Hard Possibilistic C-Means: An Empirical Study of Noise, Overlap, and Rejection

Paritosh Tiwari
Cyber-Physical Systems
Indian Institute of Science
Bengaluru, India
paritosht@iisc.ac.in

James C. Bezdek
Emeritus Professor
Australia
jcbezdek@gmail.com

Thomas A. Runkler
Siemens AG
Germany
thomas.runkler@siemens.com

Punit Rathore
Cyber-Physical Systems
Indian Institute of Science
Bengaluru, India
prathore@iisc.ac.in

***Abstract*—Hard Possibilistic C-Means (HPCM) is a prototype-based clustering method that replaces forced assignments with binary typicalities, allowing a point to be assigned to multiple clusters (overlap) or to none (rejection). This makes HPCM attractive in settings with ambiguity and background noise, but it also complicates evaluation and raises practical questions about when HPCM behaves differently from standard Hard C-Means (HCM). In this paper, we present an empirical study of HPCM through controlled synthetic experiments and small real-world benchmarks. We visualize how HPCM differs from HCM in overlap and noise regimes, and we analyze the sensitivity of HPCM to the scale parameter that defines its acceptance thresholds, showing how uniform background noise can inflate it and change the effective rejection/overlap behaviour. Since HPCM does not output a classical partition, we evaluate it using set-valued/reject-style metrics that capture the accuracy–coverage and accuracy–set-size tradeoffs. Overall, the results highlight both the intended advantages of HPCM (explicit ambiguity and noise filtering) and the conditions under which its behaviour can degenerate due to scale estimation.**

***Index Terms*—hard possibilistic c-means, possibilistic clustering, scale sensitivity, clustering evaluation**

I. Introduction

Hard c-means (HCM, i.e., k-means) is still the default baseline for prototype-based clustering because it is simple and fast, and it behaves well when clusters are compact and reasonably separated [3], [4]. But HCM has a structural limitation: every point must be assigned to exactly one cluster. So if the data contain bridge points, overlap regions, or background noise/outliers, HCM is forced to "explain" them by attaching them to some cluster anyway. In practice, this can contaminate clusters and also hides the fact that some points are genuinely ambiguous or should be treated as background.

Possibilistic c-means (PCM) was introduced to relax the forced-partition idea by replacing memberships with cluster-wise typicalities [2]. This makes it possible (at least in principle) to down-weight atypical points and reduce the influence of noise. However, baseline PCM is known to be sensitive to initialization and especially to the scale parameters (often denoted η), and it can produce coincident/redundant clusters if those scales are not well-controlled [2]. Recently, Bezdek and Runkler [1] introduced hard possibilistic c-means (HPCM) as a hard-limit counterpart of PCM, with a clean geometric interpretation. In HPCM, typicalities become binary via a distance threshold: a point can be typical for multiple clusters (multi-assignment) or for none (rejection). This is appealing because it makes two behaviours explicit that HCM cannot represent: overlap as a multi-hot label and noise as an all-zero label.

In this paper we try to understand, empirically and in a fairly controlled way, what HPCM is actually doing, and when it can be a better choice than HCM. Since HPCM outputs are not classical partitions, we also discuss evaluation choices that do not collapse its behaviour back into a forced single-label clustering. Our contributions are:

- *Controlled synthetic comparisons:* we visualize HCM vs. HPCM in regimes where they *match* (well-separated clusters) and *differ* (bridge points, background noise).
- *Scale parameter sensitivity under noise:* we isolate how uniform background noise inflates the η estimate and expands HPCM acceptance regions.
- *Real-data evaluation with set-valued metrics:* we benchmark HPCM on labelled datasets using *coverage*, *singleton rate*, and *penalized set accuracy*, since ARI/NMI do not fit multi-assignment and rejection outputs.

The rest of the paper is organized as follows. Section II reviews HPCM and related clustering formulations. Section III compares HCM and HPCM through three focused experiments designed to expose rejection and overlap. Section IV studies the noise sensitivity of the η heuristic used in HPCM, and evaluates HPCM on real datasets using metrics that account for set-valued outputs.

