0.508	0.481	0.316	0.481	0.687
	Random	0.511	0.486	0.325	0.485	0.690
	Consensus	0.647	0.585	0.566	0.584	0.781
	C component	0.654	**0.590**	**0.570**	**0.590**	**0.786**
	$\tilde{S}$ component	**0.663**	0.567	0.541	0.567	0.757
	CAKE$^{(PR)}$	0.658	0.580	0.555	0.580	0.771
	CAKE$^{(HM)}$	0.640	0.569	0.546	0.568	0.766
S7	Full	0.419	0.571	0.502	0.571	0.729
	Random	0.384	0.532	0.434	0.532	0.642
	Consensus	0.430	0.587	0.506	0.587	0.703
	C component	0.446	0.613	0.550	0.613	0.720
	$\tilde{S}$ component	0.577	0.787	0.734	0.787	0.810
	CAKE$^{(PR)}$	**0.579**	**0.801**	**0.744**	**0.801**	**0.820**
	CAKE$^{(HM)}$	0.553	0.773	0.699	0.773	0.778
IR	Full	0.533	0.730	0.682	0.727	0.825
	Random	0.526	0.715	0.658	0.710	0.844
	Consensus	0.628	0.846	0.829	0.843	0.886
	C component	0.643	**0.905**	**0.894**	**0.903**	**0.942**

		Clustering results on filtered datasets				
Data	Subset	avg Silhouette	avg NMI	avg ARI	avg AMI	avg ACC
	$\tilde{S}$ component	**0.664**	0.903	0.879	0.901	0.910
	CAKE$^{(PR)}$	**0.664**	0.903	0.879	0.901	0.910
	CAKE$^{(HM)}$	**0.664**	0.903	0.879	0.901	0.910
	Full	0.356	0.537	0.654	0.536	0.905
	Random	0.357	0.532	0.656	0.531	0.906
	Consensus	0.445	0.590	0.707	0.589	0.922
BC	C component	0.445	0.590	0.707	0.589	0.922
	$\tilde{S}$ component	0.496	0.556	0.654	0.555	0.916
	CAKE$^{(PR)}$	0.533	**0.608**	**0.715**	**0.607**	**0.937**
	CAKE$^{(HM)}$	**0.534**	0.605	0.707	0.604	0.936
	Full	0.152	0.667	0.536	0.663	0.668
	Random	0.169	0.692	0.566	0.688	0.691
	Consensus	0.199	0.800	0.716	0.797	0.757
DG	C component	0.203	0.819	0.748	0.816	0.781
	$\tilde{S}$ component	0.206	0.816	0.703	0.813	0.731
	CAKE$^{(PR)}$	**0.223**	**0.849**	**0.767**	**0.847**	**0.807**
	CAKE$^{(HM)}$	0.219	0.830	0.728	0.828	0.763
	Full	0.068	0.490	0.327	0.488	0.478
	Random	0.069	0.480	0.310	0.477	0.454
	Consensus	0.108	0.613	0.488	0.611	0.575
NG	C component	0.113	0.614	0.480	0.612	0.565
	$\tilde{S}$ component	**0.113**	**0.650**	**0.518**	**0.648**	**0.601**
	CAKE$^{(PR)}$	**0.113**	**0.650**	**0.518**	**0.648**	**0.601**
	CAKE$^{(HM)}$	**0.113**	**0.650**	**0.518**	**0.648**	**0.601**
	Full	0.171	0.505	0.353	0.505	0.515
	Random	0.173	0.499	0.346	0.498	0.510
	Consensus	0.212	0.551	0.414	0.550	0.540
FM	C component	0.226	0.528	0.394	0.527	0.510
	$\tilde{S}$ component	0.226	0.573	0.427	0.573	0.561
	CAKE$^{(PR)}$	**0.227**	0.578	0.440	0.578	0.557
	CAKE$^{(HM)}$	0.226	**0.588**	**0.456**	**0.588**	**0.581**
	Full	0.258	0.633	0.488	0.632	0.635
	Random	0.267	0.641	0.501	0.641	0.650
	Consensus	0.360	0.741	0.630	0.740	0.704
PD	C component	0.363	0.762	0.666	0.761	0.733
	$\tilde{S}$ component	0.360	0.782	0.632	0.782	0.703
	CAKE$^{(PR)}$	**0.377**	**0.810**	**0.718**	**0.810**	**0.777**
	CAKE$^{(HM)}$	0.356	0.787	0.667	0.787	0.717
	Full	0.128	0.356	0.140	0.353	0.262
	Random	0.122	0.360	0.141	0.356	0.268
	Consensus	0.165	0.425	0.187	0.421	0.311
LT	C component	**0.172**	0.430	0.191	0.426	0.310
	$\tilde{S}$ component	0.170	0.440	0.202	0.436	0.317
	CAKE$^{(PR)}$	0.169	**0.445**	**0.209**	**0.442**	**0.331**
	CAKE$^{(HM)}$	0.168	0.439	0.196	0.435	0.313
	Full	0.326	0.571	0.482	0.570	0.641
	Random	0.313	0.561	0.456	0.560	0.640
	Consensus	0.363	0.641	0.559	0.640	0.692
SA	C component	0.366	0.647	0.586	0.647	0.718
	$\tilde{S}$ component	0.437	0.716	0.637	0.715	0.741
	CAKE$^{(PR)}$	**0.442**	**0.720**	**0.651**	**0.720**	0.742
	CAKE$^{(HM)}$	0.425	0.715	0.650	0.714	**0.749**

