LET'S MAKE SOME NOISE: THE CRITICAL STATE OF EXTERNAL EVALUATION OF CLUSTERINGS WITH NOISE

Anonymous authors
Paper under double-blind review

ABSTRACT

Detecting noise is an important and beneficial feature of frequently used clustering algorithms like DBSCAN or HDBSCAN. However, the external evaluation of clusterings containing noise labels is handled inconsistently in the literature: as common external cluster validity indices (CVIs) like NMI or ARI are not defined for noise labels, a variety of ways to handle noise has emerged, most without proper foundations or explicit description. In this survey, we examine i) how papers presenting noise-detecting clustering methods evaluate their methods, and ii) the datasets that have been used. We define four desirable properties for noise-aware external CVIs and perform experiments across a wide range of existing noise-handling strategies. By highlighting where these properties are violated, we show that evaluation in this field is currently neither uniform nor scientifically adequate. Based on these findings, we propose two new noise-aware CVIs, NARI_κ and NARI_φ (Noise-Aware ARI), combining ARI on cluster points with Cohen's κ and the Matthews–Pearson φ coefficient on the binary noise classification, respectively. We show that both satisfy all four properties and confirm on real data, with held-out minority classes and foundation-model-annotated images as noise, that flawed strategies certify degenerate clusterings with perfect scores while the proposed measures do not.

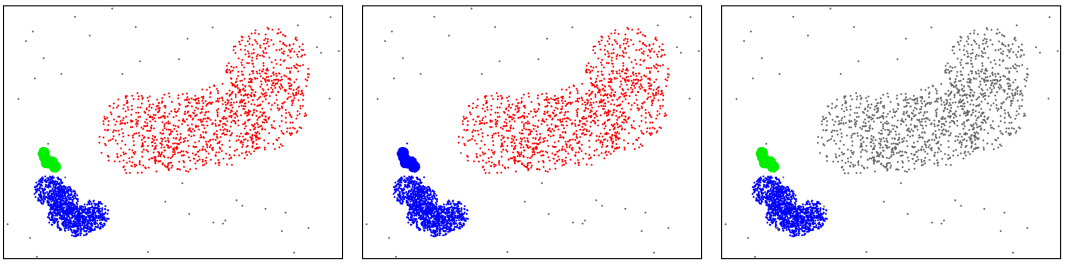
1 INTRODUCTION

External evaluation against ground-truth (GT) labels is how progress in clustering is measured: it decides which method works best for a given kind of data and whether a new method beats the state of the art. Because clustering is unsupervised, model selection cannot rely on labeled validation data, making rigorous external evaluation the primary mechanism for comparing methods. Recent clustering papers agree on the measures to use, namely the Adjusted Rand Index (ARI) and mutual-information-based measures (AMI, NMI) (Table 2). Density-based methods such as DBSCAN and HDBSCAN are evaluated with the same measures, although automatic noise detection is one of their key features and matters in applications such as astrophysics (Tramacere & Vecchio, 2013) or geospatial data (Birant & Kut, 2007). The *quality of the noise labeling* is usually neglected: dominant paradigms such as k-means do not label noise, so this aspect is absent from the evaluation conventions they established.

Terminology *Noise objects* are a property of the data: points that belong to no cluster and usually stem from a different distribution than the cluster points (Section 2 delineates them from outliers and noisy feature values); they are not defined by any algorithm's output. *Ground-truth (GT) noise labels* mark these objects in a benchmark's reference labeling, which external evaluation compares against. *Predicted noise labels* are the flags an algorithm outputs for points it assigns to no cluster. *Noise detection* is the induced binary prediction problem, predicted versus GT noise labels. Evaluating a clustering with noise labels therefore has to assess two things at once: the partition of the cluster objects and the quality of noise detection.

Neither ARI nor NMI, the two standard cluster validity indices (CVIs), is defined for noise labels, and naive application misleads. Figure 1 shows two flawed DBSCAN results on data with three clusters of varying density and 1% background noise: (b) merges the two denser clusters, (c) labels

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) $\mathrm{ARI}_c = 1.00$, $\mathrm{NMI}_c = 1.00$ (b) $\mathrm{ARI}_c = 0.58$, $\mathrm{NMI}_c = 0.74$ (c) $\mathrm{ARI}_c = 0.98$, $\mathrm{NMI}_c = 0.97$

Figure 1: Three clusters of varying density with 1% uniform background noise: (a) ground truth; (b) DBSCAN merges the two denser clusters; (c) DBSCAN labels the sparsest cluster entirely as noise. Standard implementations of ARI and NMI (subscript c) treat all noise labels as one regular cluster and score (c) near-perfectly, although it misses an entire cluster.

the sparsest cluster entirely as noise. Standard implementations (ARI_c, NMI_c), which treat all noise labels as one additional cluster, give the arguably worse clustering (c) a near-perfect score while penalizing (b) heavily, suggesting that DBSCAN recovered the structure although it discarded an entire cluster. Our literature review of how noise-labeling clustering methods are evaluated (Table 2) uncovers three recurring problems:

1. In many publications, noise labeling is not evaluated at all, or a CVI is used without stating how noise labels are handled.
2. Otherwise, ad-hoc noise-handling strategies are used that were never designed or tested for noise labels.
3. These strategies fail systematically: they reward clusterings that label most points as noise (Table 11) and score objectively worse clusterings above good ones (Figure 1).

Our contributions are: (1) a survey of the evaluation practice and the benchmark datasets of the field (Tables 2 and 6); (2) four formally stated properties for noise-aware external CVIs and a systematic evaluation of all straightforward noise-handling strategies against them (Table 1); (3) two new measures, NARI_κ and NARI_φ (Noise-Aware ARI), that provably satisfy all four properties (unbiasedness asymptotically; Section B), whereas the best existing strategy satisfies monotonicity only empirically; (4) evidence on real data, with held-out minority classes and foundation-model-annotated images as GT noise, that flawed strategies certify degenerate clusterings with perfect scores and change which method a comparison declares best (Section 3.7); (5) a Python library[1] packaging all of the above as a reusable benchmark suite.

2 State Of the Art in External Evaluation of Clusterings with Noise

Noise is associated with three different concepts: **Noisy feature values** are random perturbations of the coordinates; such points still belong to a cluster (Chaudhuri & Bhowmik, 1998). **Outliers** deviate significantly from the rest of the data, globally (rare events such as fraud) or locally (in density or neighbor distance); outlier detection isolates such rare aberrations rather than partitioning the data (Chandola et al., 2009). **Noise points** belong to no cluster and are typically modeled as a uniform background distribution (Ester et al., 1996; Campello et al., 2015). They are the noise objects of Section 1 and the focus of this paper. With *noise* we refer to noise points, and clustering methods should ideally assign predicted *noise labels* exactly to the points carrying GT noise labels.

2.1 Clustering algorithms identifying noise and benchmark datasets

Labeling objects that belong to no cluster as noise, present already in model-based clustering (Banfield & Raftery, 1993), was popularized by DBSCAN (Ester et al., 1996), extended by OPTICS (Ankerst et al., 1999) and HDBSCAN (Campello et al., 2015), and is a standard feature of density-based

[1] https://anonymous.4open.science/r/External-Evaluation-of-Clusterings-with-Noise-Labels-F777

clustering, e.g., in SS-DBSCAN (Abdulhameed et al., 2024), DP-DBSCAN (Qiu & Yi, 2025), RNN-DBSCAN (Bryant & Cios, 2018), BSA-DBSCAN (Wang et al., 2021), KR-DBSCAN (Hu et al., 2021), MDBSCAN (Wang et al., 2016), FairDen (Krieger et al., 2025), SHADE (Beer et al., 2024), DMDHC (Durani et al., 2025), DBHD (Durani et al., 2022), and DPA (d'Errico et al., 2021). Noise labeling is not exclusive to density-based methods: Sync (Böhm et al., 2010) flags points that fail to synchronize with any group, Shen et al. (2010) model an explicit noise component, and SkinnyDip (Maurus & Plant, 2016) flags points outside the detected unimodal regions. Table 2 summarizes how all these methods are evaluated.

Most labeled benchmark datasets originate from classification tasks (Table 5), where every object is assigned exactly one class label; the most common clustering benchmarks therefore lack GT noise, even where the data contain ambiguous objects that a noise-detecting method would rightfully flag. Synthetic datasets with explicit noise labels, typically uniform background points around Gaussian or arbitrarily shaped clusters, are used instead (Table 6). Real-world datasets combining cluster structure with GT noise are rare, and higher-dimensional synthetic ones almost equally so; the 100-dimensional Synth_low and Synth_high datasets of Jahn et al. (2024) are a recent exception. Outlier detection benchmarks such as ODDS (Rayana, 2016) do not fill this gap (Section G): they usually relabel entire minority classes as outliers, which discards the cluster structure and yields compact microclusters rather than scattered noise.

2.2 External CVIs and noise handling strategies

Internal measures assess a clustering without GT labels, and noise-aware ones exist, notably DBCV (Moulavi et al., 2014) and DISCO (Beer et al., 2026); we assume GT labels and focus on external evaluation. Pairwise-counting CVIs check for every pair of points whether both labelings put it into the same cluster. Examples are the Rand Index (RI) (Rand, 1971), Fowlkes–Mallows (FM) (Fowlkes & Mallows, 1983), F1 (Sundheim, 1992), and Jaccard (J) (Jaccard, 1901); the Adjusted Rand Index (ARI) (Hubert & Arabie, 1985) corrects RI for chance agreement, so that random labelings score 0 in expectation regardless of their number of clusters. Entropy-based CVIs are Normalized Mutual Information (NMI) (Strehl & Ghosh, 2002) and Adjusted Mutual Information (AMI) (Vinh et al., 2009). Purity (Amigó et al., 2009) is the fraction of points that belong to the majority GT class of their predicted cluster. V-measure (Rosenberg & Hirschberg, 2007) and BCubed (Amigó et al., 2009) are not used by any surveyed paper that evaluates on datasets with GT noise (Table 2); V-measure shares NMI's biases and BCubed has no established convention for noise labels, so we exclude both. Section N lists all formulas. RI, F1, Jaccard, Purity, and, more weakly, NMI are biased toward labelings with more clusters (Section 3.1); the chance-corrected ARI and AMI are not, which partly explains the widespread adoption of ARI.

We identified four strategies in the literature for computing an external CVI M that is not defined for noise labels; they have also been discussed for internal evaluation (Moulavi et al., 2014; Beer et al., 2026), and Table 2 lists which publication uses which.

Noise cluster (M_c): all noise labels form one additional cluster. This is the most frequent strategy in our survey, likely because it is the automatic behavior of standard implementations such as scikit-learn (Pedregosa et al., 2011), but also the least meaningful: noise objects do not share the properties of a cluster, and grouping them can produce misleadingly high scores, as in Figure 1, where assigning a full cluster to noise yields a near-perfect score. Conversely, a prediction that groups exactly the GT noise objects into one cluster receives a perfect score under M_c, although it detects no noise at all (Figure 3d). Both failures also occur on real data (Section 3.7).

Singletons (M_s): each point labeled as noise in the GT or the prediction becomes a cluster of size one. This avoids the artificial grouping of M_c but adds many clusters, which interacts with the bias of some CVIs toward higher cluster counts. **Filtering** (M_f): all points labeled as noise in either labeling are removed before computing M_f, so the score reflects only the partition of the points both labelings consider data, and noise detection is ignored entirely. **Penalty** (M_p): the filtered score is multiplied by the fraction of retained points, $M_p = M_f \cdot \frac{\#\text{non-filtered}}{n}$, a crude sensitivity to noise labeling rather than an evaluation of noise detection quality.

Beyond these, Smith et al. (2019) propose the LNP-I framework (Labeled Noise Points-Inclusive), showing that ignoring noise reduces the predictive value of external metrics and adapting Purity, NMI, and the Sum of Squared Errors (SSE); SSE needs cluster centers and is excluded here. $\mathrm{Purity}_{\text{LNP-I}}$

filters noise and thus equals Purity_f; $\mathrm{NMI}_{\text{LNP-I}}$ resembles NMI_s but also assigns a unique label to GT noise points that the prediction puts into a cluster[2].

3 Critical Problems in Evaluating Clusterings with Noise Labels

Noise handling in external evaluation is inconsistent and often not described well enough to be reproducible (Section 2). We now show that this is not merely a question of convention: many common strategies produce misleading or uninformative scores.

Notation X denotes the GT labeling and Y the predicted labeling of an n-point dataset; each assigns every point a cluster label or the noise label. $\#data$ is the number of points labeled as cluster objects by *both* X and Y, $\#noise$ the number labeled as noise by *either*, so $\#data + \#noise = n$. For a base CVI M, M_c, M_s, M_f, M_p denote the strategies of Section 2.2 and $M_{\text{LNP-I}}$ the variants of Smith et al. (2019); k_{GT} and k_{pred} are the numbers of clusters in X and Y.

We define four properties an external CVI should satisfy, each stated informally and then formally:

(P1) *Unbiasedness*: uniform random clusterings with any number of clusters should score similarly (bad). Formally, the expected score is 0 under two null models in which Y is drawn independently of X: the classical permutation null, which randomly permutes the cluster labels of Y for any k_{pred} (Hubert & Arabie, 1985; Vinh et al., 2009), and a *noise-aware* null, in which additionally each point's predicted noise flag is drawn independently of X (Proposition 1). The latter models a degenerate detector, e.g., DBSCAN at extreme ε, whose flags carry no information about the GT; it adds variation in the predicted noise prevalence, which the fixed-marginal permutation null conditions away.

(P2) *Supremum tightness*: a perfect score should be reachable on any dataset. Formally, if Y agrees with X on every noise flag and induces the same partition of the cluster objects (up to renaming), the score is 1 (Proposition 2).

(P3) *No spurious optima*: a clearly incorrect clustering should not receive a (near) perfect score. Formally, the converse of (P2): a score of 1 implies agreement on every noise flag and on the partition of the cluster objects (Proposition 3).

(P4) *Monotonicity*: the score should decrease as more points are assigned incorrect labels. Formally, for two predictions Y_1, Y_2 that flag the same GT cluster objects as noise, so that the common cluster-object subset is the same, if Y_2 agrees with X at least as well as Y_1 on both the noise flags and the partition of the common cluster objects, then Y_2 scores at least as high (Proposition 4). This specializes the perfect-oriented monotonicity of Gösgens et al. (2021) for CVIs without noise to steps that leave the common cluster-object subset untouched; steps that change it are tested empirically (Section 3.4).

Other criteria, such as runtime (Section 4), strict boundedness, or robustness to specific noise distributions, are orthogonal to whether a score reliably ranks clusterings, so we do not formalize them. We test the eight CVIs of Section 2.2 under all four strategies and the LNP-I variants (Table 3, summarized in Table 1). Each experiment starts from the GT labeling of a synthetic dataset (Table 7) and degrades it by reassigning points to incorrect labels one by one, recomputing all scores after each step; the schemes are listed in Table 8. We evaluate each property in turn and then introduce our proposed measures in Section 3.5.