II. Background

A. Hard possibilistic c-means (HPCM)

We use the *Hard Possibilistic C-Means* (HPCM) algorithm of Bezdek and Runkler [1]. Let $X = \{x_k\}_{k=1}^{n} \subset \mathbb{R}^p$ be the data, $V = \{v_i\}_{i=1}^{c}$ the prototypes, and $U = [u_{ik}]$ a binary typicality matrix. The key step in HPCM is a hard accept/reject rule:

$$u_{ik} = \begin{cases} 1, & \|x_k - v_i\|^2 < \eta_i \\ 0, & \|x_k - v_i\|^2 > \eta_i, \end{cases} \tag{1}$$

so $U \in \{0,1\}^{c\times n}$. Geometrically, each cluster i induces an acceptance region (an η_i-ball) around v_i; points can fall into

multiple balls (overlap) or none (rejection/noise), which HCM cannot represent.

Following [1], we estimate the distance scale for each cluster as the mean squared distance of its currently accepted points:

$$\eta_i = \frac{\sum_{k=1}^{n} u_{ik} \|x_k - v_i\|^2}{\sum_{k=1}^{n} u_{ik}} \quad \text{when} \sum_{k=1}^{n} u_{ik} > 0. \tag{2}$$

For convenience, the authors in [1] use a single shared threshold across clusters, and we follow the same convention throughout our experiments (i.e., $\eta_i \equiv \eta$). Unless stated otherwise, η is recomputed after each prototype update from the currently accepted points using Equation 2; if a cluster has no accepted points, its prototype and η are left unchanged, and each one-cluster extraction is run until assignments stop changing or a fixed iteration cap is reached.

Given U, prototypes are updated as the centroid of the selected points,

$$v_i = \frac{\sum_{k=1}^{n} u_{ik} x_k}{\sum_{k=1}^{n} u_{ik}} \quad \text{when} \sum_{k=1}^{n} u_{ik} > 0. \tag{3}$$

HPCM is constructed sequentially. After estimating one cluster, the next one is initialized only from leftover points that have not been accepted by any earlier cluster. Concretely, we use the leftover mask initialization from [1]:

$$u_{ik}^{(0)} = \begin{cases} 1, & i = 1 \text{ or } \sum_{j<i} u_{jk} = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The initialization can in principle depend on the data permutation through the leftover mask, although in our implementation the final solution was empirically stable across permutations on the datasets considered.

B. Related work

Hard prototype-based clustering (HCM/k-means) is a natural baseline for squared-error clustering: it alternates between nearest-prototype assignment and centroid updates [2], [3], but enforces a strict partition in which every point must belong to exactly one cluster. As a result, overlap, bridge points, and explicit rejection cannot be represented. *Fuzzy c-means* (FCM) relaxes this through soft memberships while retaining the column-sum-to-one constraint [4]–[6], which allows overlap to be modelled, but still forces even noisy or outlying points to distribute membership mass across clusters. *Possibilistic c-means* (PCM) removes the simplex constraint and instead uses cluster-wise typicalities [7], which in principle allows atypical points to be down-weighted or ignored. In practice, however, PCM is known to be sensitive to initialization and especially to the scale parameter(s) η, and may produce coincident or redundant clusters when η is poorly chosen.

Hybrid fuzzy–possibilistic objectives have been proposed to combine FCM-style overlap modelling with PCM's robustness motivation. A representative example is possibilistic-fuzzy c-means (PFCM), which optimizes both memberships and typicalities to reduce FCM's noise sensitivity while mitigating common PCM pathologies such as coincident clusters [8]. Related sequential possibilistic methods extract clusters one at a time rather than solving for all c clusters jointly; sequential possibilistic one-means (SP1M), for example, repeatedly runs a possibilistic one-cluster solver and then moves on to the next cluster [9], again highlighting that controlling η is central to stability and quality. Prototype-based fuzzy clustering has also been extended with adaptive, often Mahalanobis-type, metrics to capture ellipsoidal structure, with the Gustafson–Kessel approach as a canonical example [10]. While our experiments use Euclidean distances for controlled comparisons, these formulations help situate HPCM within the broader prototype-based clustering literature.

To connect the above formulations to the behaviours studied later, we begin with a clean baseline setting in which clusters are well separated and centroid-based assumptions hold. We generate three well-separated 2D Gaussian clusters and run HCM, FCM, PCM, and HPCM using standard baseline choices. In this setting, HCM, FCM, and HPCM all recover the three clusters essentially perfectly (Figure 1), and HPCM behaves much like a classical nearest-centre partition because there is no overlap and no reason to reject points. PCM, however, can still behave worse even on this easy mixture, mainly because its updates are governed by fixed scales η and non-competitive typicalities [7]. This baseline is not intended as a stress test; rather, it clarifies an important point for the rest of the paper: the distinctive behaviours of HPCM only become apparent once overlap, bridge points, and background noise are introduced.