Table 12: Average Silhouette, NMI, ARI, AMI, and ACC over multiple independent GMM runs (different random seeds). Results are reported for the full dataset and for subsets selected by the different filtering strategies (random, consensus-agree, C component, $\tilde{S}$ component, CAKE$^{(PR)}$, CAKE$^{(HM)}$). Best performance per dataset and metric is shown in **bold**.

		Clustering results on filtered datasets				
Data	Subset	avg Silhouette	avg NMI	avg ARI	avg AMI	avg ACC
S1	Full	0.539	0.834	0.879	0.834	0.958
	Random	0.546	0.833	0.878	0.833	0.957
	Consensus	0.538	0.836	0.881	0.836	0.959
	C component	0.538	0.836	0.881	0.836	0.959
	$\tilde{S}$ component	**0.692**	**0.983**	**0.992**	**0.983**	**0.997**
	CAKE$^{(PR)}$	**0.692**	**0.983**	**0.992**	**0.983**	**0.997**
	CAKE$^{(HM)}$	**0.692**	**0.983**	**0.992**	**0.983**	**0.997**
S2	Full	0.457	0.612	0.566	0.612	0.789
	Random	0.461	0.615	0.562	0.614	0.782
	Consensus	0.489	0.701	0.694	0.700	0.770
	C component	0.489	0.701	0.694	0.700	0.770
	$\tilde{S}$ component	0.613	0.712	0.739	0.712	0.831
	CAKE$^{(PR)}$	**0.615**	**0.716**	**0.743**	**0.716**	**0.833**
	CAKE$^{(HM)}$	**0.615**	**0.716**	**0.743**	**0.716**	**0.833**
S3	Full	0.510	0.617	0.500	0.617	0.720
	Random	0.474	0.616	0.502	0.616	0.717
	Consensus	0.673	0.840	0.864	0.839	0.932
	C component	0.673	0.840	0.864	0.839	0.932
	$\tilde{S}$ component	**0.729**	0.871	0.906	0.871	**0.949**
	CAKE$^{(PR)}$	0.723	0.833	0.857	0.833	0.905
	CAKE$^{(HM)}$	**0.729**	**0.872**	**0.907**	**0.872**	**0.949**
S4	Full	0.434	0.675	0.709	0.675	0.880
	Random	0.441	0.691	0.724	0.691	0.887
	Consensus	0.379	0.619	0.649	0.618	0.806
	C component	0.379	0.619	0.649	0.618	0.806
	$\tilde{S}$ component	**0.602**	**0.866**	**0.901**	**0.866**	**0.962**
	CAKE$^{(PR)}$	**0.602**	**0.866**	**0.901**	**0.866**	**0.962**
	CAKE$^{(HM)}$	**0.602**	**0.866**	**0.901**	**0.866**	**0.962**
S5	Full	0.573	0.954	0.973	0.954	**0.992**
	Random	0.577	0.954	0.972	0.954	0.991
	Consensus	0.457	0.682	0.575	0.681	0.725
	C component	0.457	0.682	0.575	0.681	0.725
	$\tilde{S}$ component	**0.827**	**0.962**	**0.976**	**0.962**	0.991
	CAKE$^{(PR)}$	**0.827**	**0.962**	**0.976**	**0.962**	0.991
	CAKE$^{(HM)}$	**0.827**	**0.962**	**0.976**	**0.962**	0.991
S6	Full	0.344	0.852	0.894	0.852	0.966
	Random	0.349	0.855	**0.896**	0.855	**0.967**
	Consensus	0.424	0.628	0.546	0.628	0.679
	C component	0.424	0.628	0.546	0.628	0.679
	$\tilde{S}$ component	**0.725**	**0.863**	0.895	**0.863**	0.960
	CAKE$^{(PR)}$	**0.725**	**0.863**	0.895	**0.863**	0.960
	CAKE$^{(HM)}$	**0.725**	**0.863**	0.895	**0.863**	0.960
S7	Full	0.476	0.759	0.754	0.759	0.879
	Random	0.510	0.842	0.881	0.842	0.965
	Consensus	0.441	0.601	0.508	0.601	0.710
	C component	0.439	0.601	0.508	0.601	0.703
	$\tilde{S}$ component	0.614	0.832	0.792	0.832	0.862
	CAKE$^{(PR)}$	**0.693**	**0.921**	**0.934**	**0.921**	**0.980**
	CAKE$^{(HM)}$	0.615	0.826	0.786	0.826	0.858
IR	Full	0.553	0.789	0.750	0.786	0.903
	Random	0.516	0.756	0.722	0.752	0.894
	Consensus	0.452	0.622	0.544	0.613	0.855
	C component	0.452	0.622	0.544	0.613	0.855