3.1 (P1) Unbiasedness: Independence from the Number of Clusters

An unbiased CVI does not favor clusterings with a particular number of clusters k_{pred} that is independent of the GT. We test this with random labelings of increasing k_{pred} on noise-free data, i.e., under the classical permutation null (Figure 2, $n = 1500$, $k_{GT} = 20$). RI, F1, and Jaccard increase rapidly with k_{pred}, because splitting the data into more groups trivially increases the number of agreeing pairs; NMI and Purity show a weaker trend in the same direction, well documented for NMI (Romano et al., 2014; Amelio & Pizzuti, 2017); FM shows the inverse bias, favoring fewer

[2]The paper describes $\mathrm{NMI}_{\text{LNP-I}}$ inconsistently: its text removes GT noise points assigned to clusters ($C_0 \setminus N$) from each predicted cluster, whereas its Equation 10 removes N, which would give $\mathrm{NMI}_{\text{LNP-I}} = \mathrm{NMI}_s$. We implement the textual definition, which matches the stated motivation of penalizing noise misassigned to clusters.

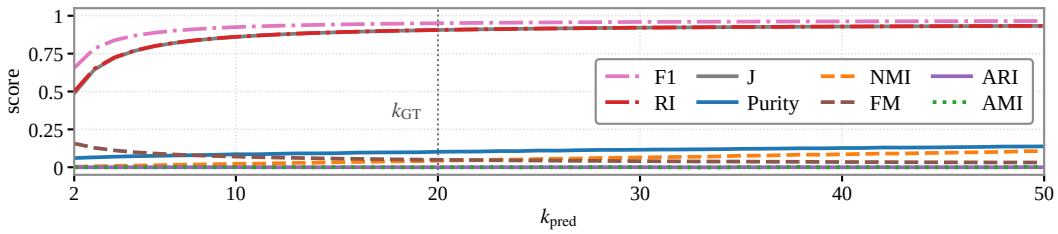


Figure 2: **(P1) Unbiasedness on random labelings with increasing k_{pred} ($k_{\text{GT}} = 20$, $n = 1500$).** RI, F1, and Jaccard saturate after very few clusters, NMI and Purity rise gradually; only ARI and AMI stay near 0.

clusters. Only ARI and AMI remain near zero for all k_{pred}, as their chance correction accounts for the number of clusters. This bias matters in combination with noise handling: under M_s, every noise point adds a cluster, so the biased measures rise as more points are labeled as noise, the failure visible in Figure 3c; in the parameter search of Section 3.6, $\text{Purity}_{\text{LNP-I}}$, J_s, and F1_s reward a clustering with many microclusters (Figure 4d). Under the noise-aware null, ARI_s, AMI_s, and the proposed measures have expected scores within ± 0.015 of zero already for $n = 20$ and within ± 0.004 for $n = 1000$ across noise ratios from 0 to 0.95 (Proposition 1, simulations in Section B).

3.2 (P2) Supremum bound: Perfect scores should be reachable

A CVI should assign its optimum (typically 1) to a prediction that matches the GT; otherwise its upper bound is an artifact of the strategy. We verify this by scoring the GT labeling against itself on all datasets of Table 7. M_p violates the property by construction: $M_p = M_f \cdot \frac{\#\text{non-filtered}}{n}$ and any dataset with noise has $\#\text{non-filtered} < n$, so even the correct clustering scores below 1, the maximum depends on the noise ratio, which makes scores incomparable across datasets, and the strategy can never certify a clustering as correct (Figure 3b).

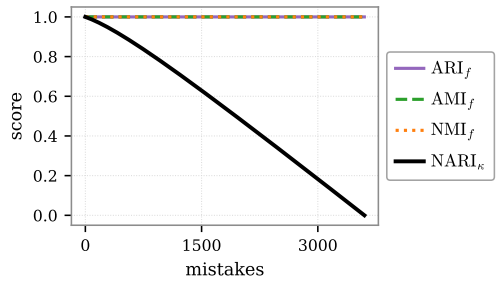
3.3 (P3) No spurious optima: Wrong clusterings should not have perfect scores

M_f violates this property directly: the noise labeling is never assessed, so a clustering that partitions the non-noise points correctly but misclassifies every noise point (or vice versa) scores 1 (Figure 3a); in the extreme, labeling everything as noise except two correctly clustered points scores 1. M_c violates it as well: a prediction that groups exactly the GT noise objects into one cluster coincides with the GT once both noise labels are treated as clusters, so it scores 1 although not a single noise object was detected (Figure 3d). Unadjusted CVIs under M_s fail differently: on noisyGaussian, labeling every point as noise still yields $\text{NMI}_s = 0.96$ and $\text{RI}_s = 0.996$ (Table 11), because the singleton encoding inflates k_{pred} (Section 3.1). Section H gives further examples.

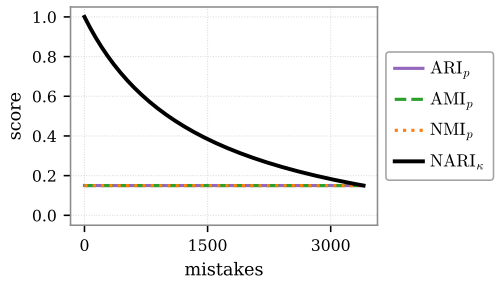
3.4 (P4) Monotonicity: The worse the clustering, the lower the score

We test monotonicity along the seven degradation schemes of Table 8 on all datasets of Table 7; Figure 3 shows the violations. M_f does not decrease when cluster points are relabeled as noise, because they are filtered (Figure 3a); under M_s, Purity_s does not react at all and the other unadjusted CVIs only marginally (Figure 3c); and under M_c, misassigning noise points to one cluster *increases* the score, up to a spurious optimum without a single correctly identified noise point (Figure 3d). These schemes change the noise flags. Conditioning on the two components in the formal definition of (P4) is what makes the property provable for chance-corrected measures at all: whenever a point moves between noise and cluster labels, the marginals of the chance correction shift, and a further wrong relabeling can then increase every chance-corrected measure we consider (counterexamples in Section B.1). This concerns arbitrary single-point changes; under the perfect splits and merges of Gösgens et al. (2021), ARI and AMI are monotone. For flag-changing degradations, monotonicity can therefore only be established empirically, for the baselines as well as for our measures (Figure 5); Table 1 marks the difference.

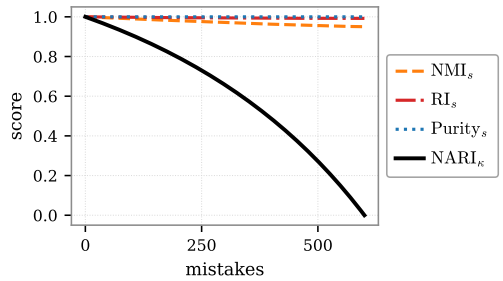
Figure 3: **Violations of (P2), (P3), and (P4).** Each panel shows three violating combinations of one strategy and NARI_κ (black) as a property-satisfying reference; full curves of all eight base CVIs in Figure 6.



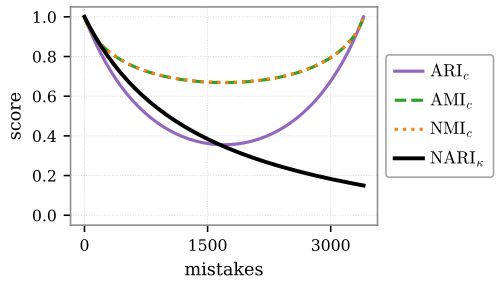
(a) M_f does not decrease as cluster points are relabeled as noise (P3, P4). Random data to noise, 10% GT noise.



(b) M_p does not decrease as GT noise is assigned to clusters, and its optimum is unreachable (P2, P4). Noise to data, 85% GT noise.



(c) M_s with unadjusted CVIs barely reacts to points misassigned to noise (P3), Purity_s not at all. Random data to noise, 85% GT noise.



(d) M_c increases as noise points are wrongly assigned to one cluster (P3, P4). Noise to one cluster, 85% GT noise.

3.5 Novel suggestions for external noise-aware CVIs

The strategies of Section 2.2 force noise labels into CVIs that were not designed for them. Instead, we evaluate clustering quality and noise identification separately and combine both into one score: (i) clustering quality is measured by the ARI on the points that both labelings consider cluster objects; we use ARI because it is the de-facto standard (Table 2) and one of only two unbiased options we test (Section 3.1), the other, AMI, being computationally more expensive (Figure 13). (ii) Noise identification is treated as a binary classification problem (noise vs. cluster object) and evaluated with a chance-corrected agreement measure that accounts for class imbalance, since the proportion of noise varies widely: Cohen's kappa (κ) (Cohen, 1960), the standard inter-rater agreement statistic, or the Matthews–Pearson phi coefficient (φ) (Matthews, 1975), the Pearson correlation of the binary indicators. Both are symmetric and equal 1 exactly at perfect agreement.[3] They capture different intuitions of agreement: κ measures chance-corrected concordance, normalized by the agreement expected from the observed marginal frequencies, whereas φ is the Pearson correlation of the binary noise indicators, normalized by the geometric mean of the marginal products. This yields two measures:

$$\mathrm{NARI}_\kappa = \frac{\#data \cdot \mathrm{ARI} + \#noise \cdot \kappa}{n} \quad (1a) \qquad \mathrm{NARI}_\varphi = \frac{\#data \cdot \mathrm{ARI} + \#noise \cdot \varphi}{n} \quad (1b)$$

where $\#data$ is the number of points labeled as cluster objects by *both* labelings, $\#noise$ the number labeled as noise by *either*, and n the dataset size. The weights match each component's influence to its prevalence: ARI dominates when noise is rare, κ resp. φ when noise dominates.

[3] F1 was excluded as asymmetric and accuracy as uncorrected for class imbalance. Degenerate cases are resolved by convention: identical constant noise indicators give $\kappa = \varphi = 1$; a coefficient undefined because of a zero marginal while the indicators differ is set to 0. With these conventions, Propositions 2 and 3 hold without exception.

Why the weights depend on both labelings This dependence is required for (P3). With GT-only weights, a dataset without GT noise gives the noise component weight 0, so a prediction that labels 90% of the points as noise while preserving the partition on the rest scores 1; with prediction-only weights, on a dataset with 85% GT noise, a prediction without any noise labels that preserves the GT partition on the cluster objects scores 1. Only the symmetric definition routes every disagreement about noise into the κ/φ component, so that missed GT noise and spurious predicted noise carry the same penalty, which is what makes Proposition 3 hold in both directions (ablation in Section C). The weights also have a direct reading: $\#data/n$ is the fraction of points on which both labelings agree to compare cluster structure, $\#noise/n$ the fraction on which noise detection is at stake. Cluster points wrongly predicted as noise are thus re-routed rather than discounted: each leaves the ARI subset and enters the noise-detection problem as a disagreement. With five equal clusters and 10% GT noise ($n = 1000$), moving $m = 100$, 400, and 800 of the 900 cluster points to noise yields $\mathrm{NARI}_\kappa = 0.92, 0.60, 0.12$, strictly decreasing and comparable to ARI_s (0.86, 0.43, 0.02; Table 4). The two measures coincide when noise and cluster objects are balanced and diverge under imbalance: κ becomes conservative when one class dominates (the majority-baseline prediction is "free" under chance correction), whereas φ amplifies the minority class. In our experiments both track each other closely; NARI_φ is more sensitive to mistakes on the smaller class (a few missed cluster points on a high-noise dataset, or a few missed noise points on a low-noise one), so we recommend NARI_φ when those mistakes matter and NARI_κ as the conservative default. By construction, both measures avoid the failure modes above: noise identification enters explicitly through κ or φ, and unbiasedness is inherited from ARI and from the chance-corrected κ and φ (Proposition 1). Section B proves (P2)–(P4) for both and (P1) asymptotically (Propositions 1 to 4), and Section E shows empirically that they behave as intended on all datasets and degradation schemes (Figure 5). For the strongest baselines, ARI_s and AMI_s, no proof of (P4) exists: under the componentwise conditions of Proposition 4 (same noise flags, ordered restricted ARI), they violate monotonicity in 12.2% and 14.1% of randomly sampled comparable pairs (Section B.1), i.e., improving the cluster partition with the noise flags held fixed can decrease ARI_s or AMI_s; NARI_κ and NARI_φ provably cannot behave this way.

3.6 Parameter search on benchmark datasets.

The previous experiments degrade GT labelings synthetically. To show that the problems affect practical use, we grid-search DBSCAN parameters on benchmark datasets ($\mathrm{minPts} = 2, \ldots, 16$; 15 values of ε on an exponential scale around the threshold point of the sorted k-dist graph (Ester et al., 1996)) and let each CVI select the clustering that maximizes it. A flawed strategy guides the search toward a suboptimal or degenerate clustering.

On CLUTO-t4.8k (Figure 4), NARI_κ (a) and ARI_s (b) select clusterings that identify all six clusters and mislabel only a few noise points; ARI_f (c) selects three microclusters with most of the data labeled as noise, because the few remaining points happen to be clustered correctly, and $\mathrm{Purity}_{\text{LNP-I}}$ (d) selects many microclusters, consistent with Purity's bias toward finer splits (Section 3.1).

Section I shows the same on denseAndSparse, where ARI_c scores a clustering that merges the sparse cluster into noise at 0.97 (cf. Figure 1), and on noisyGaussian, where M_f, M_p, and $\mathrm{NMI}_{\text{LNP-I}}$ select clusterings with spurious clusters in the GT noise while ARI_s selects a realistic one. Flawed CVI–strategy combinations thus select objectively worse clusterings. These failures are properties of the combinations, not of DBSCAN: they recur with HDBSCAN (Section J).

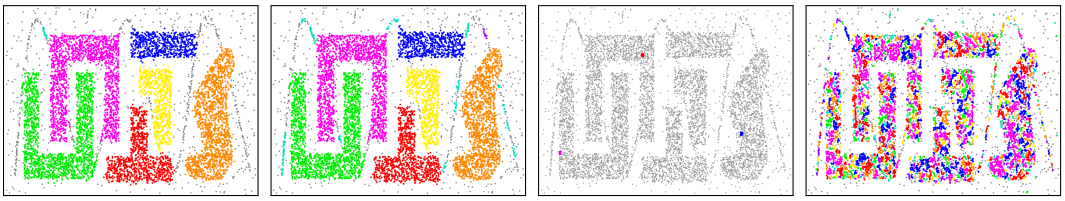
3.7 Real-world impact

The selection failures are not an artifact of synthetic noise. Real-world datasets with reliable, diffuse GT noise are scarce (Section 2.1), so we obtain real, non-injected noise labels in two ways (Section M): first, minority classes held out as GT noise ($\approx 5\%$) on optdigits, pendigits, MiceProtein, and HAR; second, a foundation model (CLIP, ViT-B/32) annotating the CIFAR-10 test images it cannot assign to any class with sufficient confidence as GT noise, at 5% and 10% noise. In both settings, we run full DBSCAN and HDBSCAN parameter sweeps and let every CVI–strategy combination select its argmax configuration. The failure modes reproduce on all six datasets: ARI_f selects degenerate clusterings with 99–100% of the points labeled as noise and $\mathrm{Purity}_{\text{LNP-I}}$ selects 98–100%-noise microcluster solutions, both at a perfect self-score of 1.0, whereas ARI_s, AMI_s, NARI_κ, and NARI_φ select structurally sensible clusterings (e.g., $k_{pred} = 8 = k_{GT}$ on optdigits). Notably, the

CLIP-flagged images are not geometric outliers in embedding space, so the two settings differ in the origin of the noise but not in the behavior of the measures: an evaluation measure only sees the two label vectors, and its behavior is driven by their combinatorial structure, not by the mechanism that produced the noise.