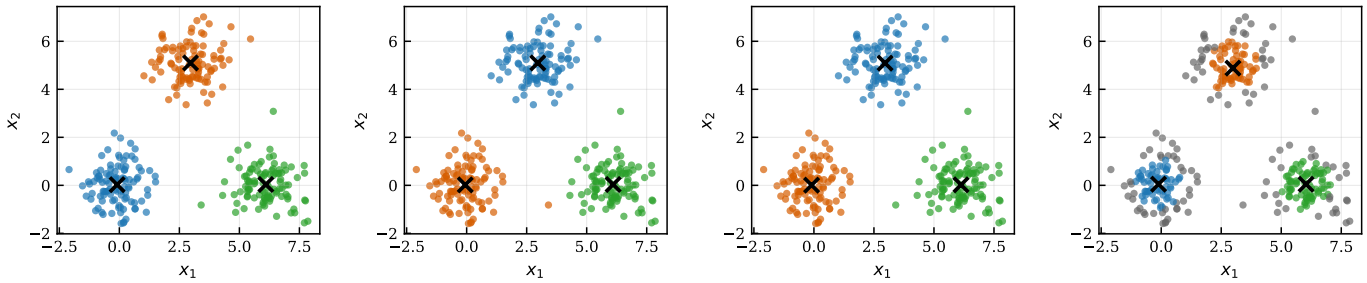
III. HCM and HPCM: A Comparative Analysis

HCM produces a forced partition in which every point belongs to exactly one of c clusters. HPCM relaxes this by allowing (i) *multi-cluster assignment* (a point can be typical to multiple clusters) and (ii) *rejection* (a point can be assigned to no cluster). In this section, we use three controlled experiments to study when HPCM can be a better choice than HCM.

A. Experiment 1: Clean two-cluster data (convergence behaviour and conservativeness)

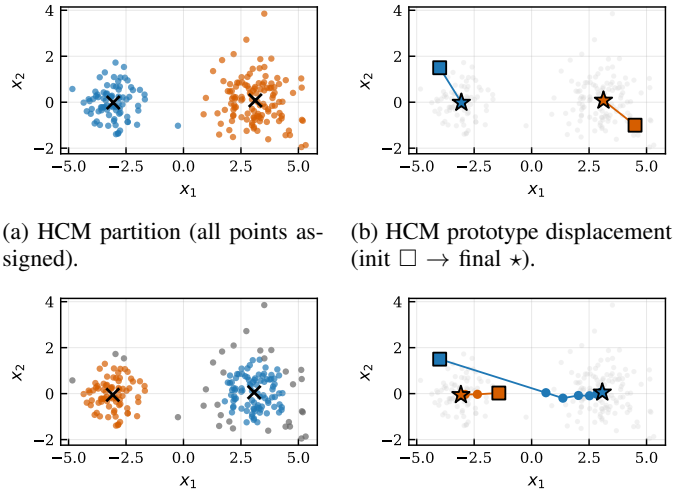
This experiment starts with a clean two-cluster setting (200 points total) designed mainly to compare behaviour rather than robustness. Figure 2 shows that HCM assigns all points and yields cluster sizes $[81, 119]$. HPCM calculates $\eta = 2.1257$ and produces cluster sizes $[91, 77]$, with 32 unassigned points and 0 overlap (no point is simultaneously typical to both clusters). These outcomes are consistent with the basic semantics which say that HCM is compelled to explain every point, while HPCM is allowed to reject points that are not sufficiently typical under the $\|x - v_i\|^2 \leq \eta$ test.

Qualitatively, Figure 2a shows the clean forced partition from HCM, and Figure 2c shows HPCM keeping compact cores but leaving a noticeable boundary/tail region unassigned (gray). Here, HPCM is *more conservative* than HCM, and even though the data are clean, $32/200 = 16\%$ points are rejected. This highlights a practical point for HPCM, which



(a) Hard C-Means (b) Hard Possibilistic C-Means (c) Fuzzy C-Means (d) Possibilistic C-Means

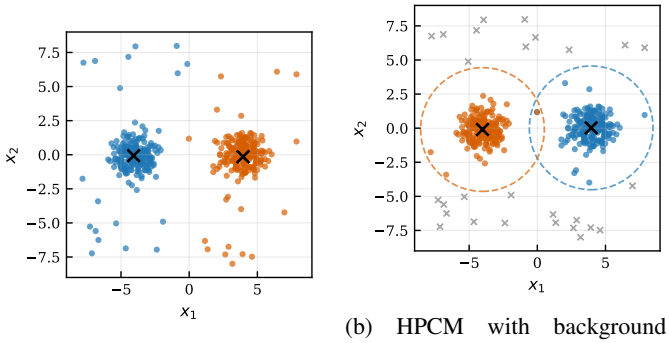
Fig. 1: Clustering results on three well separated Gaussian clusters.