		Clustering results on filtered datasets				
Data	Subset	avg Silhouette	avg NMI	avg ARI	avg AMI	avg ACC
	$\tilde{S}$ component	**0.720**	**0.961**	**0.979**	**0.960**	**0.990**
	CAKE$^{(PR)}$	**0.720**	**0.961**	**0.979**	**0.960**	**0.990**
	CAKE$^{(HM)}$	**0.720**	**0.961**	**0.979**	**0.960**	**0.990**
	Full	0.277	0.031	0.069	0.030	0.649
	Random	0.241	0.033	0.074	0.031	0.651
	Consensus	0.291	0.302	0.404	0.301	0.819
BC	C component	0.291	0.302	0.404	0.301	0.819
	$\tilde{S}$ component	**0.292**	**0.543**	**0.690**	**0.542**	**0.920**
	CAKE$^{(PR)}$	**0.292**	**0.543**	**0.690**	**0.542**	**0.920**
	CAKE$^{(HM)}$	**0.292**	**0.543**	**0.690**	**0.542**	**0.920**
	Full	0.128	0.619	0.498	0.615	0.658
	Random	0.132	0.612	0.464	0.604	0.645
	Consensus	0.196	0.808	0.758	0.804	**0.779**
DG	C component	0.170	0.794	0.731	0.789	0.735
	$\tilde{S}$ component	**0.222**	**0.834**	**0.775**	**0.831**	0.770
	CAKE$^{(PR)}$	**0.222**	**0.834**	**0.775**	**0.831**	0.770
	CAKE$^{(HM)}$	**0.222**	**0.834**	**0.775**	**0.831**	0.770
	Full	0.071	0.505	0.301	0.503	0.503
	Random	0.071	0.505	0.285	0.503	0.489
	Consensus	0.109	0.576	0.430	0.574	0.564
NG	C component	0.112	0.580	0.438	0.578	0.567
	$\tilde{S}$ component	**0.122**	**0.654**	**0.514**	**0.652**	**0.602**
	CAKE$^{(PR)}$	**0.122**	**0.654**	**0.514**	**0.652**	**0.602**
	CAKE$^{(HM)}$	**0.122**	**0.654**	**0.514**	**0.652**	**0.602**
	Full	0.124	0.521	0.332	0.521	0.510
	Random	0.119	0.516	0.325	0.516	0.492
	Consensus	0.222	0.602	0.442	0.602	0.590
FM	C component	0.230	0.600	0.464	0.600	0.608
	$\tilde{S}$ component	**0.258**	**0.632**	**0.467**	**0.631**	**0.611**
	CAKE$^{(PR)}$	**0.258**	**0.632**	**0.467**	**0.631**	**0.611**
	CAKE$^{(HM)}$	**0.258**	**0.632**	**0.467**	**0.631**	**0.611**
	Full	0.178	0.585	0.425	0.584	0.586
	Random	0.184	0.610	0.456	0.609	0.621
	Consensus	0.306	0.745	0.665	0.744	0.713
PD	C component	0.322	0.753	0.672	0.752	0.715
	$\tilde{S}$ component	**0.382**	**0.814**	**0.704**	**0.813**	**0.772**
	CAKE$^{(PR)}$	**0.382**	**0.814**	**0.704**	**0.813**	**0.772**
	CAKE$^{(HM)}$	**0.382**	**0.814**	**0.704**	**0.813**	**0.772**
	Full	0.036	0.341	0.112	0.338	0.241
	Random	0.039	0.356	0.121	0.341	0.252
	Consensus	0.251	0.570	0.357	0.560	0.426
LT	C component	0.261	0.600	0.382	0.590	0.450
	$\tilde{S}$ component	0.263	0.600	0.384	0.591	0.452
	CAKE$^{(PR)}$	**0.281**	**0.601**	**0.391**	**0.592**	**0.460**
	CAKE$^{(HM)}$	**0.281**	**0.601**	**0.391**	**0.592**	**0.460**
	Full	0.171	0.575	0.530	0.574	0.660
	Random	0.170	0.574	0.496	0.573	0.637
	Consensus	0.169	0.565	0.514	0.565	0.655
SA	C component	0.169	0.565	0.514	0.565	0.655
	$\tilde{S}$ component	**0.409**	**0.754**	**0.748**	**0.754**	**0.821**
	CAKE$^{(PR)}$	**0.409**	**0.754**	**0.748**	**0.754**	**0.821**
	CAKE$^{(HM)}$	**0.409**	**0.754**	**0.748**	**0.754**	**0.821**

C. Effect of Ensemble Diversity on CAKE Score Quality

The analysis in this section investigates how the construction of the clustering ensemble (§3.1) influences the quality of the confidence scores produced by CAKE (Eq. 8). The objective is to assess whether the proposed score (Eq. 8b) remains informative under different homogeneous and heterogeneous ensemble constructions (Fig. 10), and to clarify how changes in ensemble diversity relate to the quality of the induced confidence ranking. Rather than seeking a single universally optimal ensemble design, the goal is to characterize the extent to which the practical usefulness of CAKE depends on the way the ensemble is formed.