The choice of strategy also changes which method a comparison declares best. In a case study of three noise-detecting methods, SHADE (Beer et al., 2024), DBSCAN, and HDBSCAN, on real and high-dimensional synthetic benchmarks (Section L), ARI_c scores SHADE and HDBSCAN both at 1.000 on Synth_high although SHADE flags only 0.3% of the points as noise against the true 10%; NARI_κ ranks it below DBSCAN (0.896 vs. 0.977). On the HTRU2 pulsar benchmark, the cross-method ranking inverts completely: ARI_c prefers DBSCAN and HDBSCAN configurations that put 86% of the data into one cluster plus a handful of microclusters, whereas NARI_κ ranks SHADE first (0.671 vs. 0.545 and 0.532), whose two main clusters approximately recover the binary GT. Published rankings of noise-detecting methods thus depend on an evaluation choice that most papers do not even report.

Figure 4: **Flawed CVI–strategy combinations steer DBSCAN parameter search toward degenerate clusterings** (CLUTO-t4.8k; subcaptions give the selecting measure and its ε, minPts).



(a) NARI_κ: six clusters (8.87, 16) (b) ARI_s: six clusters (6.76, 8) (c) ARI_f: mostly noise (3, 11) (d) $\mathrm{Purity}_{\text{LNP-I}}$: microclusters (3, 2)

4 Discussion

Overview Table 1 summarizes which combinations satisfy the four properties; the patterns are consistent across base CVIs. M_c and M_f admit spurious optima (grouped noise can coincidentally align with GT structure; filtered noise is never evaluated), M_p caps scores below 1 whenever noise is present, and M_s is the best-behaved existing strategy, but only with a base CVI that is itself unbiased: NMI, RI, F1, Jaccard, and Purity are sensitive to the number of clusters, which the many singletons of M_s amplify. Unbiasedness is thus inherited from the base CVI, not from the strategy, and among the base CVIs only ARI and AMI have it. Only four combinations satisfy all four properties: AMI_s, ARI_s, NARI_κ, and NARI_φ. AMI_s has a practical drawback: its chance-correction term requires $\Theta(k_{pred} \cdot k_{GT})$ log-gamma evaluations, which makes AMI roughly an order of magnitude slower than the other three across $n = 10^3$ to 10^6 (Figure 13), an overhead that matters in parameter searches.

Why not simply ARI_s or AMI_s? ARI_s and AMI_s are the strongest existing baselines, which is precisely why we test them so prominently. NARI_κ and NARI_φ satisfy the same four properties and are computationally lightweight, and they offer four advantages over the singleton encoding. (1) *Provable properties.* For NARI_κ and NARI_φ, (P2), (P3), and the componentwise form of (P4) are proved and (P1) is established under both null models (Section B); for ARI_s and AMI_s, (P4) rests on empirical observation only, and the componentwise conditions of Proposition 4 are measurably violated by both (Section B.1). (2) *Decomposability.* NARI_κ is a weighted mean of an ARI term (cluster quality) and a κ term (noise detection as a binary problem); both components are reported separately by our library, so a practitioner sees *why* a clustering scored badly. ARI_s conflates both error types inside one partition score over an artificial encoding and admits no such decomposition. (3) *Sensitivity to failed noise detection.* Treating each noise point as a singleton is an encoding trick, and pair counting gives noise errors very little weight. For example, on the noisyGaussian dataset, a labeling that keeps the three Gaussian clusters perfectly intact but groups 30% of the GT noise points into spurious five-point microclusters still receives $\mathrm{ARI}_s = 0.98$: the microclusters contribute almost no pairs, so pair counting barely registers them, and even AMI_s only drops to 0.72. In contrast, $\mathrm{NARI}_\kappa = 0.39$ and $\mathrm{NARI}_\varphi = 0.49$ expose the failed noise detection through their explicit noise component ($\kappa = 0.32$). This is exactly the regime where the measures disagree, and the answer of the

Table 1: Properties of the chance-corrected CVI–strategy combinations and the proposed measures (all 36 combinations in Table 3). ✓: proved (for NARI_κ and NARI_φ by Propositions 1 to 4, componentwise for (P4); for the baselines, (P1)–(P3) follow from the chance correction and the identity of the encoded labelings). (✓): established empirically only (Figure 5); the componentwise conditions of Proposition 4 are violated by ARI_s and AMI_s (Section B.1). ✗: violated.

Measure	*unbiased*	*supremum bound*	*non-spurious optimum*	*monotonic*
ARI_c, AMI_c	✓	✓	✗	✗
ARI_s, AMI_s	✓	✓	✓	(✓)
ARI_f, AMI_f	✓	✓	✗	✗
ARI_p, AMI_p	✓	✗	✓	✗
NARI_κ, NARI_φ	✓	✓	✓	✓

noise-aware measures is the informative one. The singleton encoding moreover cannot distinguish a one-point predicted cluster from predicted noise, which is why the expected value of ARI_s under the noise-aware null depends on how such points are encoded, whereas NARI_κ and NARI_φ are invariant to this choice (Section B). (4) *Runtime versus* AMI_s, as discussed above: κ and φ are $O(n)$. This makes the proposed measures particularly suitable when noise detection quality is an explicit concern.

Limitations Our four properties are necessary, not sufficient: a measure could satisfy all four and still fail on labelings constructed to expose other weaknesses. For instance, NARI_κ and NARI_φ do not distinguish whether GT noise points wrongly assigned to clusters contaminate an existing cluster or form a separate spurious one, since both cases enter only the noise component. The properties are also stated relative to choices, (P1) to a null model and (P4) to a comparison order; the axiomatic framework of Gösgens et al. (2021) for CVIs without noise offers a complementary lens, and its perfect-oriented monotonicity for steps that change the common cluster-object subset is the natural generalization of Proposition 4 to explore. Since single wrong relabelings can increase every chance-corrected measure we tested (Section B.1), such a generalization has to characterize when flag-correcting steps are provably monotone, which we do not claim here. The synthetic degradation schemes are controlled tests and do not exhaust the failure modes of real noise-detecting clusterings. Finally, real-world datasets with cluster structure and GT noise remain rare (Section 2.1), which limits any CVI, including ours. Our two real-data protocols are proxies: held-out classes form compact microclusters rather than diffuse noise, and foundation-model annotations flag semantic ambiguity rather than low density. Neither protocol models how noise arises in the wild; both supply the combinatorial structure that determines a measure's behavior, namely noise fraction, cluster-size distribution, and the overlap of predicted and GT noise, on real feature distributions and real cluster geometry.

Call to Action External evaluation of clusterings with noise is currently neither uniform nor scientifically adequate: many publications do not state how they handle noise, or use strategies that violate basic desiderata. We call on the community to (i) report the noise-handling strategy explicitly and choose a combination that satisfies the properties in Table 1, and (ii) publish benchmark datasets with both GT cluster structure and noise labels, ideally higher-dimensional and real-world. To support both, we will release the experiments and CVI implementations of this paper as a pip-installable benchmark suite upon acceptance, enabling consistent computation and reporting of property-satisfying noise-aware CVIs with a single dependency.

5 Conclusion

External evaluation of clusterings with noise has developed ad hoc, with strategies chosen for convenience rather than soundness. Most existing methodologies violate at least one of four basic properties (unbiasedness, supremum tightness, no spurious optima, monotonicity), and the noise-handling strategy changes both which clustering a parameter search selects and which method a comparison declares best. Our measures NARI_κ and NARI_φ satisfy all four properties while explicitly evaluating noise identification. Adopting these or any other property-satisfying measure, and stating the choice, is a small change that would substantially improve the rigor of the field.

AI USE STATEMENT

In this work, we used generative AI tools (large language models) to implement standard methods, namely for writing and refactoring experiment and plotting code that calls existing library implementations of the compared CVIs and clustering algorithms; all such code was verified and tested for correctness by the authors, and every reported number was reproduced from the released scripts. We have not used generative AI tools to generate synthetic datasets, to develop the theoretical framework or the desirable properties, to formulate or prove the mathematical claims, to propose or refine hypotheses, to design the experiments, or to interpret the results; the remaining tasks with required disclosure (translation, dataset cleaning and reformatting, qualitative or thematic data analysis) are not applicable to this work. Additionally, we used generative AI tools to edit the text for readability. We have reviewed all AI-assisted work, checked every AI-edited statement against our own results and the cited literature.

ETHICS STATEMENT

This work studies evaluation methodology for clustering algorithms. It involves no human subjects, no personal data, and no sensitive attributes; all datasets are public benchmark datasets used under their respective licenses (Sections D and M). We do not foresee negative societal consequences beyond those inherent to more reliable evaluation; if anything, exposing evaluation strategies that reward degenerate clusterings should reduce misleading claims in the literature. The literature survey (Table 2) reports how published work handles noise in evaluation, not the validity of that work's scientific claims.

REPRODUCIBILITY STATEMENT

All experiments can be reproduced from the anonymized repository linked in Section 1. The controlled degradation experiments are specified by the datasets in Table 7 and the schemes in Table 8; the DBSCAN and HDBSCAN parameter grids are given in Sections 3.6 and J; the real-data protocols (held-out minority classes and foundation-model-annotated noise) are fully described in Section M, including dataset sources, preprocessing, and the exact grids. The repository contains per-entry reproduction scripts for Table 3, the simulation scripts for the null model of Proposition 1 and the monotonicity search of Section B.1, and the weighting ablation of Section C. All proofs are given in full in Section B, with all assumptions stated in the propositions. The CVI implementations are packaged as a Python library that we will release as a pip-installable package upon acceptance.

REFERENCES

Mohamed Abbas, Adel El-Zoghabi, and Amin Shoukry. DenMune: Density peak based clustering using mutual nearest neighbors. *Pattern Recognition*, 109:107589, 2021.

Tiba Zaki Abdulhameed, Suhad A Yousif, Venus W Samawi, and Hasnaa Imad Al-Shaikhli. Ss-dbscan: Semi-supervised density-based spatial clustering of applications with noise for meaningful clustering in diverse density data. *IEEE Access*, 12:131507–131520, 2024.

Naoki Abe, Bianca Zadrozny, and John Langford. Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 504–509, 2006.

Stefan Aeberhard and M. Forina. Wine. UCI Machine Learning Repository, 1992. DOI: https://doi.org/10.24432/C5PC7J.

E. Alpaydin and Fevzi. Alimoglu. Pen-Based Recognition of Handwritten Digits. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5MG6K.

E. Alpaydin and C. Kaynak. Optical Recognition of Handwritten Digits. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C50P49.

Alessia Amelio and Clara Pizzuti. Correction for closeness: Adjusting normalized mutual information measure for clustering comparison. *Computational Intelligence*, 33(3):579–601, 2017.

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486, 2009.

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *Proceedings of the 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pp. 437–442, 2013.

Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. OPTICS: Ordering points to identify the clustering structure. *ACM SIGMOD Record*, 28(2):49–60, 1999.

Tharindu R Bandaragoda, Kai Ming Ting, David Albrecht, Fei Tony Liu, and Jonathan R Wells. Efficient anomaly detection by isolation using nearest neighbour ensemble. In *2014 IEEE International conference on data mining workshop*, pp. 698–705. IEEE, 2014.

Jeffrey D. Banfield and Adrian E. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821, 1993.

Anna Beer, Pascal Weber, Lukas Miklautz, Collin Leiber, Walid Durani, Christian Böhm, and Claudia Plant. Shade: Deep density-based clustering. In *2024 IEEE International Conference on Data Mining (ICDM)*, pp. 675–680. IEEE, 2024.

Anna Beer, Lena Krieger, Pascal Weber, Martin Ritzert, Ira Assent, and Claudia Plant. Internal evaluation of density-based clusterings with noise. In *The Fourteenth International Conference on Learning Representations*, 2026.

Derya Birant and Alp Kut. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007. doi: 10.1016/j.datak.2006.01.013.

Christian Böhm, Claudia Plant, Junming Shao, and Qinli Yang. Clustering by synchronization. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 583–592, 2010.

Avory Bryant and Krzysztof Cios. RNN-DBSCAN: A density-based clustering algorithm using reverse nearest neighbor density estimates. *IEEE Transactions on Knowledge and Data Engineering*, 30(6):1109–1121, 2018.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 160–172. Springer, 2013.

Ricardo JGB Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1):1–51, 2015.

Guilherme O Campos, Arthur Zimek, Jörg Sander, Ricardo JGB Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30 (4):891–927, 2016.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009. doi: 10.1145/1541880.1541882.

B. B. Chaudhuri and P. R. Bhowmik. An approach of clustering data with noisy or imprecise feature measurement. *Pattern Recognition Letters*, 19(14):1307–1317, 1998. doi: 10.1016/S0167-865 5(98)00112-3.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

Li Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. ISSN 10535888. doi: 10.1109/MSP.2012.221 1477. URL `https://ieeexplore.ieee.org/document/6296535`.

Maria d'Errico, Elena Facco, Alessandro Laio, and Alex Rodriguez. Automatic topography of high-dimensional data sets by non-parametric density peak clustering. *Information Sciences*, 560: 476–492, 2021.

Andrew Draganov, Pascal Weber, Rasmus Skibdahl Melanchton Jørgensen, Anna Beer, Claudia Plant, and Ira Assent. Ultrametric cluster hierarchies: I want 'em all! In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Walid Durani, Dominik Mautz, Claudia Plant, and Christian Böhm. DBHD: Density-based clustering for highly varying density. In *2022 IEEE International Conference on Data Mining (ICDM)*, pp. 921–926. IEEE, 2022.

Walid Durani, Dominik Mautz, Claudia Plant, and Christian Böhm. DMDHC: Discovery of Multi-Density hierarchical cluster structures. In *Proceedings of the 2025 SIAM International Conference on Data Mining (SDM)*, pp. 261–269. SIAM, 2025.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 226–231, 1996.

R. A. Fisher. Iris. UCI Machine Learning Repository, 1936. DOI: https://doi.org/10.24432/C56C76.

Edward B Fowlkes and Colin L Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

B. German. Glass Identification. UCI Machine Learning Repository, 1987. DOI: https://doi.org/10.24432/C5WW2P.

Jan-Mark Geusebroek, Gertjan J Burghouts, and Arnold WM Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005.

Martijn M. Gösgens, Alexey Tikhonov, and Liudmila Prokhorenkova. Systematic analysis of cluster similarity indices: How to validate validation measures. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3799–3808. PMLR, 2021.

Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: An efficient clustering algorithm for large databases. *ACM SIGMOD Record*, 27(2):73–84, 1998.

Clara Higuera, Katheleen J. Gardiner, and Krzysztof J. Cios. Self-organizing feature maps identify proteins critical to learning in a mouse model of Down syndrome. *PLoS ONE*, 10(6):e0129126, 2015.

Lihua Hu, Hongkai Liu, Jifu Zhang, and Aiqin Liu. KR-DBSCAN: A density-based clustering algorithm based on reverse nearest neighbor and influence space. *Expert Systems with Applications*, 186:115763, 2021.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994. doi: 10.1109/34.291440.