(a) HCM partition (all points assigned). (b) HCM prototype displacement (init □ → final ⋆).

(c) HPCM partition (unassigned points in gray). (d) HPCM prototype trajectories (init □, iterates ○, final ⋆).

Fig. 2: Experiment 1 (clean two-cluster data): HCM vs. HPCM. (a–b) HCM assigns all points; (c–d) HPCM leaves some points unassigned (gray). Black × denote learned prototypes. In prototype plots, □ is initialization, ○ intermediate iterates, and ⋆ the final prototype.



(a) HCM with background noise (forced assignment). (b) HPCM with background noise (rejection via η-acceptance regions).

Fig. 3: Experiment 2 (two clusters with uniform background noise): (a) HCM assigns all points, including noise. (b) HPCM rejects non-typical points (gray ×) using η-based acceptance regions (dashed circles).

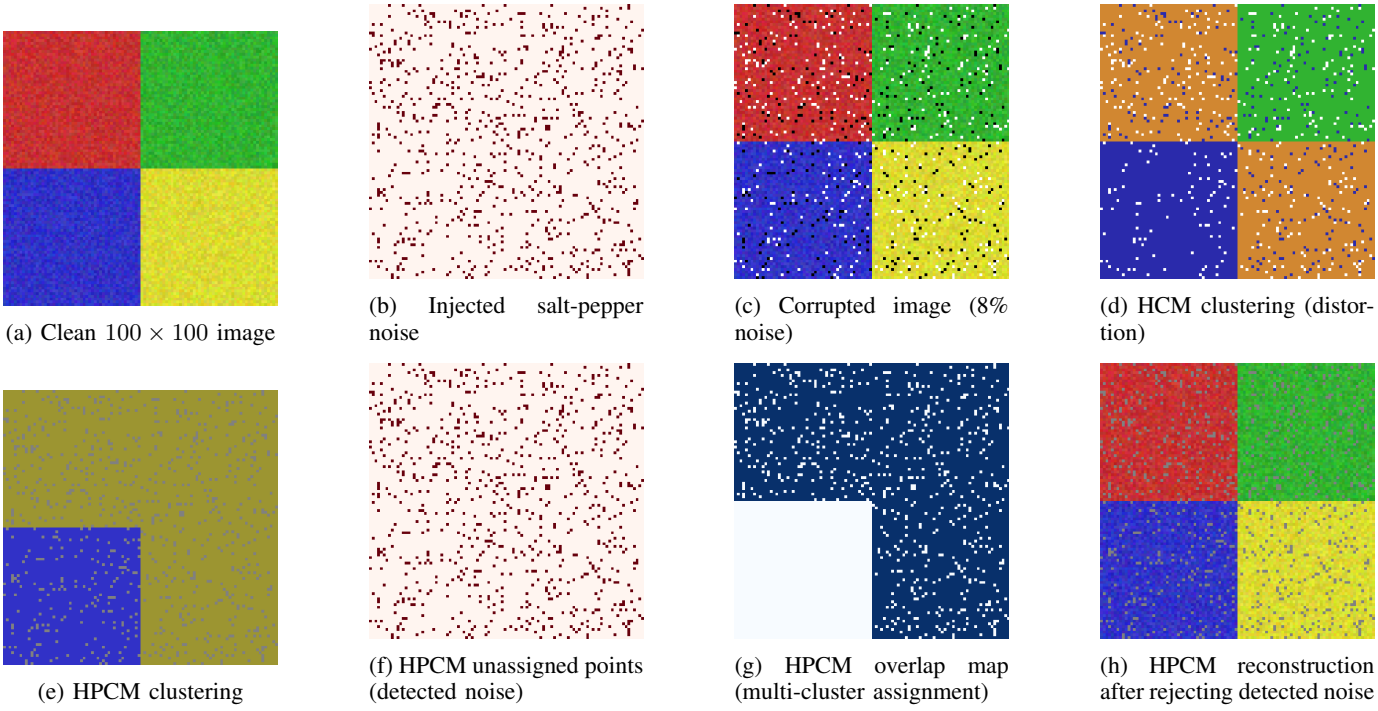
is that with a single global η, it can be too strict for the more dispersed parts of a cluster and will reject legitimate tail points. The trajectory plots (Figures 2b, 2d) also support the expected optimization behaviour. HCM moves both centres under a global reassignment step, while HPCM updates clusters sequentially and its centre paths reflect repeated thresholding and reassignment within each cluster.