For each dataset, multiple clustering ensembles were generated using different clustering methods and parameter settings. The considered constructions include both homogeneous ensembles, in which all runs are produced by the same clustering method, and heterogeneous ensembles, in which runs from different algorithms are combined within the same ensemble. More specifically, the analysis includes centroid-based constructions derived from k-means, probabilistic constructions derived from Gaussian mixture models, and hierarchical constructions derived from BIRCH, together with heterogeneous combinations of these approaches.

Variation within each clustering "family" was introduced through changes in key model settings, such as initialization choices, covariance structure, and threshold parameters, depending on the method under consideration. This design allows the effect of moderate variation within each construction, as well as stronger variation introduced by combining distinct clustering methods, to be examined in a unified way. For every ensemble construction, the number of runs was kept fixed at $R = 20$ and the number of clusters was set to the ground-truth number of classes (Table 1). To reduce sensitivity to random initialization, the complete procedure was repeated across multiple random seeds and the reported values were averaged over these repetitions.

Ensemble diversity was quantified through the disagreement among the partitions generated by the ensemble members. Let $\mathcal{E} = \{z^{(1)}, z^{(2)}, \dots, z^{(R)}\}$ denote an ensemble of R clustering runs, where $z^{(r)}$ is the assignment vector produced by run r. For each pair of runs, the normalized mutual information (NMI) between the corresponding partitions was computed, and ensemble diversity was defined as:

$$D(\mathcal{E}) \;=\; 1 - \frac{2}{R(R-1)} \sum_{r<s} \mathrm{NMI}\big(z^{(r)}, z^{(s)}\big)\,. \tag{16}$$

Under this definition, smaller values indicate that the ensemble members are highly similar to each other, whereas larger values indicate stronger disagreement and therefore a more diverse clustering ensemble. Since NMI is invariant to label permutations, this measure provides a label-independent summary of ensemble variability.

For each ensemble construction, $\mathrm{CAKE}^{\mathrm{HM}}$ scores (Eq. 8b) were computed, yielding one confidence value per instance. The quality of the resulting confidence scores was then evaluated through a ranking-based coverage analysis. Instances were sorted from highest to lowest confidence, and for a given coverage level L, the evaluation was restricted to the top-ranked $L\%$ of the dataset. Clustering quality over this high-confidence portion was measured using the adjusted Rand index (ARI) between the corresponding consensus assignments (i.e., the final ensemble-level clustering labels obtained after aggregating the individual runs) and the ground-truth labels. This procedure produces an ARI–coverage curve as a function of the confidence coverage level. In the experiments, the coverage level was varied over the fixed grid $L \in \{0.40, 0.45, 0.50, \dots, 0.90\}$. A single summary quantity, referred to as *score quality*, was obtained as the normalized area under this ARI–coverage curve:

$$Q \;=\; \frac{1}{L_{\max} - L_{\min}} \int_{L_{\min}}^{L_{\max}} \mathrm{ARI}(L)\, dL. \tag{17}$$

Intuitively, this quantity measures how well the confidence score orders instances according to assignment reliability. A high value indicates that points receiving higher confidence tend to have more reliable cluster assignments, so that the corresponding ARI remains high over the upper part of the confidence ranking.

Illustratively, Fig. 14 reports, for the representative datasets, the mean CAKE score quality $\bar{Q}$, the mean ARI of the consensus clustering on the dataset, and the mean ensemble diversity $\bar{D}$ for each ensemble construction, all averaged across repetitions. This comparison makes it possible to examine how changes in ensemble diversity relate both to the overall clustering quality and to the usefulness of the confidence ranking induced by CAKE.