Félix Iglesias, Tanja Zseby, and Arthur Zimek. Clustering refinement. *International Journal of Data Science and Analytics*, 12(4):333–353, 2021.

Paul Jaccard. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bull Soc Vaudoise Sci Nat*, 37:241–272, 1901.

Philipp Jahn, Christian MM Frey, Anna Beer, Collin Leiber, and Thomas Seidl. Data with density-based clusters: A generator for systematic evaluation of clustering algorithms. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 3–21. Springer, 2024.

George Karypis. CLUTO – a clustering toolkit. Technical Report 02-017, University of Minnesota, Department of Computer Science, 2002.

George Karypis, Eui-Hong Han, and Vipin Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, 1999.

Uranus Kazemi and Seyfollah Soleimani. A new approach data processing: density-based spatial clustering of applications with noise (DBSCAN) clustering using game-theory. *Soft Computing*, 29(3):1331–1346, 2025.

Fabian Keller, Emmanuel Muller, and Klemens Bohm. HiCS: High contrast subspaces for density-based outlier ranking. In *2012 IEEE 28th international conference on data engineering*, pp. 1037–1048. IEEE, 2012.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. UCI machine learning repository, 2023. URL `http://archive.ics.uci.edu`. Last accessed: 2026-05-05.

Lena Krieger, Anna Beer, Pernille Matthews, Anneka Myrup Thiesson, and Ira Assent. Fairden: Fair density-based clustering. In *The Thirteenth International Conference on Learning Representations*, 2025.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Hongfu Liu, Jun Li, Yue Wu, and Yun Fu. Clustering with outlier removal. *IEEE Transactions on Knowledge and Data Engineering*, 33(6):2369–2379, 2021.

Robert J. Lyon. HTRU2. UCI Machine Learning Repository, 2015.

Brian W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451, 1975.

Samuel Maurus and Claudia Plant. Skinny-dip: clustering in a sea of noise. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1055–1064, 2016.

Barbora Micenková, Brian McWilliams, and Ira Assent. Learning outlier ensembles: The best of both worlds–supervised and unsupervised. In *Proceedings of the ACM SIGKDD 2014 Workshop on Outlier Detection and Description (ODD2)*, 2014.

Barbora Micenková, Joost van Beusekom, and Faisal Shafait. Stamp verification for automated document authentication. In *Computational Forensics (IWCF 2012 and 2014), LNCS 8915*, pp. 117–129. Springer, 2015.

Tom Mitchell. Twenty Newsgroups. UCI Machine Learning Repository, 1997. DOI: https://doi.org/10.24432/C5C323.

Davoud Moulavi, Pablo A Jaskowiak, Ricardo JGB Campello, Arthur Zimek, and Jörg Sander. Density-based clustering validation. In *Proceedings of the 2014 SIAM international conference on data mining*, pp. 839–847. SIAM, 2014.

Sameer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia object image library (COIL-20). Technical Report CUCS-005-96, Department of Computer Science, Columbia University, February 1996a.

Sameer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia object image library (COIL-100). Technical Report CUCS-006-96, Department of Computer Science, Columbia University, February 1996b.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Roberto Pirrone, Vincenzo Cannella, Sergio Monteleone, and Gabriella Giordano. Linear density-based clustering with a discrete density model. *arXiv preprint arXiv:1807.08158*, 2018.

Yuan Qiu and Ke Yi. Approximate DBSCAN under differential privacy. *Proceedings of the ACM on Management of Data*, 3(3):1–24, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

Shebuti Rayana. ODDS library, 2016. URL `https://shebuti.com/outlier-detection-datasets-odds/`. Last accessed: 2026-05-05.

Simone Romano, James Bailey, Vinh Nguyen, and Karin Verspoor. Standardized mutual information for clustering comparisons: one step further in adjustment for chance. In *International conference on machine learning*, pp. 1143–1151. PMLR, 2014.

Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pp. 410–420, 2007.

Saket Sathe and Charu Aggarwal. LODES: Local density meets spectral outlier detection. In *Proceedings of the 2016 SIAM international conference on data mining*, pp. 171–179. SIAM, 2016.

Yijing Shen, Wei Sun, and Ker-Chau Li. Dynamically weighted clustering with noise set. *Bioinformatics*, 26(3):341–347, 2010.

David Slate. Letter Recognition. UCI Machine Learning Repository, 1991. DOI: https://doi.org/10.24432/C5ZP40.

Rob Smith, Jebediah Rosen, and Dan Ventura. Adapting standard external clustering metrics for repetitive, noisy observations. In *2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 908–914. IEEE, 2019. doi: 10.1109/ICMLA.2019.00157.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(Dec):583–617, 2002.

Beth M. Sundheim. Overview of the fourth Message Understanding Evaluation and Conference. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pp. 3–21, 1992.

Michael C Thrun. Distance-based clustering challenges for unbiased benchmarking studies. *Scientific Reports*, 11(1):18988, 2021.

A. Tramacere and C. Vecchio. γ-ray DBSCAN: a clustering algorithm applied to Fermi-LAT γ-ray data. I. detection performances with real and simulated data. *Astronomy & Astrophysics*, 549:A138, 2013. doi: 10.1051/0004-6361/201220133.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning*, pp. 1073–1080, 2009.

Limin Wang, Honghuan Wang, Xuming Han, and Wei Zhou. A novel adaptive density-based spatial clustering of application with noise based on bird swarm optimization algorithm. *Computer Communications*, 174:205–214, 2021.

Shimei Wang, Yun Liu, and Bo Shen. MDBSCAN: Multi-level density based spatial clustering of applications with noise. In *Proceedings of the 11th International Knowledge Management in Organizations Conference on The changing face of Knowledge Management Impacting Society*, pp. 1–5, 2016.

William Wolberg, Olvi Mangasarian, Nick Street, and W. Street. Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository, 1993. DOI: https://doi.org/10.24432/C5DW2B.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, volume 17, 2004.

Ke Zhang, Marcus Hutter, and Huidong Jin. A new local distance-based outlier detection approach for scattered real-world data. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 813–822. Springer, 2009.

A SURVEY OF EVALUATION STRATEGIES IN THE LITERATURE.

Table 2 provides the full overview of how recent publications handle noise in external evaluation, complementing the summary in Section 2.

Table 2: Survey of publications that evaluate or should evaluate clusterings with noise labels. Most of them present a new clustering algorithm (bold, second column) and compare to other similar methods with or without noise labeling using the external measures given in the third column. While papers in the upper part of the table do not compute CVIs on any dataset with noise labels, papers in the second part use datasets with ground truth noise labeling and apply different strategies given in the last column. Occurrence of ground-truth noise labels in datasets, author and year, performed clustering algorithms, evaluation measure or method, and noise handling strategy. Highlighted algorithms were presented in the papers themselves.

Author(s), year of publication, source	Algorithms	External Evaluation	Strategy
No datasets with noise labels			
Kazemi & Soleimani (2025)	**untitled method**, k-means, SOM, SCGPSC, DBSCAN	Purity, RI, ARI	-
Abdulhameed et al. (2024)	**SS-DBSCAN**, DBSCAN, HDBSCAN, RNN-DBSCAN, KR-DBSCAN, ISBDBSCAN, ISDBSCAN, RECORD	V, ARI	-
Hu et al. (2021)	**KR-DBSCAN**, RNN-DBSCAN, ISBDBSCAN, RECORD	NMI, ARI, V	-
Thrun (2021)	DBSCAN, HDBSCAN	F1	-
Wang et al. (2016)	**MDBSCAN**, DBSCAN	F1	-
Campello et al. (2015)	**HDBSCAN***, SS-DBSCAN, OPTICS, AUTO-HDS	ARI	-
Campello et al. (2013)	**HDBSCAN**, OPTICS, AUTO-HDS	ARI, F1	-
Evaluation on datasets with noise labels			
Durani et al. (2025)	**DMDHC**, OPTICS, HDBSCAN, PERCH	NMI	singleton noise
Krieger et al. (2025)	**FairDen**, DBSCAN, FairSC, Fairlet, Scalable Fair Clustering	ARI, NMI	filter, penalty
Qiu & Yi (2025)	**DP-DBSCAN**, DP-Kmeans, DBSCAN, k-means	ARI, AMI	noise cluster
Beer et al. (2024)	**SHADE**, DCN, DEC, IDEC, DipDECK, DipEncoder, DDC	ARI, NMI	filter, other[a]
Durani et al. (2022)	**DBHD**, DBSCAN, OPTICS, HDBSCAN, k-means, SpectralACL, Spectral, DPC, Mean-shift, SNN-DPC, LSDBC, DBADV	AMI, ARI	noise cluster
d'Errico et al. (2021)	**DPA**, Spectral, HDBSCAN, GMM, DP	NMI[b]	noise cluster, filter
Iglesias et al. (2021)	Minibatch k-means, Agglomerative hierarchical clustering, GMM, BIRCH, HDBSCAN, OPTICS	AMI	not specified
Wang et al. (2021)	**BSA-DBSCAN**, DBSCAN, AP, DP	F1, RI, Jaccard	not specified
Liu et al. (2021)	**COR**, k-means, k-means$--$	NMI, RI[c]	cluster
Maurus & Plant (2016)	**SkinnyDip**, DBSCAN, DipMeans, PgMeans, Sync, RIC, STSC, FOSSCLU	AMI	filter, noise cluster, other[d]
Moulavi et al. (2014)	DBSCAN, OPTICS, HDBSCAN	ARI	penalty[e]
Böhm et al. (2010)	**SynC**, k-means, Spectral, AP, Mean-shift, X-means, RIC, OCI	NMI, AMI, AVI	not specified
Shen et al. (2010)	**DWCN**, k-means, tight clustering, PW-Kmeans	RI	noise cluster, filter[f]
Ester et al. (1996)	**DBSCAN**, CLARANS	visual comparison	-

[a] For comparing to competitors that do not produce noise labels, each detected noise point is assigned to its closest cluster, before computing evaluation measures. [b] On top of NMI with the two different strategies, FP and FN rates are also reported regarding noise detection. [c] To evaluate noise identification, Jaccard index and F1-score are reported w.r.t. noise and data labels. [d] On real-world datasets without ground-truth noise, k-means is performed to assign noise points to clusters, for computing evaluation scores. [e] The penalty term approach is selected after ruling out multiple others, including singleton noise, for reasons related to internal evaluation. [f] A weighted average of RI with the two approaches is also reported.

B PROPERTIES OF THE PROPOSED CVIS.

The empirical classification of all 36 CVI–strategy combinations against the four properties is shown in Table 3; the proofs below establish the entries for NARI_κ and NARI_φ.

Table 3: Our criteria on evaluation metrics and noise handling strategies. Note, that Purity_f and $\mathrm{Purity}_{\text{LNP-I}}$ are equivalent. ✓: proved (componentwise for (P4), Proposition 4); (✓): established empirically only, on the seven degradation schemes and eight datasets of Figure 5 (no proof is available for the monotonicity of any baseline, and the componentwise conditions of Proposition 4 are violated by ARI_s and AMI_s, see Section B.1); ✗: violated.

Measure	*unbiased*	*perfect*	*no spurious optima*	*mono-tonic*
Purity_c	✗	✓	✗	✗
Purity_s	✗	✓	✗	✗
Purity_f	✗	✓	✗	✗
Purity_p	✗	✗	✓	✗
NMI_c	✗	✓	✗	✗
NMI_s	✗	✓	✗	(✓)
NMI_f	✗	✓	✗	✗
NMI_p	✗	✗	✓	✗
AMI_c	✓	✓	✗	✗
AMI_s	✓	✓	✓	(✓)
AMI_f	✓	✓	✗	✗
AMI_p	✓	✗	✓	✗
RI_c	✗	✓	✗	✗
RI_s	✗	✓	✗	(✓)
RI_f	✗	✓	✗	✗
RI_p	✗	✗	✓	✗
ARI_c	✓	✓	✗	✗
ARI_s	✓	✓	✓	(✓)
ARI_f	✓	✓	✗	✗
ARI_p	✓	✗	✓	✗
FM_c	✗	✓	✗	✗
FM_s	✗	✓	✓	(✓)
FM_f	✗	✓	✗	✗
FM_p	✗	✗	✓	✗
$\mathrm{F1}_c$	✗	✓	✗	✗
$\mathrm{F1}_s$	✗	✓	✗	(✓)
$\mathrm{F1}_f$	✗	✓	✗	✗
$\mathrm{F1}_p$	✗	✗	✓	✗
J_c	✗	✓	✗	✗
J_s	✗	✓	✗	(✓)
J_f	✗	✓	✗	✗
J_p	✗	✗	✓	✗
$\mathrm{Purity}_{\text{LNP-I}}$	✗	✓	✗	✗
$\mathrm{NMI}_{\text{LNP-I}}$	✗	✓	✗	(✓)
NARI_κ	✓	✓	✓	✓
NARI_φ	✓	✓	✓	✓

We now show formally that the proposed measures NARI_κ and NARI_φ defined in Section 3.5 satisfy the four properties from Section 3: unbiasedness via chance correction (exactly for the ARI component and asymptotically in n for the κ/φ component), supremum tightness and no spurious optima exactly, and monotonicity componentwise. For the degenerate binary cases of κ and φ we use the convention of Section 3.5: identical constant noise indicators give $\kappa = \varphi = 1$, and a coefficient that is undefined because of a zero marginal while the indicators differ is set to 0. Throughout, X denotes the ground-truth labeling on an n-point dataset and Y the predicted labeling; both assign each point either to a cluster or to a noise label. We write $\#data$ for the number of points labeled as

cluster objects by both X and Y, and $\#noise$ for the number of points labeled as noise by either X or Y. Note that $\#data + \#noise = n$ for every pair of labelings. By convention, when $\#data = 0$ we set $\#data \cdot \mathrm{ARI} := 0$, and likewise $\#noise \cdot \kappa := 0$ and $\#noise \cdot \varphi := 0$ when $\#noise = 0$, so that the components are well-defined in degenerate cases.

Proposition 1 (Chance correction under the noise-aware null)**.** *Fix a ground-truth labeling X with cluster-object set S_X and GT noise proportion $r_X = 1 - |S_X|/n$, and let Y be sampled independently of X by the following two-step procedure:*

(i) Noise flags. *For every point independently, Y labels the point as noise with probability r_Y, independently of X. Let S_Y be the resulting cluster-object set of Y and $S = S_X \cap S_Y$ the common cluster-object set.*

(ii) Cluster labels. *Conditional on S_Y, the cluster labels of Y on S are a uniformly random positional permutation of the cluster labels of X on S, i.e., the classical permutation null of the ARI (Hubert & Arabie, 1985) applied on the intersection, so that in particular the cluster structure of X is randomized and not merely renamed. The cluster labels of Y on $S_Y \setminus S$, i.e., on GT noise objects that Y does not flag, are drawn independently from the empirical distribution of the cluster labels of X on S_X.*

Then $\mathbb{E}[\mathrm{ARI}(X|_S, Y|_S) \mid S_Y] = 0$ for every n on the event that $|S| \geq 2$ and $X|_S$ contains at least two clusters (on the complementary event the restricted ARI is degenerate; its probability decays exponentially in n for $r_Y < 1$), and for nondegenerate noise proportions $r_X, r_Y \in (0, 1)$, $\mathbb{E}[\mathrm{NARI}_\kappa(X, Y)] \to 0$ and $\mathbb{E}[\mathrm{NARI}_\varphi(X, Y)] \to 0$ as $n \to \infty$.