B. Experiment 2: Two clusters with uniform background noise (noise filtering)

This experiment adds a background noise component and is the regime where HPCM is expected to help. Figure 3 shows the outcomes. There are 35 true noise points and 435 total points. HCM, by design, assigns all 435 points to the two clusters (Figure 3a, which necessarily spreads noise across clusters and contaminates cluster support. HPCM rejects 27 points as noise with $\eta = 20.5764$ (Figure 3b). Importantly, in the illustrated run, all 27 rejected points are true noise points, i.e., 27/35 noise points detected and 0 false rejections. The corresponding detection precision is $27/27 = 100\%$ and the recall is $27/35 \approx 77.1\%$. For this setting, HPCM is strictly more expressive than HCM. It separates cluster-typical points from a subset of the background points without any additional post-processing. The trade-off is that a small number of noise points remain assigned (8 out of 35), but the result still highlights the intended behaviour: HPCM can function as an intrinsic noise gate, whereas HCM lacks any mechanism to represent rejection.

C. Experiment 3: Synthetic image with salt-and-pepper noise (explicit noise detection)

This experiment considers a pixel-clustering task on a 100×100 image (10,000 pixels) composed of four regions (R/G/B/Y) with 800 corrupted pixels corresponding to 8% salt-and-pepper noise (Figure 4). HPCM is run using the algorithmic setting in [1] and calculates $\eta = 19637.5$. In the illustrated run, the algorithm rejects exactly 800 pixels (8.0%), all of which correspond to injected noise, i.e., 800/800 noisy pixels are detected with zero false rejections. At the same time, 6,878 pixels (68.8%) are assigned to multiple clusters,



(a) Clean 100×100 image

(b) Injected salt-pepper noise

(c) Corrupted image (8% noise)

(d) HCM clustering (distortion)

(e) HPCM clustering

(f) HPCM unassigned points (detected noise)

(g) HPCM overlap map (multi-cluster assignment)

(h) HPCM reconstruction after rejecting detected noise

Fig. 4: Experiment 3 on a 100×100 synthetic image with four regions and 8% salt-and-pepper noise. HCM absorbs the corrupted pixels and shows visible distortion, whereas HPCM rejects all 800 noisy pixels with no false rejections. HPCM also produces substantial overlap (68.8%) and reconstructs 92.0% of pixels correctly.

reflecting overlap across region boundaries, while 9,200 out of 10,000 pixels (92.0%) are correctly recovered. Overall, this provides a clear demonstration, in this controlled example, of HPCM's rejection semantics. All injected noise is identified as not typical for any cluster, while no clean pixels are discarded. At the same time, this experiment exposes another practical issue, overlap can become very large (here 68.8%) when η is large relative to the separation between regions in the chosen feature space. In images, many pixels can be close in raw RGB (or whatever features are used), so it is common for $\|x - v_i\|^2 \leq \eta$ to hold for multiple clusters. This is not necessarily a failure; it is HPCM reporting ambiguity. But if a downstream task needs a single label per pixel, the overlap must be resolved (e.g., pick the nearest typical centre) or reduced by using a stricter η. The 92% recovery is consistent with this: overlap indicates multi-eligibility under the threshold, not directly incorrect recovery.

Overall, these controlled experiments show that HPCM is most useful when explicit rejection or overlap is desired, especially in the presence of background noise. When a downstream task requires a strict partition, HCM remains the simpler default, or HPCM overlap can be resolved afterward by assigning the nearest typical prototype.