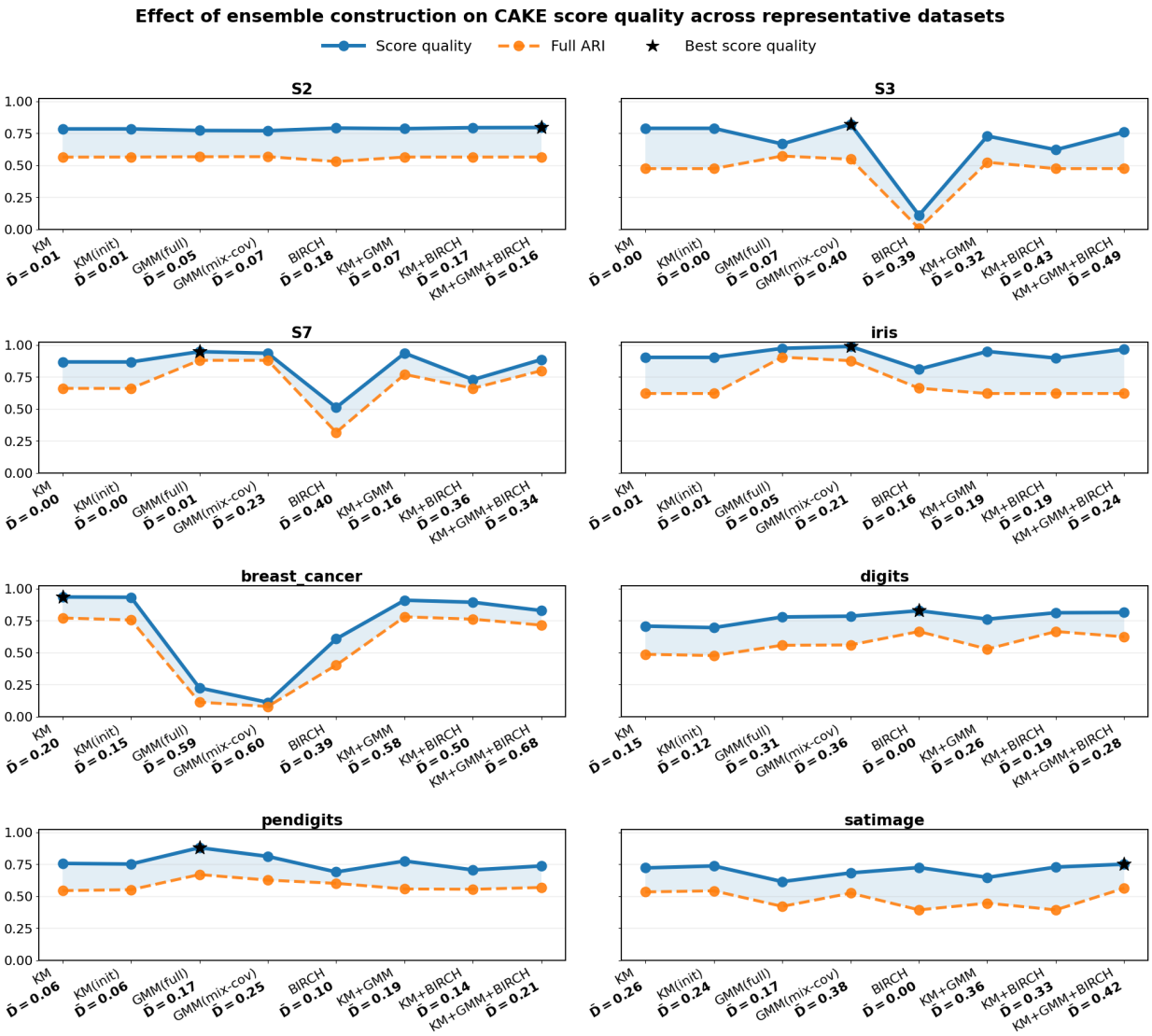


Figure 14: Effect of ensemble construction on CAKE score quality across representative datasets. Each panel corresponds to one dataset. The solid line reports the mean score quality $\bar{Q}$, while the dashed line reports the mean ARI of the corresponding consensus clustering; both quantities are averaged across run repetitions. The black star marks the ensemble construction attaining the highest $\bar{Q}$. The horizontal axis lists the ensemble constructions: KM: k-means; KM (init): k-means with varied initialization settings; GMM (full): GMMs with full covariance; GMM (mix-cov): GMMs with varying covariance structures; BIRCH: the corresponding hierarchical construction; and KM+GMM, KM+BIRCH, and KM+GMM+BIRCH: heterogeneous ensembles combining runs from the indicated clustering methods. Beneath each ensemble label, the corresponding mean ensemble diversity $\bar{D}$ is also reported.

The results in Fig. 14 indicate that the relation between ensemble diversity and CAKE score quality is nuanced rather than monotonic. In several cases, more diverse or heterogeneous ensemble constructions lead to a more informative confidence ranking (e.g., S2, S3, *Iris*, and *Satimage*), suggesting that structured variation among ensemble members can help CAKE better identify reliable assignments. At the same time, the figure also shows that increased diversity is not beneficial by itself: on some datasets, homogeneous constructions remain more effective (e.g., S7 and *Pendigits*), while highly diverse or poorly matched constructions can degrade both clustering quality and score quality (e.g., *Breast Cancer*). Additionally, in all the cases shown in Fig. 14, the mean score quality is higher than the mean consensus-clustering ARI. This is expected because the two quantities describe different aspects of performance. The mean

consensus-clustering ARI summarizes the quality of the final consensus clustering over the full dataset, averaged across repetitions, whereas score quality summarizes how well CAKE orders instances by looking at clustering quality over the higher-confidence retained subsets across coverage levels. At the same time, the two quantities often show similar trends across ensemble constructions, suggesting that stronger overall clustering structure is often accompanied by a more informative confidence ranking. The consistent gap therefore indicates that the points ranked near the top by CAKE tend to be more reliably assigned and easier to cluster than the dataset as a whole.

Taken together, these findings support the view that CAKE is robust across a wide range of ensemble constructions, but that the benefit of additional diversity depends on whether it captures useful complementary information rather than introducing unstable or incompatible clustering behavior.