Proof. Condition on the noise flags of Y, i.e., on S_Y, and hence on S, $\#data = |S|$, and $\#noise = n - |S|$. By (ii), the labels of Y on S are a uniform positional permutation of the labels of X on S, so the contingency table of $X|_S$ and $Y|_S$ has the fixed marginals of the classical permutation model, under which the expected Rand index equals its chance term and hence $\mathbb{E}[\mathrm{ARI}(X|_S, Y|_S) \mid S_Y] = 0$ (Hubert & Arabie, 1985). The first summand of NARI_κ therefore vanishes in expectation for every n. For the second summand, the noise indicators $\mathbf{1}[X = \text{noise}]$ and $\mathbf{1}[Y = \text{noise}]$ are independent by (i), with marginal probabilities r_X and r_Y. Cohen's $\kappa = (p_o - p_e)/(1 - p_e)$ and the φ coefficient $(p_{11} - \hat{r}_X \hat{r}_Y)/\sqrt{\hat{r}_X(1 - \hat{r}_X)\hat{r}_Y(1 - \hat{r}_Y)}$ are ratio estimators built from the observed agreement p_o, the observed joint noise frequency p_{11}, and plug-in estimates $\hat{r}_X, \hat{r}_Y$ of the marginals. Under independence their population values are 0, because the population agreement equals the agreement p_e expected from the marginals and the population joint frequency equals $r_X r_Y$. By the law of large numbers and the continuous-mapping theorem, both statistics converge to 0 almost surely and, being bounded, in expectation as $n \to \infty$; the denominators converge to positive constants for $r_X, r_Y \in (0, 1)$. Since $\#noise/n \leq 1$, $\mathbb{E}[\#noise \cdot \kappa]/n \to 0$, and likewise for φ. Combining both summands gives the claim. □

Finite-sample behavior and the role of the completion in (ii). The ARI component is exactly chance-corrected for every n, but κ and φ, as ratio estimators, carry a finite-sample deviation of order $1/n$; this is why Proposition 1 is stated asymptotically. Simulations (script `experiments/rebuttal/bias_null_model.py` in the repository, which draws the cluster labels of Y i.i.d. from the marginal of X in place of the positional permutation in (ii), leaving the asymptotics unchanged) quantify it: for $n = 20$, $k_{GT} = 3$, $r_X = 0.3$, $\mathbb{E}[\mathrm{NARI}_\kappa] = \mathbb{E}[\mathrm{NARI}_\varphi] = -0.010$; for $n = 1000$ with five equal clusters and $r_Y = r_X$, $|\mathbb{E}[\mathrm{NARI}_\kappa]| \leq 0.004$ across $r_X \in \{0, 0.1, 0.3, 0.5, 0.7, 0.85, 0.95\}$, and ARI_s, AMI_s, and ARI_c are within ± 0.003 of zero in the same sweep. The tiny example $X = [0, 0, 1, 1, 2, \text{noise}, \text{noise}]$ ($n = 7$) illustrates two subtleties. First, under the null of Proposition 1 exactly as stated, with $r_Y = 2/7$, simulation (script `experiments/rebuttal/reviewer_model_variants.py`) gives $\mathbb{E}[\mathrm{ARI}_s] = -0.005$ and $\mathbb{E}[\mathrm{NARI}_\kappa] = +0.06$ ($+0.13$ for $r_Y = 0.5$): at $n = 7$ the finite-sample deviation just described is still visible, and it shrinks to -0.010 at $n = 20$ and below 0.004 at $n = 1000$, as reported above. Second, the completion in step (ii) for the points in $S_Y \setminus S$ matters for the singleton strategy but not for the proposed measures: if those points receive fresh singleton labels instead of labels drawn from the marginal of X, $\mathbb{E}[\mathrm{ARI}_s]$ rises to about $+0.10$ and does not vanish with n, because the singleton encoding cannot distinguish a one-point predicted cluster from predicted noise. NARI_κ and NARI_φ are invariant to this choice, since points outside S never enter their ARI component; this encoding conflation is a further reason for an explicit noise component. Finally, we stress that the reading of (ii)

in which the cluster structure of X is kept intact and only noise flags are added (each cluster object independently flagged with probability r_Y, all other labels unchanged, which the ARI ignores up to renaming) is *not* a null model: it keeps every cluster intact and every GT noise object correctly flagged, and correspondingly yields $\mathbb{E}[\mathrm{ARI}_s] \approx 0.58$ and $\mathbb{E}[\mathrm{NARI}_\kappa] \approx 0.78$ for the same X, as a labeling that is correct up to random false flags should.

Proposition 2 (Supremum tightness)**.** *Suppose X and Y satisfy:*

(i) for every point p, X labels p as noise if and only if Y does, and

(ii) on the subset of points labeled as cluster objects (which by (i) is the same in both X and Y), X and Y induce the same partition up to permutation of cluster labels.

Then $\mathrm{NARI}_\kappa(X, Y) = \mathrm{NARI}_\varphi(X, Y) = 1$.

Proof. By (ii), the ARI of X and Y restricted to the cluster-object subset equals 1, since ARI attains its maximum on identical partitions. By (i), the binary noise-classification vectors of X and Y are identical, so any agreement coefficient computed on these two vectors equals 1; in particular, $\kappa = \varphi = 1$. Substituting,

$$\mathrm{NARI}_\kappa = \frac{\#data \cdot 1 + \#noise \cdot 1}{n} = \frac{\#data + \#noise}{n} = \frac{n}{n} = 1,$$

and the same identity holds for NARI_φ. □

Proposition 3 (No spurious optima)**.** *Conversely,* $\mathrm{NARI}_\kappa(X, Y) = 1$ *implies that X and Y satisfy conditions (i) and (ii) of Proposition 2, and the same implication holds for* NARI_φ.

Proof. Using $\#data + \#noise = n$, the equation $\mathrm{NARI}_\kappa(X, Y) = 1$ rearranges to

$$\#data\,(1 - \mathrm{ARI}) \;+\; \#noise\,(1 - \kappa) \;=\; 0. \tag{2}$$

Both $\mathrm{ARI} \le 1$ and $\kappa \le 1$, with equality iff perfect agreement on their respective inputs (standard properties of the two measures). Hence each summand in (2) is non-negative, and we proceed by cases on which weights are positive.

Case 1: $\#data > 0$ *and* $\#noise > 0$. Both coefficients in (2) are strictly positive, so the equation forces $\mathrm{ARI} = 1$ and $\kappa = 1$. Since κ is computed on the binary noise-indicator vectors of length n, $\kappa = 1$ means the two indicators agree at every point, which is exactly (i). And $\mathrm{ARI} = 1$ on the (now common) cluster-object subset means the induced partitions agree up to permutation, which is (ii).

Case 2: $\#data = 0$. Then $\#noise = n$ and (2) reduces to $\kappa = 1$, which forces every noise indicator to agree, giving (i); (ii) is vacuous since the cluster-object intersection is empty.

Case 3: $\#noise = 0$. Then $\#data = n$ and (2) reduces to $\mathrm{ARI} = 1$, giving (ii); (i) holds because neither labeling has any noise.

The same case analysis applies verbatim to NARI_φ, since $\varphi \le 1$ with equality iff perfect agreement on binary labels. □

Proposition 4 (Componentwise monotonicity)**.** *Fix a ground-truth labeling X, and let Y_1, Y_2 be two predicted labelings such that:*

(i) Y_1 and Y_2 have the same common cluster-object subset with X, i.e., $S_X \cap S_{Y_1} = S_X \cap S_{Y_2} =: S$: they flag the same GT cluster objects as noise, but may differ on the GT noise objects (so $\#data = |S|$ and $\#noise = n - |S|$ are common to both);

(ii) $\kappa(X, Y_2) \ge \kappa(X, Y_1)$ on the binary noise indicators;

(iii) $\mathrm{ARI}(X|_S, Y_2|_S) \ge \mathrm{ARI}(X|_S, Y_1|_S)$, where S is the common cluster-object subset.

Then $\mathrm{NARI}_\kappa(X, Y_2) \ge \mathrm{NARI}_\kappa(X, Y_1)$*, with strict inequality whenever at least one of (ii), (iii) is strict and the corresponding weight ($\#noise/n$ or $\#data/n$) is positive. The analogous statement holds for* NARI_φ.

Proof. Under (i), $\#data$ and $\#noise$ are shared between Y_1 and Y_2, and the ARI component depends only on the labels on S, so

$$\begin{aligned}&\mathrm{NARI}_\kappa(X, Y_2) - \mathrm{NARI}_\kappa(X, Y_1)\\ &\quad= \frac{\#data}{n}\big[\mathrm{ARI}(X|_S, Y_2|_S) - \mathrm{ARI}(X|_S, Y_1|_S)\big] + \frac{\#noise}{n}\big[\kappa(X, Y_2) - \kappa(X, Y_1)\big].\end{aligned}$$

Both bracketed terms are non-negative by (ii) and (iii), and both coefficients are non-negative, so the difference is non-negative. Strictness in either bracketed term combined with a positive coefficient yields strict inequality. The same argument applies to NARI_φ. □

Proposition 1 establishes chance correction, which in turn implies unbiasedness: a uniformly random labeling has expected score zero regardless of the number of clusters or the noise ratio (exactly for the ARI component and asymptotically in n for the κ/φ component, see the remark following Proposition 1). Propositions 2 and 3 together establish the supremum-tightness and no-spurious-optima properties: NARI_κ and NARI_φ attain their maximum on every dataset, and only on labelings that match the ground truth on both noise indicators and cluster partition. This addresses the failure modes of M_p (which violates supremum tightness by construction) and M_c, M_f (which assign perfect or near-perfect scores to clearly incorrect clusterings; Table 1). Proposition 4 formalizes monotonicity for a fixed common cluster-object subset: improving either component, the partition agreement on that subset or the noise-detection agreement on the GT noise objects, strictly increases the overall score. A monotonicity guarantee under arbitrary single-point relabelings is not available for any of the four measures (Section B.1), since the chance-correction term itself depends non-trivially on the marginals as the cluster-object subset changes; the stronger empirical claim that the score also tracks quality across the degradation schemes that move points between noise and cluster labels is supported by the experiments in Section 3.4 and Figure 5.

B.1 Monotonicity of the baselines is empirical

The monotonicity entries of all baselines in Tables 1 and 3 are established by the grid of seven degradation schemes and eight datasets (Figure 5 and Section 3.4); no proof exists, and none can be given in the componentwise sense of Proposition 4. To show this, we sample GT labelings X ($n = 20$, $k_{GT} = 3$, GT noise ratio 0.3) and pairs of predictions Y_1, Y_2 that share a random noise-indicator vector, a special case of condition (i) of Proposition 4 (predicted noise ratio drawn uniformly from $[0.1, 0.6]$) but carry independent random cluster labels on their cluster objects (script `experiments/rebuttal/monotonicity_search.py`). Since the noise indicators coincide, κ and φ coincide as well, and each pair is ordered so that $\mathrm{ARI}(X|_S, Y_2|_S) > \mathrm{ARI}(X|_S, Y_1|_S)$; pairs with equal restricted ARI are discarded. Conditions (i)–(iii) of Proposition 4 thus hold, and Proposition 4 guarantees $\mathrm{NARI}_\kappa(X, Y_2) > \mathrm{NARI}_\kappa(X, Y_1)$ and $\mathrm{NARI}_\varphi(X, Y_2) > \mathrm{NARI}_\varphi(X, Y_1)$. Out of 20,000 sampled pairs, 18,257 are comparable, and among these ARI_s is *smaller* for Y_2 than for Y_1 in 2,218 cases (12.2%) and AMI_s in 2,571 cases (14.1%): with the noise flags held fixed, improving the agreement on the cluster partition can decrease the singleton-encoded scores, because the singleton clusters enter the marginals of the chance-correction term of the encoded partition. A concrete instance is

$$\begin{aligned}X &= [0, 0, \text{noise}, 1, 0, \text{noise}, 1, \text{noise}, 0, 1, 2, \text{noise}, 1, \text{noise}, \text{noise}, \text{noise}, 0, \text{noise}, 0, 0],\\ Y_1 &= [\text{noise}, \text{noise}, \text{noise}, 2, 2, 0, 0, \text{noise}, 0, 0, 0, 2, 1, \text{noise}, 2, \text{noise}, 2, 1, 1, 0],\\ Y_2 &= [\text{noise}, \text{noise}, \text{noise}, 0, 1, 0, 0, \text{noise}, 0, 2, 1, 2, 1, \text{noise}, 0, \text{noise}, 1, 0, 2, 0],\end{aligned}$$

with restricted ARI -0.197 for Y_1 and -0.164 for Y_2, but $\mathrm{ARI}_s = -0.042$ for Y_1 and -0.052 for Y_2 (AMI_s: -0.034 vs. -0.046). The checkmarks of ARI_s and AMI_s in the monotonicity column therefore cannot be upgraded to a theorem, which is why Tables 1 and 3 mark them as empirical.

The same random search shows that none of the four measures is monotone under arbitrary single-point relabelings, in contrast to the perfect splits and merges of Gösgens et al. (2021), under which ARI and AMI are monotone. Starting from $Y = X$ at $n = 20$ and flipping one still-correct label of Y at a time to an arbitrary wrong label, including flips between noise and cluster labels, the score increases in 18.2% of the steps for ARI_s, 17.4% for AMI_s, 15.1% for NARI_κ, and 14.7% for NARI_φ. Such flag-changing single steps shift the marginals of the chance correction and are exactly the steps that Proposition 4 excludes; the structured degradation schemes of Figure 5, in which many points move in a consistent direction, are the regime in which we claim, and observe,

empirical monotonicity for all four recommended measures. Characterizing the conditions under which flag-correcting steps are provably monotone, in the perfect-oriented sense of Gösgens et al. (2021), is the natural generalization to pursue.

C Weighting ablation: why the weights depend on both labelings

NARI_κ and NARI_φ weight their two components by $\#data/n$ and $\#noise/n$, where $\#data$ counts the points labeled as cluster objects by *both* X and Y and $\#noise$ the points labeled as noise by *either*. We compare this symmetric weighting with the two one-sided alternatives, GT-only weights ($\#data$ and $\#noise$ computed from X alone) and prediction-only weights (computed from Y alone). The ARI and κ components are identical in all three variants; only the weights differ (script `experiments/rebuttal/weighting_ablation.py`).

Both one-sided weightings admit spurious optima. With GT-only weights, a GT without noise gives the noise component weight 0, so the score reduces to the ARI on the common cluster objects. A prediction that labels m of $n = 1000$ points (five equal clusters, no GT noise) as noise but preserves the partition on the remaining points then scores 1.000 for every m, up to $m = 900$, i.e., 90% of the data dumped into noise at a perfect score (Table 4, middle block); the symmetric weighting scores the same predictions $1 - m/n$. With prediction-only weights, the mirror failure occurs: on a GT with 85% noise ($n = 1000$, three clusters of 50 points each), a prediction that labels no point as noise and assigns each GT noise object a random cluster label, while preserving the GT partition on the cluster objects, has $\#noise = 0$ under prediction-only weights and scores 1.000; the symmetric weighting scores it 0.150 (Table 4, bottom block). In both cases the one-sided variant violates (P3). The symmetric definition is the only one of the three for which Proposition 3 holds, because it is the only one that routes every disagreement about noise into the κ/φ component, so that missed GT noise and spurious predicted noise carry the same penalty.