IV. Evaluation on Noisy and Real-World Data

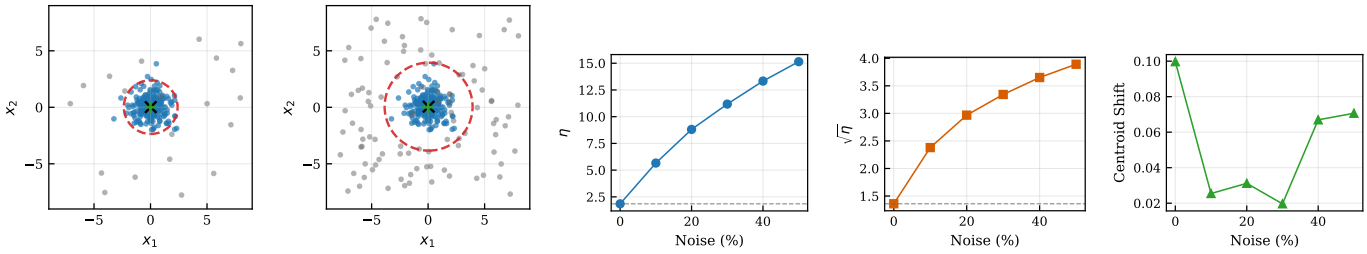
A. Noise sensitivity of η in HPCM

In HPCM, the scale η_i is defined in Equation 2 from the points currently accepted by cluster i. In the one-cluster noise experiment below, to isolate how background noise changes the effective acceptance radius, we track the corresponding global mean-squared-distance quantity

$$\eta = \frac{1}{n} \sum_{k=1}^{n} \|x_k - v\|^2, \tag{5}$$

as an auxiliary diagnostic, rather than as a separate update rule.

To isolate this effect, we generate a single 2D Gaussian cluster ($n = 200$, centre $(0, 0)$, $\sigma = 1$) and add uniform background noise from $[-8, 8]^2$ at increasing fractions (0% to 50%). We run HPCM with $c = 1$ for a small fixed number of iterations and track $\eta = \frac{1}{n} \sum_k \|x_k - v\|_2^2$, the corresponding radius $\sqrt{\eta}$, and the centroid shift $\|v - (0, 0)\|_2$. As noise increases, η inflates monotonically from 1.85 (0% noise) to 15.15 (50% noise), i.e., about an $8\times$ increase, and the radius expands from 1.36 to 3.89 (see Figs. 5c–5d). This happens even though the true inlier cluster is unchanged, so the effect is really from the *estimator* mixing inliers and outliers. The centre itself stays stable in this one-cluster setting (centroid shift ≤ 0.10 across noise levels; Fig. 5e), but the membership decisions change through η: with 0% noise only $128/200$ inliers fall inside the (small) η-ball, while at 50% noise essentially all inliers are accepted ($200/200$), and a small (non-zero) number of noise points are also admitted (e.g., $13/100$ at 50%). Overall, this shows a practical issue: the common global mean-squared-distance rule for η can become overly permissive under background noise, and in multi-cluster



(a) 10% noise, $\sqrt{\eta} = 2.38$ (b) 50% noise, $\sqrt{\eta} = 3.89$ (c) η inflation (d) η-ball radius (e) Centroid drift

Fig. 5: Noise sensitivity of η in HPCM: A single Gaussian cluster with increasing uniform background noise. Blue points denote inliers, grey points denote noise, the green marker is the learned centre, and the red dashed circle is the η-ball of radius $\sqrt{\eta}$.

($c > 1$) sequential extraction this can propagate into larger overlap and more noise capture.

B. HPCM Evaluation Criteria

A practical issue in evaluating HPCM on labelled benchmarks is that its output is not a classical partition. In HCM, every point receives exactly one label, so external indices like Adjusted Rand Index (ARI) [11] and Normalized Mutual Information (NMI) [12] are well-defined and widely used. In HPCM, a column can be *all zeros* (rejected point) or can contain *multiple ones* (overlap / multiple assignment), which are both valid behaviours in the hard possibilistic model [1]. If we force HPCM into a single-label partition (e.g., by hardening with arbitrary tie-breaking), then ARI/NMI mostly measure the *hardening heuristic*, not the possibilistic behaviour we actually want to quantify. So, instead of ARI/NMI, we evaluate HPCM as a *set-valued predictor with a reject option*, which is a standard framing in classification literature. Set-valued classification explicitly studies the accuracy–set-size trade-off [13], and selective classification (classification with reject) studies the risk–coverage trade-off [14], [15].

Let $y_k \in \{1, \dots, c\}$ be the ground-truth class of point x_k. HPCM returns a binary membership matrix $U \in \{0,1\}^{c \times n}$. We convert column k into a predicted label set $S_k = \{\, i \in \{1, \dots, c\} : u_{ik} = 1 \,\}$. Thus, $|S_k| = 0$ means *reject* and $|S_k| > 1$ means *overlap*. As usual for external evaluation, we align cluster IDs to class IDs using a one-to-one mapping (Hungarian matching), and interpret S_k after this alignment.