D. Adaptive CAKE filtering

To complement the fixed-coverage instance-removal experiment in §5.2, an additional adaptive thresholding analysis is included here. In the main experiment, filtering is evaluated at a fixed retained coverage of 70%, which provides a controlled and directly comparable setting across datasets and confidence criteria, and is also practically relevant when a target retained fraction is specified in advance. This evaluation setting isolates ranking quality itself and allows it to be assessed separately from threshold-selection effects, since CAKE is intended primarily as a pointwise confidence score that induces a ranking over instances. A natural follow-up question is whether the retained fraction can also be selected in a fully data-driven manner rather than fixed in advance. The present section addresses that question.

Rather than fixing the retained fraction *a priori*, the retained coverage is selected automatically from a predefined grid

$$c \in \{0.40, 0.45, 0.50, \dots, 0.90\}\,.$$

For each candidate coverage c, the top-c fraction of instances is retained according to the ranking induced by CAKE$^{(HM)}$ (Eq. 8b), k-means is re-fitted on the retained subset, and the resulting partition is evaluated using the Silhouette score. The selected coverage is the one that maximizes this internal, label-free clustering-quality criterion over the subsets induced by the CAKE ranking. The standard sample-averaged Silhouette is used here, so the adaptive rule reflects overall subset quality at the instance level. In settings where balanced per-cluster retention is of greater interest, a macro-averaged variant could also be considered by first averaging Silhouette values within each cluster and then averaging across clusters, thereby reducing the influence of cluster size on coverage selection. After this retained size is determined, all competing filtering criteria (§5.2) are evaluated at the same matched subset size. This yields a simple adaptive filtering rule that remains fully unsupervised while avoiding a fixed global threshold.

This experiment serves a different purpose from the fixed-coverage analysis. The fixed-coverage analysis evaluates CAKE under a controlled retained size that is shared across methods, so that the comparison focuses directly on ranking quality. The adaptive analysis instead considers the setting in which the retained fraction is selected automatically from the data, and asks whether CAKE remains competitive under such a data-driven selection rule. Because the retained fraction is chosen by optimizing an internal clustering-validity criterion over a grid of candidate coverages, this analysis is presented as a supplementary adaptive filtering experiment. In particular, because the adaptive threshold is determined through the internal Silhouette objective, the comparison naturally becomes more favorable to the geometric component $\tilde{S}$ than in the fixed-coverage setting. This makes the adaptive setting a useful stress test: strong CAKE performance under this protocol is particularly informative, since the selection criterion itself is more favorable to geometry-driven filtering.

Table 13 reports representative results on six real-world datasets chosen to summarize the main empirical patterns observed under adaptive coverage selection: BC, DG, NG, FM, PD, and SA. Together, these datasets span multiple modalities, lower- and higher-dimensional settings, smaller and larger sample sizes, and problems with varying levels of structural complexity and class overlap. This allows the adaptive filtering behavior of CAKE to be examined under conditions in which the relative contribution of geometry and assignment stability may differ.

Several trends are consistent across these results. CAKE remains highly competitive under adaptive threshold selection, even though the procedure itself is based on internal Silhouette and therefore tends to strengthen geometry-based filtering rules. Under this more favorable setting for the geometric baseline, CAKE$^{(HM)}$ still attains the best performance on all three external metrics on DG and PD, while CAKE$^{(PR)}$ attains the best performance on all three metrics on NG

and FM. In addition, CAKE$^{(HM)}$ remains favorable on SA, where it achieves the best performance on two of the three reported metrics. Exact ties with the geometric baseline are observed on BC. At the same time, the confidence-based filtering criteria consistently remain well above the random baseline across all six datasets, indicating that the gains are not simply a consequence of using a smaller retained subset, but reflect a meaningful ordering of points by estimated assignment reliability.

Overall, the experiment supports the same qualitative conclusion as the fixed-coverage analysis: CAKE continues to provide a strong pointwise confidence ranking under an automatically chosen retained fraction, while the strongest individual filtering rule may still depend on dataset structure.

Table 13: Adaptive filtering with coverage selected by maximizing the Silhouette score over a fixed grid of retained fractions. For each dataset, the retained size is chosen from the CAKE$^{(HM)}$ ranking, and all competing filtering criteria are then evaluated at the same matched subset size. ARI, AMI, and ACC are reported as means over multiple independent runs, with 95% Student's t-confidence intervals shown in brackets. The highest reported value for each metric is shown in **bold**.