Cluster points wrongly predicted as noise are not discounted. A natural worry is that, because such points leave the ARI subset, their negative impact on the score is weakened. They are re-routed rather than omitted: each such point enters the noise-detection problem as a disagreement and increases the weight of that component. Table 4 (top block) shows a GT with five equal clusters and 10% GT noise ($n = 1000$), in which m randomly chosen cluster points of a perfect prediction are relabeled as noise. NARI_κ decreases strictly at every step, from 1.000 ($m = 0$) over 0.923 ($m = 100$) and 0.600 ($m = 400$) to 0.122 ($m = 800$), comparable to ARI_s (1.000, 0.863, 0.428, 0.021) with a slightly milder slope at small m. With GT-only weights, the same predictions would still score 0.902 at $m = 800$.

D Benchmark Datasets for clustering

Table 5 shows typically used benchmark datasets for clustering methods. However, their GT labels do not contain noise as they often originate from classification tasks.

In Table 6, we list the synthetic benchmark datasets that were used by the relevant papers listed in Section 2.1 to externally evaluate clusterings with noise labels. However, all but the last two of these datasets are 2-dimensional and, thus, only of limited use for evaluating modern clustering methods.

For generating synthetic datasets that contain density-based clusters, Beer et al. (2024) use the DENSIRED generator (Jahn et al., 2024). Furthermore, Maurus & Plant (2016) use a collection of 560 synthetic datasets available at `https://github.com/samhelmholtz/skinny-dip`, and Iglesias et al. (2021) use a collection of synthetic datasets available at `https://github.com/CN-TU/py_clureal-experiments`.

For the experiments in Section 3 and Section E, the datasets in Table 7 were used. The noisy-Gaussian dataset was generated using the Scikit-learn framework Pedregosa et al. (2011), and the denseAndSparse was created using the DENSIRED framework (Jahn et al., 2024). The other eight datasets have no associated feature space — they consist only of ground-truth label vectors, which we degrade step by step in our controlled experiments (Section 3.4).

Table 4: Weighting ablation ($n = 1000$). Top: m cluster points of a perfect prediction relabeled as noise (five equal clusters, 10% GT noise). Middle: m points of a perfect prediction relabeled as noise on a GT *without* noise (five equal clusters). Bottom: on a GT with 85% noise (three clusters of 50 points), a prediction without any noise labels that preserves the GT partition on the cluster objects. Symmetric weights are the definition used in the paper; the one-sided weightings certify spurious optima (orange). Entries marked with "–" are not applicable to the respective setting.

setting	m	NARI_κ symmetric	NARI_κ GT-only weights	NARI_κ prediction-only weights	ARI_s
cluster points → noise (10% GT noise)	0	1.000	1.000	–	1.000
	10	0.994	0.995	–	0.987
	25	0.984	0.988	–	0.967
	50	0.966	0.977	–	0.933
	100	0.923	0.962	–	0.863
	200	0.824	0.941	–	0.719
	400	0.600	0.920	–	0.428
	800	0.122	0.902	–	0.021
points → noise (no GT noise)	0	1.000	1.000	–	–
	100	0.900	1.000	–	–
	300	0.700	1.000	–	–
	500	0.500	1.000	–	–
	900	0.100	1.000	–	–
no predicted noise (85% GT noise)	–	0.150	–	1.000	–

Table 5: Common classification-origin benchmark datasets reused in clustering evaluation. None provide ground-truth noise labels, since every object carries exactly one class label. Columns: number of points (n), number of features (d), number of classes treated as clusters (k) and domain/source

Dataset	n	d	k	Domain
Iris (Fisher, 1936)	150	4	3	UCI / botany
Wine (Aeberhard & Forina, 1992)	178	13	3	UCI / chemistry
Glass (German, 1987)	214	9	6	UCI / materials
Breast Cancer (WDBC) (Wolberg et al., 1993)	569	30	2	UCI / medical
Optdigits (Alpaydin & Kaynak, 1998)	5 620	64	10	UCI / handwriting
USPS (Hull, 1994)	9 298	256	10	images / digits
Pendigits (Alpaydin & Alimoglu, 1996)	10 992	16	10	UCI / handwriting
Letter (Slate, 1991)	20 000	16	26	UCI / characters
COIL-20 (Nene et al., 1996a)	1 440	16 384	20	images / objects
COIL-100 (Nene et al., 1996b)	7 200	16 384	100	images / objects
MNIST (Deng, 2012)	70 000	784	10	images / digits
Fashion-MNIST (Xiao et al., 2017)	70 000	784	10	images / clothing
20 Newsgroups (Mitchell, 1997)	18 846	varies	20	text / news

Table 6: Synthetic benchmark datasets with noise labels that are used. Columns: Name of dataset, number of points (n), number of dimensions (d), number of ground-truth clusters (k), number of noise points, clustering model, source, and usage in related work.

Dataset	n	d	k	#noise	Model	Source	Used by
AGG	20 000	2	7	8 108	density	d'Errico et al. (2021)	d'Errico et al. (2021)
CLUS8	20 000	2	8	5 219	density	d'Errico et al. (2021)	d'Errico et al. (2021)
SPIR2	38 358	2	2	5 349	density	d'Errico et al. (2021)	d'Errico et al. (2021)
SPIR3	20 000	2	3	2 809	density	d'Errico et al. (2021)	d'Errico et al. (2021)
HORSE	20 000	2	3	7 346	density	d'Errico et al. (2021)	d'Errico et al. (2021)
Complex	2 155	2	6	155	density	Iglesias et al. (2021)	Iglesias et al. (2021)
High-noise	2 345	2	6	346	Gaussian	Iglesias et al. (2021)	Iglesias et al. (2021)
Low-noise	2 111	2	5	111	Gaussian	Iglesias et al. (2021)	Iglesias et al. (2021)
CURE-t2.4k	4 200	2	6	200	density	Guha et al. (1998)	Durani et al. (2022)
CLUTO-t4.8k	8 000	2	6	764	density	Karypis (2002)	Qiu & Yi (2025), Bryant & Cios (2018), Beer et al. (2026), Pirrone et al. (2018), Abbas et al. (2021)
CLUTO-t5.8k	8 000	2	6	1 153	density	Karypis (2002)	Qiu & Yi (2025), Beer et al. (2026), Pirrone et al. (2018), Abbas et al. (2021)
CLUTO-t7.10k	10 000	2	8	792	density	Karypis et al. (1999)	Karypis et al. (1999), Qiu & Yi (2025), Durani et al. (2022), Bryant & Cios (2018), Beer et al. (2026), Pirrone et al. (2018), Abbas et al. (2021)
CLUTO-t8.8k	8 000	2	8	323	density	Karypis et al. (1999)	Karypis et al. (1999), Beer et al. (2026), Pirrone et al. (2018), Abbas et al. (2021)
Zelnik2	303	2	2	106	density	Zelnik-Manor & Perona (2004)	Zelnik-Manor & Perona (2004)
Zelnik4	622	2	4	138	density	Zelnik-Manor & Perona (2004)	Zelnik-Manor & Perona (2004)
Synth_low	5 000	100	10	500	density	Jahn et al. (2024)	Beer et al. (2024); Draganov et al. (2025)
Synth_high	5 000	100	10	500	density	Jahn et al. (2024)	Beer et al. (2024); Draganov et al. (2025)

Table 7: Synthetic datasets created for the controlled experiments in Sections 3 and E. The noisy-Gaussian dataset was generated using scikit-learn Pedregosa et al. (2011) and denseAndSparse using DENSIRED Jahn et al. (2024); the other eight datasets are label vectors without a feature space. Columns: ground-truth clusters (k), noise ratio, and relative cluster sizes.

Dataset	k	%noise	Cluster ratios
noisyGaussian	3	90%	$1:1:1$
denseAndSparse	3	1%	$1:1:1$
1centNoise	3	1%	$1:1:1$
10centNoise	7	10%	random
85centNoise	4	85%	random
equalClusters	7	15%	$1:\ldots:1$
linearClusters	7	15%	$1:\ldots:7$
expClusters	7	15%	$e^1:\ldots:e^7$
fewClusters	2	15%	$1:1$
manyClusters	30	15%	random

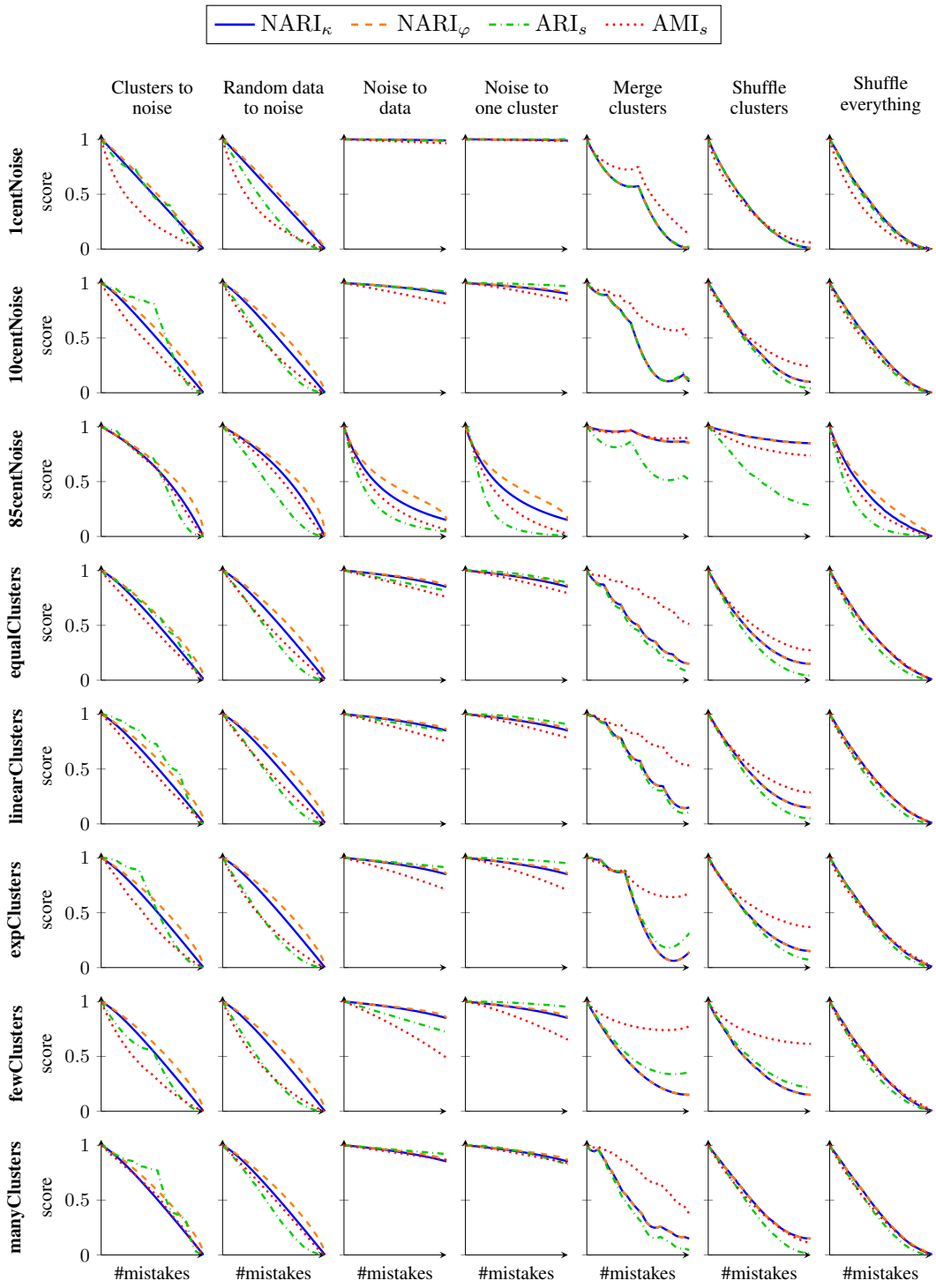
E RECOMMENDED CVIS BEHAVE AS INTENDED ACROSS DATASETS AND DEGRADATION SCHEMES

Figure 5 shows the scores of our proposed NARI_κ and NARI_φ alongside ARI_s and AMI_s — the two chance-corrected baselines that, like the proposed measures, satisfy all four properties (Table 1) — across the eight datasets in Table 7 and the seven degradation schemes in Table 8. Two takeaways. First, on every one of the 56 (dataset, scheme) cells, NARI_κ and NARI_φ decrease as the degradation progresses, up to small transient rebounds in the "merge clusters" scheme (at most 0.01 per step), where completing the absorption of one cluster into another temporarily restores some pair agreement; this generalizes the empirical monotonicity claim of Section 3.4 from the few dataset–scheme combinations shown there to all 56. Second, the proposed measures track ARI_s and AMI_s qualitatively across all cells. The grid is therefore a robustness check — the proposed measures are not only analytically sound (Section B) but behave consistently with their nearest P1–P4-satisfying competitors across the full range of synthetic degradations we tested.

Table 8: The seven degradation schemes used in the monotonicity experiments (Section 3.4). Starting from the GT labeling, points are reassigned to incorrect labels one by one according to each scheme. All schemes are deterministic given the stored point order (`noiseval/degradation.py`); each curve in Figure 5 is a single run.

Degradation scheme	Description
Clusters to noise:	the points of one cluster after another (in label order) are assigned a noise label.
Random data to noise:	data points are assigned a noise label one by one in their stored order, which is not grouped by cluster.
Noise to data:	noise points are assigned to the existing clusters in round-robin order.
Noise to one cluster:	noise points are assigned one new cluster label.
Merge clusters:	the points of all other clusters are assigned, cluster by cluster and point by point, to the first cluster.
Shuffle clusters:	data points are assigned the existing cluster labels in round-robin order, one point at a time.
Shuffle everything:	all points are assigned the cluster labels and the noise label in round-robin order, one point at a time.

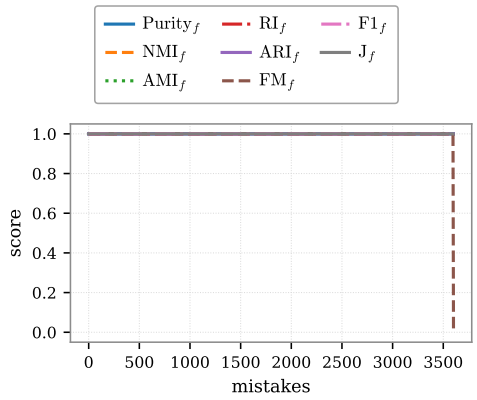
Figure 5: **Across all eight controlled datasets and seven degradation schemes, our proposed NARI_κ and NARI_φ behave consistently with the chance-corrected baselines ARI_s and AMI_s.** Columns: experiments. Rows: datasets.