Following the motivations above, we report a compact set of metrics that separately measure *rejection*, *ambiguity/overlap*, and *correctness* when the method commits.

a) Coverage (non-reject rate).:

$$\text{Coverage} = \frac{1}{n} \sum_{k=1}^{n} \mathbb{I}\big[|S_k| \geq 1\big]. \tag{6}$$

This is the key quantity in selective classification: by allowing rejection, one can trade coverage for lower error on the covered subset [14], [15].

b) Singleton rate (commitment rate).:

$$\text{Sing.Rate} = \frac{1}{n} \sum_{k=1}^{n} \mathbb{I}\big[|S_k| = 1\big]. \tag{7}$$

TABLE I: Summary of datasets.

Dataset	Samples (n)	Features (d)	Classes (c)
Iris	150	4	3
Wine	178	13	3
Glass	214	9	6
Ecoli	336	7	8
Yeast	1484	8	10

This tells us how often HPCM behaves like a classical hard partition (one label per point). It is a simple proxy for how often the output is decisive.

c) Singleton accuracy (accuracy when it commits).:

$$\text{Acc}_{\text{sing}} = \frac{\sum_{k=1}^{n} \mathbb{I}[|S_k| = 1]\, \mathbb{I}[y_k \in S_k]}{\sum_{k=1}^{n} \mathbb{I}[|S_k| = 1]}. \tag{8}$$

If $\text{sing} = 0$, this is undefined (we report N/A). This number is useful because it isolates the reliable core predictions, which is exactly the spirit of reject-option models [14].

d) Penalized set accuracy (accuracy-set size trade-off).:

$$\text{Acc}_{\text{pen}} = \frac{1}{n} \sum_{k=1}^{n} \frac{\mathbb{I}[y_k \in S_k]}{|S_k| + \epsilon}, \qquad \epsilon \approx 10^{-12}. \tag{9}$$

Set-valued classification emphasizes that containing the true label is not enough; the set size must also be accounted for [13]. The $1/|S_k|$ factor directly penalizes overlap. Predicting all labels for every point cannot score well.

e) Average set size on covered points.:

$$|S|_{\text{cov}} = \frac{\sum_{k=1}^{n} |S_k|\, \mathbb{I}[|S_k| \geq 1]}{\sum_{k=1}^{n} \mathbb{I}[|S_k| \geq 1]}. \tag{10}$$

This quantifies how ambiguous the non-rejected decisions are. Together with coverage, it summarizes whether HPCM is mostly filtering (rejecting) or hedging (assigning to many clusters).

C. Results and Discussion

1) Datasets: We evaluate on five well-known real-world benchmark datasets summarized in Table I. The Iris dataset [16] contains 150 samples of flower measurements with four real features and three class labels, and Wine [17] consists of 178 samples with 13 continuous features and three class

TABLE II: HPCM performance on real-world datasets.

Data	Cov.	Sing.R.	$\mathrm{Acc}_{\mathrm{sing}}$	$\mathrm{Acc}_{\mathrm{pen}}$	$\lvert S\rvert_{\mathrm{cov}}$	HCM Acc
Iris	0.913	0.320	1.000	0.617	1.650	0.784
Wine	0.517	0.000	–	0.172	3.000	0.924
Glass	0.701	0.000	–	0.117	6.000	0.461
Ecoli	0.676	0.000	–	0.084	8.000	0.583
Yeast	0.747	0.000	–	0.075	10.000	0.406

labels representing wine cultivars. Glass Identification has nine continuous attributes and six glass types (for a forensic classification task) [18], Ecoli includes seven real-valued features for protein cellular localization [19], and Yeast comprises eight features for predicting protein subcellular localization [20]. All datasets are used in a standard multi-class clustering/evaluation setting without manual preprocessing beyond standardization.

2) Results: Table II shows a consistent pattern. On the Iris dataset, HPCM has reasonably high coverage (≈ 0.91) and a non-trivial singleton rate (≈ 0.32). More importantly, $\mathrm{Acc}_{\mathrm{sing}} = 1.0$, which means whenever HPCM commits to a single class, it is essentially always correct. This is exactly the kind of reliable subset behaviour that reject-option and selective classification are designed to expose. The model is conservative, but its committed decisions are high precision [14], [15]. At the same time, the modest $|S|_{\mathrm{cov}} \approx 1.65$ indicates limited overlap among covered points, so the good $\mathrm{Acc}_{\mathrm{pen}}$ is not coming from predicting large sets. For Wine, Glass, Ecoli and Yeast datasets, the behaviour is different: $\mathrm{Sing.Rate} = 0$ (no singleton predictions), and $|S|_{\mathrm{cov}}$ equals c which is the class distribution (e.g., $3, 6, 8, 10$), meaning that whenever a point is covered, it is assigned to *all* clusters. In this regime, $\mathrm{Acc}_{\mathrm{sing}}$ is undefined (correctly reported as N/A), and $\mathrm{Acc}_{\mathrm{pen}}$ becomes small because the $1/|S_k|$ penalty is maximal. This suggests that, on these datasets, the shared η inherited from [1] is too permissive relative to inter-prototype separations after standardization, so HPCM collapses into near-universal overlap rather than producing informative singleton assignments. Importantly, this does *not* look artificially good under our metrics. The penalized accuracy prevents the "predict everything" strategy from scoring well [13].