Dataset	Subset	Retained	ARI	AMI	ACC
BC	Full	569/569	0.663 [0.6558, 0.6698]	0.543 [0.5343, 0.5518]	0.908 [0.9058, 0.9101]
	Random	228/569	0.612 [0.6071, 0.6175]	0.484 [0.4786, 0.4892]	0.893 [0.8909, 0.8942]
	Consensus	228/569	0.831 [0.8233, 0.8393]	0.730 [0.7139, 0.7469]	0.957 [0.9553, 0.9596]
	C component	228/569	0.831 [0.8233, 0.8393]	0.730 [0.7139, 0.7469]	0.957 [0.9553, 0.9596]
	$\tilde{S}$ component	228/569	**0.924** [0.9239, 0.9239]	**0.851** [0.8515, 0.8515]	**0.987** [0.9868, 0.9868]
	CAKE$^{(PR)}$	228/569	**0.924** [0.9239, 0.9239]	**0.851** [0.8515, 0.8515]	**0.987** [0.9868, 0.9868]
	CAKE$^{(HM)}$	228/569	**0.924** [0.9239, 0.9239]	**0.851** [0.8515, 0.8515]	**0.987** [0.9868, 0.9868]
DG	Full	1797/1797	0.630 [0.6032, 0.6568]	0.731 [0.7204, 0.7421]	0.747 [0.7153, 0.7785]
	Random	719/1797	0.538 [0.5051, 0.5711]	0.677 [0.6563, 0.6967]	0.671 [0.6255, 0.7155]
	Consensus	719/1797	0.766 [0.7140, 0.8187]	0.833 [0.8098, 0.8560]	0.769 [0.7269, 0.8105]
	C component	719/1797	0.738 [0.6838, 0.7913]	0.821 [0.7969, 0.8450]	0.727 [0.6801, 0.7736]
	$\tilde{S}$ component	719/1797	0.808 [0.7486, 0.8673]	0.901 [0.8763, 0.9266]	0.780 [0.7281, 0.8316]
	CAKE$^{(PR)}$	719/1797	0.746 [0.6787, 0.8125]	0.884 [0.8602, 0.9074]	0.731 [0.6735, 0.7885]
	CAKE$^{(HM)}$	719/1797	**0.823** [0.7407, 0.9046]	**0.911** [0.8772, 0.9451]	**0.788** [0.7028, 0.8733]
NG	Full	18846/18846	0.352 [0.3433, 0.3612]	0.509 [0.5006, 0.5179]	0.513 [0.4992, 0.5259]
	Random	7538/18846	0.356 [0.3477, 0.3638]	0.514 [0.5076, 0.5212]	0.513 [0.4948, 0.5313]
	Consensus	7538/18846	0.465 [0.4459, 0.4844]	0.580 [0.5775, 0.5823]	0.532 [0.5205, 0.5427]
	C component	7538/18846	0.441 [0.4201, 0.4612]	0.570 [0.5685, 0.5721]	0.488 [0.4722, 0.5036]
	$\tilde{S}$ component	7538/18846	0.579 [0.5458, 0.6112]	0.699 [0.6904, 0.7081]	0.629 [0.6040, 0.6533]
	CAKE$^{(PR)}$	7538/18846	**0.592** [0.5602, 0.6246]	**0.701** [0.6877, 0.7134]	**0.639** [0.6090, 0.6695]
	CAKE$^{(HM)}$	7538/18846	0.567 [0.5310, 0.6030]	0.688 [0.6754, 0.7006]	0.626 [0.5921, 0.6603]
FM	Full	60000/60000	0.353 [0.3416, 0.3641]	0.506 [0.4966, 0.5163]	0.514 [0.4880, 0.5391]
	Random	24000/60000	0.356 [0.3433, 0.3681]	0.517 [0.5071, 0.5260]	0.505 [0.4835, 0.5263]
	Consensus	24000/60000	0.520 [0.4765, 0.5629]	0.598 [0.5798, 0.6165]	0.572 [0.5318, 0.6122]
	C component	24000/60000	0.397 [0.3822, 0.4120]	0.593 [0.5818, 0.6051]	0.492 [0.4805, 0.5038]
	$\tilde{S}$ component	24000/60000	0.540 [0.5070, 0.5724]	0.653 [0.6422, 0.6634]	0.599 [0.5660, 0.6311]
	CAKE$^{(PR)}$	24000/60000	**0.560** [0.5263, 0.5932]	**0.671** [0.6543, 0.6878]	**0.634** [0.6094, 0.6596]
	CAKE$^{(HM)}$	24000/60000	0.541 [0.5015, 0.5803]	0.663 [0.6468, 0.6793]	0.620 [0.5774, 0.6631]
PD	Full	10992/10992	0.541 [0.5121, 0.5690]	0.676 [0.6667, 0.6846]	0.669 [0.6326, 0.7057]
	Random	6046/10992	0.529 [0.5083, 0.5496]	0.672 [0.6644, 0.6793]	0.670 [0.6404, 0.6992]
	Consensus	6046/10992	0.727 [0.6996, 0.7553]	0.796 [0.7898, 0.8024]	0.739 [0.7083, 0.7690]
	C component	6046/10992	0.714 [0.6913, 0.7361]	0.789 [0.7720, 0.8050]	0.741 [0.7156, 0.7665]
	$\tilde{S}$ component	6046/10992	0.756 [0.7179, 0.7942]	0.858 [0.8423, 0.8737]	0.779 [0.7454, 0.8132]
	CAKE$^{(PR)}$	6046/10992	0.723 [0.6647, 0.7821]	0.833 [0.8093, 0.8559]	0.756 [0.7095, 0.8019]
	CAKE$^{(HM)}$	6046/10992	**0.771** [0.7213, 0.8203]	**0.861** [0.8418, 0.8804]	**0.796** [0.7533, 0.8393]
SA	Full	6430/6430	0.510 [0.4657, 0.5550]	0.595 [0.5556, 0.6346]	0.663 [0.6298, 0.6972]
	Random	2894/6430	0.524 [0.4974, 0.5504]	0.614 [0.5951, 0.6333]	0.670 [0.6526, 0.6869]
	Consensus	2894/6430	0.540 [0.5008, 0.5793]	0.597 [0.5799, 0.6134]	0.712 [0.6715, 0.7524]
	C component	2894/6430	0.540 [0.5008, 0.5793]	0.597 [0.5799, 0.6134]	0.712 [0.6715, 0.7524]
	$\tilde{S}$ component	2894/6430	0.722 [0.6517, 0.7919]	**0.780** [0.7466, 0.8125]	0.806 [0.7618, 0.8495]
	CAKE$^{(PR)}$	2894/6430	0.668 [0.5904, 0.7449]	0.748 [0.7020, 0.7946]	0.760 [0.6996, 0.8209]
	CAKE$^{(HM)}$	2894/6430	**0.731** [0.6573, 0.8048]	0.778 [0.7373, 0.8191]	**0.808** [0.7478, 0.8679]