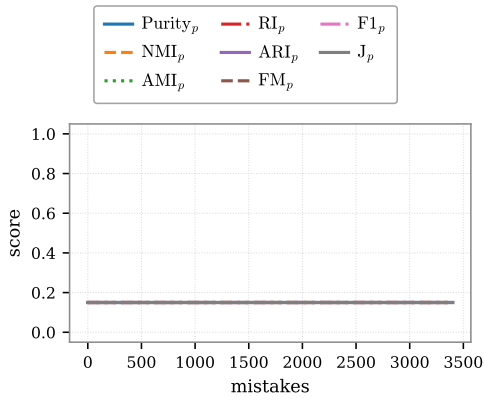
F FULL CURVES FOR THE PROPERTY VIOLATIONS IN FIGURE 3

Figure 3 in the main text shows, per noise-handling strategy, three violating CVI–strategy combinations and NARI_κ as a property-satisfying reference. Figure 6 shows the same four experiments with all eight base CVIs under the respective strategy. The curves of one strategy largely coincide, which is the point: under M_f and M_p the noise labeling is never assessed, so every base CVI is equally blind to the degradation; under M_s every unadjusted CVI is inflated by the singleton clusters; and under M_c the chance-corrected CVIs recover a perfect score once all noise has been absorbed into one cluster that coincides with the noise "cluster" of the GT.

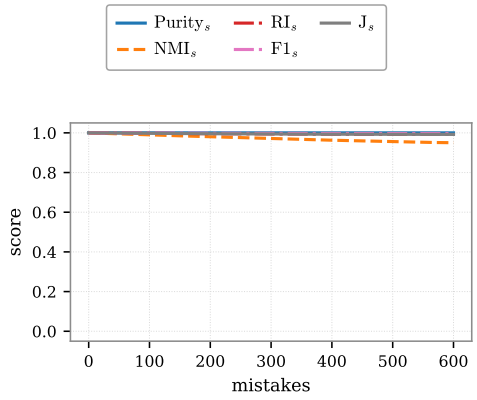
Figure 6: **Full curves of Figure 3: all eight base CVIs under one noise-handling strategy per panel.**



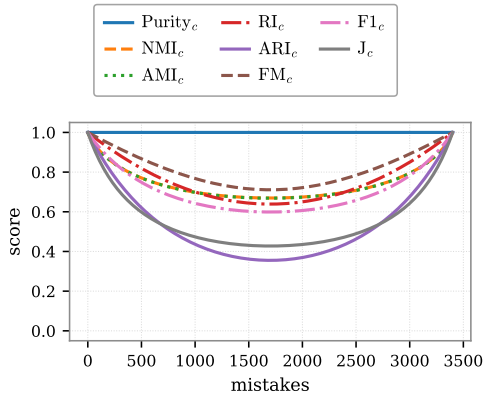
(a) M_f, scheme "Random data to noise" on data with 10% noise (cf. Figure 3a).



(b) M_p, scheme "Noise to data" on data with 85% noise (cf. Figure 3b).



(c) M_s without the chance-corrected CVIs, scheme "Random data to noise" on data with 85% noise (cf. Figure 3c).



(d) M_c, scheme "Noise to one cluster" on data with 85% noise (cf. Figure 3d).

G OUTLIER EVALUATION DATASETS.

As outliers and noise are a related concept (see Section 2), and there are more benchmark datasets for outlier detection than for clusterings with noise labels, we investigate them in Table 9. However, among the reviewed benchmarks, none is recommendable for evaluating the quality of noise-detecting clustering methods.

Table 9: Real-world datasets with outliers/noise in the ground truth labels, based on the dataset collections of Campos et al. (2016) and Rayana (2016). Name of dataset, number of points (n), dimensions (d), number of outliers, source, and applicability in clustering with noise labels. Some datasets have been modified for use in outlier detection research (by taking a subset of points, features, or both), indicated by the two partitions of the table.

	Dataset	n	d	#outliers	Source	Applicability
used without modifications	ANNThyroid	7 200	21	534	Campos et al. (2016), Kelly et al. (2023)	✗ default label is a class, not noise
	Arrhythmia	450	259	206	Campos et al. (2016), Kelly et al. (2023)	✗ outliers are minority classes
	Cardiotocography	2 126	21	471	Campos et al. (2016), Kelly et al. (2023)	✗ default label is a class, not noise
	Ecoli	336	7	9	Rayana (2016), Sathe & Aggarwal (2016)	✗ outliers are minority classes
	Glass	214	7	9	Campos et al. (2016), Keller et al. (2012)	✗ outliers are minority class
	Heart (SPECTF)	224	44	10	Rayana (2016), Kelly et al. (2023)	✗ binary classification dataset
	HeartDisease	270	13	120	Campos et al. (2016), Kelly et al. (2023)	✗ binary classification dataset
	Hepatitis	80	19	13	Campos et al. (2016), Kelly et al. (2023)	✗ binary classification dataset
	InternetAds	3 264	1 555	454	Campos et al. (2016), Kelly et al. (2023)	✗ binary classification dataset
	Ionosphere	351	32	126	Campos et al. (2016), Keller et al. (2012)	✗ binary classification dataset
	Lymphography	148	19	6	Campos et al. (2016), Kelly et al. (2023)	✗ outliers are minority classes
	Mammography	11 183	6	260	Rayana (2016), Abe et al. (2006)	✗ outliers are minority class
	PageBlocks	5 473	10	560	Campos et al. (2016), Kelly et al. (2023)	✗ outliers are minority classes
	Parkinson	195	22	147	Campos et al. (2016), Kelly et al. (2023)	✗ binary classification dataset
	Satellite	6 435	36	2036	Rayana (2016), Kelly et al. (2023)	✗ outliers are minority classes
	Seismic	2 584	11	170	Rayana (2016), Kelly et al. (2023)	✗ binary classification dataset
	Shuttle	1 013	9	13	Campos et al. (2016), Zhang et al. (2009)	✗ outliers are minority classes
	SpamBase	4 601	57	1 813	Campos et al. (2016), Kelly et al. (2023)	✗ binary classification dataset
	Speech	3 686	400	61	Rayana (2016), Micenková et al. (2014)	✗ outliers are minority classes
	Stamps	340	9	31	Campos et al. (2016), Micenková et al. (2015)	✗ binary classification dataset
	WPBC	198	33	47	Campos et al. (2016), Kelly et al. (2023)	✗ binary classification dataset
modified	ALOI	50 000	27	1 508	Campos et al. (2016), Geusebroek et al. (2005)	✗ outliers are downsampled classes
	KDDCup99	60 632	41	246	Campos et al. (2016), Kelly et al. (2023)	✗ outliers are minority classes
	Letter Recognition	1 600	32	100	Rayana (2016), Kelly et al. (2023)	✗ outliers are downsampled classes
	MNIST	7 603	100	700	Rayana (2016), Bandaragoda et al. (2014)	✗ outliers are downsampled class
	Musk	3 062	166	97	Rayana (2016), Kelly et al. (2023)	✗ outliers are minority classes
	OptDigits	5 216	64	150	Rayana (2016), Kelly et al. (2023)	✗ outliers are downsampled class
	PenDigits	9 868	16	20	Campos et al. (2016), Alpaydin & Alimoglu (1996)	✗ outliers are downsampled class
	Vertebral	240	6	30	Rayana (2016), Kelly et al. (2023)	✗ outliers are downsampled class
	Vowels	1 456	12	50	Rayana (2016), Kelly et al. (2023)	✗ outliers are downsampled class
	Waveform	3 443	21	100	Campos et al. (2016), Kelly et al. (2023)	✗ outliers are downsampled class
	WBC	454	9	10	Campos et al. (2016), Kelly et al. (2023)	✗ outliers are downsampled class
	WDBC	367	30	10	Campos et al. (2016), Kelly et al. (2023)	✗ outliers are downsampled class
	Wine	129	13	10	Rayana (2016), Kelly et al. (2023)	✗ outliers are downsampled class

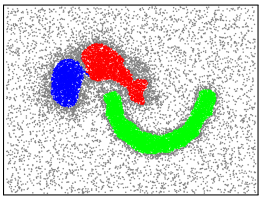
H DETAILED CLUSTERING EXAMPLES AND CVI SCORES.

In the following, we provide some per-dataset case studies referenced from Section 3: scatter plots show the labelings on the datasets and we give a detailed list of CVIs in the respective tables.

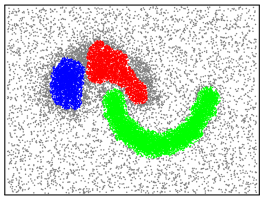
H.1 NOISY HORSE DATASET: DBSCAN CLUSTERING VS. CENTROID-BASED CLUSTERING

Figure 7 and Table 10 show the results for two different clusterings on the HORSE dataset (Table 6) that is similar to the well-known two-moons dataset and additionally has a lot of background noise: 7346 points out of 20 000 are GT noise. Using M_f does not reflect that the centroid-based clustering is significantly worse (i.e., less close to the GT) than the DBSCAN clustering.

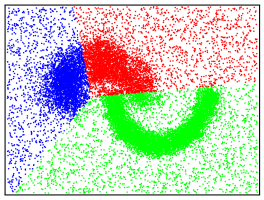
Figure 7: No spurious optima (P3): M_f rewards the Voronoi partition that ignores the cluster structure entirely. (a) ground truth of the HORSE dataset; (b) DBSCAN clustering; (c) manual Voronoi partition.



(a) ground truth



(b) DBSCAN: $\mathrm{ARI}_s = 0.87$, $\mathrm{NARI}_\kappa = 0.94$, $\mathrm{NARI}_\varphi = 0.94$, $\mathrm{Purity}_{\text{LNP-I}} = 1.00$



(c) Voronoi cells: $\mathrm{ARI}_s = 0.46$, $\mathrm{NARI}_\kappa = 0.59$, $\mathrm{NARI}_\varphi = 0.59$, $\mathrm{Purity}_{\text{LNP-I}} = 0.98$

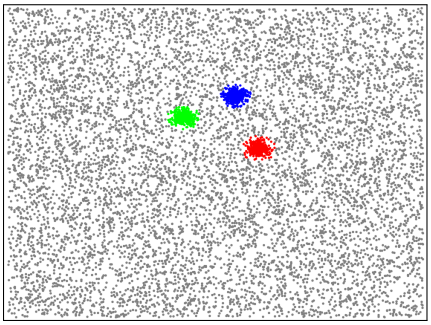
Table 10: The Voronoi partition shown in Figure 7 is clearly worse than the DBSCAN clustering, which should be reflected by CVIs. However, M_f rates the clearly wrong Voronoi partition almost as well as DBSCAN; our suggested scores highlight the difference correctly. Row colors indicate desired (green), acceptable (blue), and misleading (orange) behavior.

Measure	DBSCAN	Voronoi
NMI_f	0.9946	0.8969
AMI_f	0.9946	0.8969
RI_f	0.9993	0.9655
ARI_f	0.9986	0.9295
FM_f	0.9992	0.9597
$\mathrm{F1}_f$	0.9994	0.9699
J_f	0.9988	0.9416
NMI_p	0.6221	0.5675
AMI_p	0.6221	0.5675
RI_p	0.6251	0.6109
ARI_p	0.6246	0.5881
FM_p	0.6250	0.6072
$\mathrm{F1}_p$	0.6251	0.6137
J_p	0.6248	0.5957
$\mathrm{Purity}_{\text{LNP-I}}$	0.9992	0.9776
$\mathrm{NMI}_{\text{LNP-I}}$	0.9922	0.9861
AMI_s	0.8547	0.2438
ARI_s	0.8678	0.4625
FM_s	0.8942	0.6405
NARI_κ	0.9352	0.5881
NARI_φ	0.9382	0.5881

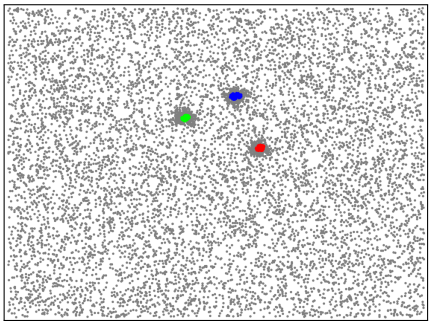
H.2 NOISY GAUSSIANS: DBSCAN VS LABELING EVERYTHING AS NOISE

Figure 8 and Table 11 show the results for two different clusterings on the noisyGaussian dataset (Table 7) that contains three equally sized Gaussian clusters and 90% background noise. We compare the DBSCAN clustering shown in Figure 8b with a labeling that assigns all points to noise. Using M_f does not even yield any well-defined scores as it disregards each noise point, which is in this case the full dataset. Under M_s, the unadjusted CVIs cannot distinguish the quite good DBSCAN clustering from the rather bad labeling that assigns everything to noise, whereas ARI_s, AMI_s, and the proposed measures assign the latter a score of 0.

Figure 8: No spurious optima (P3): under the singleton strategy, unadjusted CVIs score the labeling that assigns every point of noisyGaussian to noise near-perfectly (Table 11). (a) ground truth; (b) DBSCAN clustering.



(a) ground truth



(b) DBSCAN:
$\mathrm{ARI}_s = 0.18$, $\mathrm{NARI}_\kappa = 0.47$, $\mathrm{NARI}_\varphi = 0.55$, $\mathrm{Purity}_{\text{LNP-I}} = 1.00$

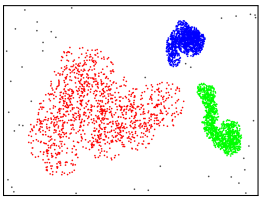
Table 11: Unadjusted CVIs under M_s cannot distinguish a relatively good DBSCAN clustering from a bad labeling that assigns everything to noise; the chance-corrected CVIs and the proposed measures can. Row colors indicate desired (green), acceptable (blue), and misleading (orange) behavior.

Measure	DBSCAN	all-noise
NMI_f	1.0000	-
AMI_f	1.0000	-
RI_f	1.0000	-
ARI_f	1.0000	-
FM_f	1.0000	-
$\mathrm{F1}_f$	1.0000	-
J_f	1.0000	-
NMI_s	0.9707	0.9612
RI_s	0.9960	0.9956
$\mathrm{F1}_s$	0.9980	0.9978
J_s	0.9960	0.9956
$\mathrm{Purity}_{\text{LNP-I}}$	1.0000	-
$\mathrm{NMI}_{\text{LNP-I}}$	0.9707	0.9612
AMI_s	0.3950	0.0000
ARI_s	0.1834	0.0000
FM_s	0.3184	0.0000
NARI_κ	0.4675	0.0000
NARI_φ	0.5536	0.0000

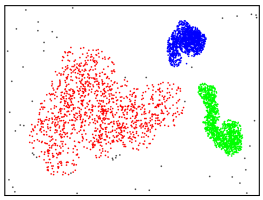
H.3 M_c MAKES BAD DBSCAN CLUSTERINGS LOOK GOOD

Figure 9 and Table 12 show the results for two different DBSCAN clusterings on the denseAndSparse dataset (Table 7) that contains three arbitrarily shaped clusters of different densities and 1% background noise. Figure 9b shows a very good DBSCAN clustering and Figure 9c shows a bad DBSCAN clustering, where the sparsest cluster is not detected, but instead completely labeled as noise. Using M_c does not show the quality difference of both clusterings as the scores in Table 12 show.