Table II reports mean over several experiments seeds to match the usual evaluation protocol, but for HPCM it is not very informative. In our setup, and as per the algorithm defined in [1], the only source of randomness is the data permutation used for the sequential-style initialization; the η computation and the hard updates are deterministic. Empirically, HPCM converges to the same fixed point for different permutations on these datasets (std ≈ 0 in all five metrics). So, unlike HCM (where random seeding can meaningfully change the solution), repeated runs for HPCM mostly confirm permutation-invariance in this implementation rather than providing a genuine variance estimate.

V. Conclusion

We presented an empirical study of hard possibilistic c-means (HPCM) [1] and contrasted it with hard c-means (HCM) to clarify what HPCM adds in practice. The main advantage is representational: HPCM can explicitly mark overlap via multi-assignment and filter background points via rejection, instead of forcing every sample into exactly one cluster as HCM does. Our experiments also show a key sensitivity: the behaviour is strongly driven by the η scale estimate that defines the acceptance threshold, and background noise can distort this estimate and push HPCM toward either excessive rejection or excessive overlap. On labelled benchmarks, set-valued/reject-style metrics are more aligned with HPCM than ARI/NMI, and they reveal that HPCM sometimes yields a high-precision "core" but can also degenerate into overlap-heavy assignments. A natural next step is to make η estimation more robust or explicitly tune a scalar multiplier on η, since the real-data results suggest that an overly large shared threshold can drive HPCM toward all-cluster assignments. If a downstream task requires a partition, overlap can be resolved after HPCM by assigning the nearest typical prototype, or reduced beforehand by using a stricter acceptance threshold.

References

[1] J. C. Bezdek and T. A. Runkler, "Geometric foundations of possibilistic clustering: A hard possibilistic clustering algorithm," *Fuzzy Sets and Systems*, p. 109775, 2026.

[2] S. P. Lloyd, "Least squares quantization in pcm," *IEEE Trans. Inf. Theory*, vol. 28, pp. 129–136, 1982.

[3] J. MacQueen, "Multivariate observations," in *Proceedings ofthe 5th Berkeley Symposium on Mathematical Statisticsand Probability*, vol. 1, 1967, pp. 281–297.

[4] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[5] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers Geosciences*, vol. 10, no. 2, pp. 191–203, 1984.

[6] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, 1st ed., ser. Advanced Applications in Pattern Recognition. New York, NY: Springer, 1981.

[7] R. Krishnapuram and J. Keller, "A possibilistic approach to clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 98–110, 1993.

[8] N. Pal, K. Pal, J. Keller, and J. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," *Fuzzy Systems, IEEE Transactions on*, vol. 13, pp. 517 – 530, 09 2005.

[9] T. A. Runkler and J. M. Keller, "Sequential possibilistic one-means clustering," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE Press, 2017, p. 1–6.

[10] D. Gustafson and W. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," vol. 2, 02 1978, pp. 761 – 766.

[11] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[12] A. Strehl and J. Ghosh, "Cluster ensembles — a knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, no. null, p. 583–617, Mar. 2003.

[13] E. Chzhen, C. Denis, M. Hebiri, and T. Lorieul, "Set-valued classification – overview via a unified framework," 2021.

[14] R. El-Yaniv and Y. Wiener, "On the foundations of noise-free selective classification," *Journal of Machine Learning Research*, vol. 11, no. 53, pp. 1605–1641, 2010.

[15] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 41–46, 1970.

[16] R. A. Fisher, "Iris," UCI Machine Learning Repository, 1936.

[17] S. Aeberhard and M. Forina, "Wine," UCI Machine Learning Repository, 1992.

[18] B. German, "Glass Identification," UCI Machine Learning Repository, 1987.

[19] K. Nakai, "Ecoli," UCI Machine Learning Repository, 1996.

[20] ——, "Yeast," UCI Machine Learning Repository, 1991.