E. Comparison Against GMM Confidence Signals

As a complement to the pointwise AUPRC/AUROC experiments in Tables 3 and 4, additional comparisons are reported here for native GMM-based confidence signals. In this setting, correctness is defined with respect to a reference Gaussian Mixture Model (GMM) fitted at the ground-truth number of components k. Let $\hat{z}_i^{\mathrm{GMM}}$ denote the label assigned to point x_i by the reference GMM, and let π denote the Hungarian alignment mapping to ground-truth labels. The binary target is then $\mathbb{1}\left[\pi(\hat{z}_i^{\mathrm{GMM}}) = y_i\right]$. As in the main pointwise evaluation, discriminative performance is measured using AUPRC and AUROC.

Three GMM confidence scores were considered. Let $p_{i\ell}$ denote the posterior responsibility of point x_i for component ℓ, with $\sum_{\ell=1}^{k} p_{i\ell} = 1$. The first score is maximum posterior confidence,

$$\mathrm{GMM}_i^{(1)} = \max_{\ell} p_{i\ell}, \tag{18}$$

the second is posterior margin,

$$\mathrm{GMM}_i^{(2)} = p_{i(1)} - p_{i(2)}, \tag{19}$$

where $p_{i(1)} \geq p_{i(2)}$ are the two largest posterior responsibilities, and the third is entropy-based confidence,

$$\mathrm{GMM}_i^{(3)} = 1 - \frac{H_i}{\log k}, \qquad H_i = -\sum_{\ell=1}^{k} p_{i\ell} \log\big(p_{i\ell} + \varepsilon\big). \tag{20}$$

These scores are compared against $\mathrm{CAKE}^{(\mathrm{HM})}$ (Eq. 8b) computed from a GMM ensemble. Specifically, an ensemble of $R = 20$ GMM runs is constructed using the same number of mixture components k and different random seeds, and $\mathrm{CAKE}^{(\mathrm{HM})}$ scores are computed from the resulting assignments in the same way as in the main experiments. The corresponding results for the real-world datasets (Table 1) are reported in Table 14.

Dataset	Metric	$\mathrm{GMM}^{(1)}$	$\mathrm{GMM}^{(2)}$	$\mathrm{GMM}^{(3)}$	$\mathrm{CAKE}^{(\mathrm{HM})}$
Ir	AUPRC	**0.997**	**0.997**	**0.997**	0.996
	AUROC	**0.927**	**0.927**	**0.927**	0.919
Bc	AUPRC	0.756	0.756	0.758	**0.947**
	AUROC	0.641	0.641	0.641	**0.915**
Dg	AUPRC	0.847	0.848	0.841	**0.967**
	AUROC	0.533	0.537	0.511	**0.853**
Ng	AUPRC	0.628	0.638	0.610	**0.639**
	AUROC	0.659	0.664	0.645	**0.687**
Fm	AUPRC	0.418	0.418	0.413	**0.684**
	AUROC	0.500	0.500	0.487	**0.715**
Pd	AUPRC	0.644	0.638	0.679	**0.924**
	AUROC	0.537	0.535	0.559	**0.858**
Lt	AUPRC	0.518	0.516	**0.544**	0.444
	AUROC	0.686	0.686	**0.689**	0.609
Sa	AUPRC	0.695	0.692	0.711	**0.806**
	AUROC	0.692	0.690	0.690	**0.747**

Table 14: Per-point GMM-correctness prediction (AUPRC and AUROC) with $\mathrm{GMM}^{(1)}$ (max posterior), $\mathrm{GMM}^{(2)}$ (posterior margin), $\mathrm{GMM}^{(3)}$ (entropy confidence), and $\mathrm{CAKE}^{(\mathrm{HM})}$ computed from GMM ensembles. The highest value for each dataset and metric is shown in **bold**.

The results in Table 14 show that $\mathrm{CAKE}^{(\mathrm{HM})}$ performs more strongly on most of the real-world datasets. In particular, it achieves the highest values on Bc, Dg, Ng, Fm, Pd, and Sa. This suggests that CAKE confidence scores derived from GMM ensembles can provide a more informative pointwise reliability signal than native posterior-based confidence scores alone.