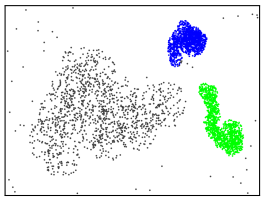
Figure 9: Noise-cluster strategy gives near-equal scores to two clearly different DBSCAN clusterings on denseAndSparse. (a) ground truth; (b) DBSCAN-1 (correct); (c) DBSCAN-2 (sparse cluster absorbed into noise).



(a) ground truth



(b) DBSCAN 1: $\mathrm{ARI}_s = 1.00$, $\mathrm{NARI}_\kappa = 1.00$, $\mathrm{NARI}_\varphi = 1.00$, $\mathrm{NMI}_c = 0.99$



(c) DBSCAN 2: $\mathrm{ARI}_s = 0.73$, $\mathrm{NARI}_\kappa = 0.67$, $\mathrm{NARI}_\varphi = 0.71$, $\mathrm{NMI}_c = 0.97$

Table 12: NMI_c assigns near-identical scores to the two qualitatively different clusterings in Figure 9; NARI_κ, NARI_φ, and ARI_s separate them by $\Delta \approx 0.3$. AMI_s by $\Delta \approx 0.6$. Row colors indicate desired (green), acceptable (blue), and misleading (orange) behavior.

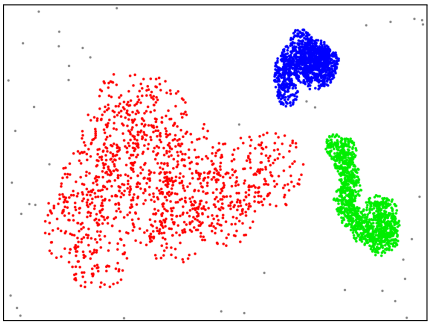
Measure	DBSCAN 1	DBSCAN 2
NMI_c	0.9888	0.9701
AMI_c	0.9888	0.9701
RI_c	0.9980	0.9911
ARI_c	0.9954	0.9798
FM_c	0.9969	0.9865
$\mathrm{F1}_c$	0.9985	0.9933
J_c	0.9970	0.9867
AMI_s	0.9895	0.3921
ARI_s	0.9955	0.7261
FM_s	0.9970	0.8143
NARI_κ	0.9983	0.6713
NARI_φ	0.9984	0.7059

I ADDITIONAL PARAMETER SEARCH RESULTS.

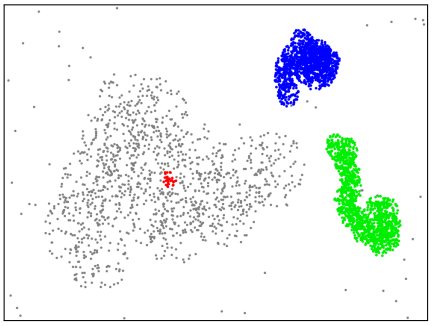
In addition to the CLUTO-t4.8k results in Section 3.6, Figures 10 and 11 show parameter search results on the denseAndSparse and noisyGaussian datasets. These highlight that some currently used external CVIs cannot be used in order to find the best clustering across a range of parameter settings: Figure 10b shows that ARI_c scores a DBSCAN clustering that drops the sparse cluster at 0.97 on the denseAndSparse dataset (Table 7), barely below the correct clustering. Figure 11 shows that ARI_f, ARI_p, and NMI_{LNP-I} mark DBSCAN clusterings as optimal that find a lot of spurious clusters in uniform random noise on the noisyGaussian dataset (Table 7).

In contrast, on both datasets, ARI_s yields meaningful parameter settings.

Figure 10: On denseAndSparse, the noise-cluster strategy (ARI_c) assigns a near-optimal score of 0.97 to a parameter setting that drops the sparse cluster, while ARI_s recovers the structure. DBSCAN parameter search: (a) selected by ARI_s — correct; (b) near-optimum of ARI_c at 0.97 — sparse cluster mostly assigned to noise.

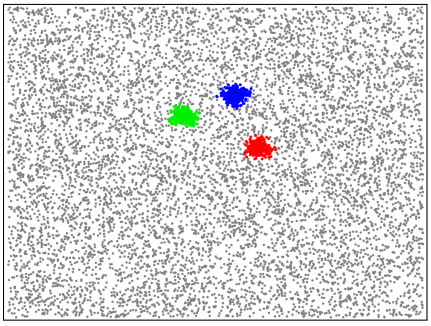


(a) Optimal clustering according to ARI_s ($\varepsilon = 0.89$, $\mathrm{minPts} = 4$)

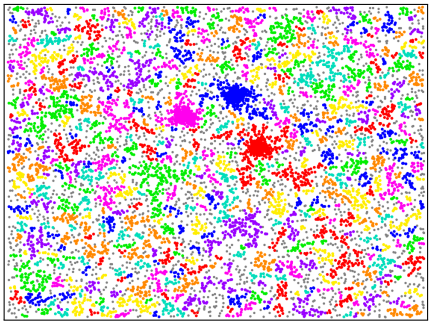


(b) Near-optimal clustering according to ARI_c, with a score of 0.97 ($\varepsilon = 0.45$, $\mathrm{minPts} = 15$)

Figure 11: Filtering, penalty, and $\mathrm{NMI}_{\text{LNP-I}}$ all select a clustering that finds spurious clusters in pure noise; ARI_s selects the realistic one. DBSCAN parameter search on noisyGaussian: (a) selected by ARI_s; (b) selected by ARI_f, ARI_p, and $\mathrm{NMI}_{\text{LNP-I}}$.



(a) Optimal clustering according to ARI_s ($\varepsilon = 0.23$, $\mathrm{minPts} = 14$)



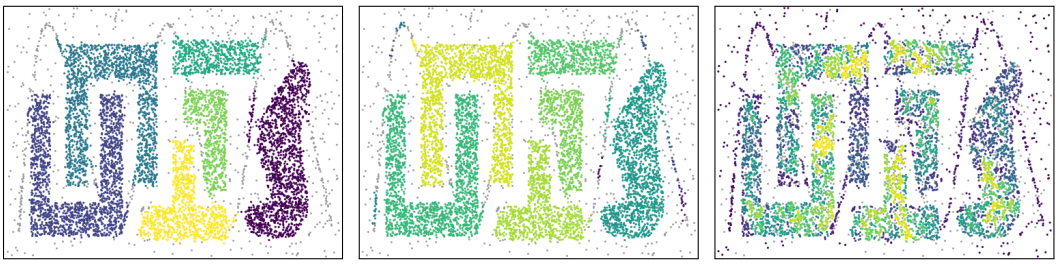
(b) Optimal clustering according to ARI_f, ARI_p, and $\mathrm{NMI}_{\text{LNP-I}}$ ($\varepsilon = 0.21$, $\mathrm{minPts} = 4$)

J PARAMETER SEARCH WITH HDBSCAN.

To verify that the parameter-search failure modes shown in Section 3.6 are not specific to DBSCAN, we repeat the experiment with HDBSCAN (Campello et al., 2015) on the CLUTO-t4.8k dataset, gridding over *min_cluster_size* $\in \{2, \ldots, 200\}$ (15 values, geometric scale) and *min_samples* $\in \{1, \ldots, 15\}$. (The case study in Section L uses a coarser 4×4 grid, which is why its NARI_κ-picked HDBSCAN configuration differs.)

Figure 12 shows the same pattern observed under DBSCAN: NARI_κ selects a clustering that correctly identifies all six clusters; ARI_f achieves a perfect score of 1.000 on a clustering that over-segments the data into 16 clusters, since filtering removes the discarded points before evaluation; and $\mathrm{Purity}_{\text{LNP-I}}$ likewise scores 1.000 on a degenerate clustering with 2368 microclusters. The failures are properties of the CVI–strategy combinations, not of the clustering algorithm.

Figure 12: Optimal HDBSCAN clusterings on CLUTO-t4.8k according to three CVIs. (a) is selected by our proposed NARI_κ and correctly identifies all six clusters. (b) is selected by ARI_f with a perfect score of 1.000, despite over-segmenting into 16 clusters. (c) is selected by $\mathrm{Purity}_{\text{LNP-I}}$ with a perfect score, despite assigning 2368 microclusters.



(a) NARI_κ: *min_cluster_size* $= 28$, *min_samples* $= 13$, $k = 6$.

(b) ARI_f: *min_cluster_size* $= 4$, *min_samples* $= 13$, $k = 16$.

(c) $\mathrm{Purity}_{\text{LNP-I}}$: *min_cluster_size* $= 2$, *min_samples* $= 1$, $k = 2368$.

K RUNTIME COMPARISON OF CVIS.

To support the runtime claim in Section 4, we measured the wall-clock cost of computing the four base CVIs that, paired with an appropriate noise-handling strategy, satisfy all four properties: ARI, AMI, Cohen's κ, and Matthews' φ. Inputs are independent uniformly random labelings on n points with $K_{\text{true}} = K_{\text{pred}} = 20$ clusters for the multi-class measures (ARI, AMI), and independent Bernoulli noise indicators with noise ratio 0.2 for the binary measures (κ, φ). For AMI_s, the singleton encoding raises k_{pred} to the order of the number of noise points, so the fixed-K timing is a lower bound on its cost. Each timing is the median of seven runs after one warm-up call, using the standard scikit-learn implementation. Measurements were taken on an AMD EPYC-Genoa CPU with 32 GiB RAM, running Linux 6.8, Python 3.13.9, scikit-learn 1.8.0, NumPy 2.4.4, and SciPy 1.17.1. Absolute times are hardware-dependent; the multiplicative ratios between measures are what supports the argument.

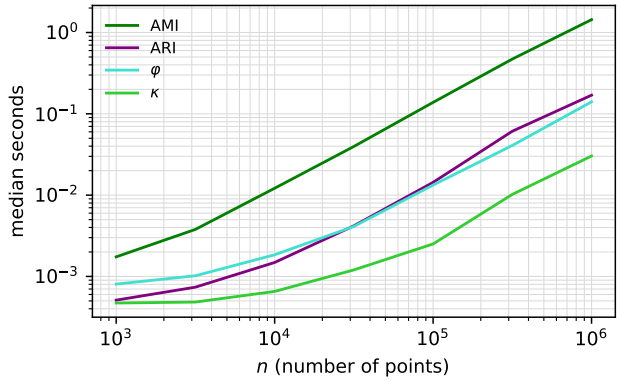


Figure 13: **AMI is roughly an order of magnitude slower than ARI across the entire $n = 10^3$–10^6 sweep; κ and φ are the fastest.** Wall-clock cost of computing the four CVIs satisfying all four properties, on independent random labelings of n points (multi-class with $K = 20$ clusters; binary at noise ratio 0.2).

At $n = 10^6$, AMI takes approximately 1.4 seconds per call, compared to 0.17 seconds for ARI and 0.03 seconds for κ. The roughly $8 - 10\times$ multiplicative gap between AMI and ARI is approximately constant across the log sweep, consistent with the $\Theta(R \cdot C)$ log-gamma overhead in AMI's chance-correction formula. The script that produced these measurements is at `experiments/runtime/run.py`.

L REAL-METHOD CASE STUDY: SHADE, DBSCAN, HDBSCAN.

To confirm that the issues identified in Section 3 affect not only DBSCAN parameter search but also other, more recent noise-producing clustering methods, we compare three methods that assign noise labels: SHADE (Beer et al., 2024) (a deep density-based clustering method), DBSCAN, and HDBSCAN. We pick SHADE as the deep-clustering representative because it is the most recent noise-detecting clustering method in our survey that primarily optimizes clustering quality (Table 2)[4]. Following the deep-clustering evaluation convention, SHADE is run at the hyperparameters reported in (Beer et al., 2024) across three seeds; DBSCAN and HDBSCAN are run on the parameter grids from Sections 3.6 and J and we report the NARI_κ-argmax configuration. We highlight two datasets; full results for all benchmarks are in our repository.

Designed-noise regime (Synth_high). On the synthetic benchmark from Jahn et al. (2024) with 5000 points in 100 dimensions and 10% uniform background noise, all three methods recover the cluster structure on the data points: ARI_c saturates at or near 1 (Table 13). The methods nonetheless differ markedly in how much they flag as noise — SHADE assigns $0.3-2\%$ noise labels, while HDBSCAN matches the GT proportion. ARI_c absorbs that difference into a noise cluster and reports the methods as indistinguishable; NARI_κ separates them by an order of magnitude in the residual error. The ranking changes as well: under ARI_c, SHADE ties with HDBSCAN at 1.000 and DBSCAN ranks last (0.967), whereas NARI_κ ranks DBSCAN (0.977) above SHADE (0.896), because SHADE flags only a fraction of the noise the GT contains.

Table 13: Designed-noise regime: scores at the NARI_κ-picked configuration on Synth_high (5000 points, 100 dimensions, 10% uniform noise). ARI_c saturates and hides differences in noise emission; NARI_κ surfaces them.

method	ARI_c	NMI_c	ARI_s	NARI_κ	NARI_φ
DBSCAN	0.967	0.948	0.974	0.977	0.977
HDBSCAN	1.000	1.000	1.000	1.000	1.000
SHADE	1.000	1.000	0.954	0.896	0.895

No-designed-noise regime (htru2). On the htru2 pulsar-detection benchmark (Lyon, 2015) ($n = 17898$, two classes, no GT noise), the three methods nonetheless emit very different amounts of spurious noise. The result is a complete cross-method ranking inversion (Table 14 and fig. 14): under ARI_c, DBSCAN ranks highest and SHADE lowest; under NARI_κ, this is reversed. Because the GT contains no noise, every point a method flags as noise disagrees with a constant reference, so $\kappa = 0$ (and $\varphi = 0$ by the convention of Section 3.5) and NARI_κ reduces to $\frac{\#data}{n}\ \mathrm{ARI}(X|_S, Y|_S) = \mathrm{ARI}_p$: spurious noise emission is penalized linearly through the weight, and the partition is scored on the points the method did cluster. ARI_s penalizes the same points through all pairs they form with their GT class, which is why it sides with ARI_c here. ARI_c absorbs all method-emitted noise into a single cluster, which can either help or hurt depending on its alignment with the underlying classes. Neither evaluator is "wrong", but they disagree on which method one would report. This is the central concern of Section 3.

Table 14: No-designed-noise regime: scores at the NARI_κ-picked configuration on htru2. ARI_c ranks DBSCAN > HDBSCAN > SHADE; NARI_κ inverts this to SHADE > HDBSCAN ≈ DBSCAN.

method	ARI_c	NMI_c	ARI_s	NARI_κ	NARI_φ
DBSCAN	**0.525**	0.302	0.534	0.532	0.532
HDBSCAN	0.430	0.295	0.454	0.545	0.545
SHADE	0.359	0.293	0.385	**0.671**	0.671

The htru2 inversion is adjudicable structurally. htru2 has no GT noise to settle the disagreement directly, but the GT itself is a known binary classification ($\approx 9\%$ positives, $n = 17{,}898$), so a faithful clustering should produce two structurally meaningful groups. The configurations that ARI_c favors do not. Table 15 shows the ARI_c-picked configuration of each method: DBSCAN and HDBSCAN

[4]FairDen is newer but optimizes for fairness rather than clustering quality alone, which would conflate the comparison

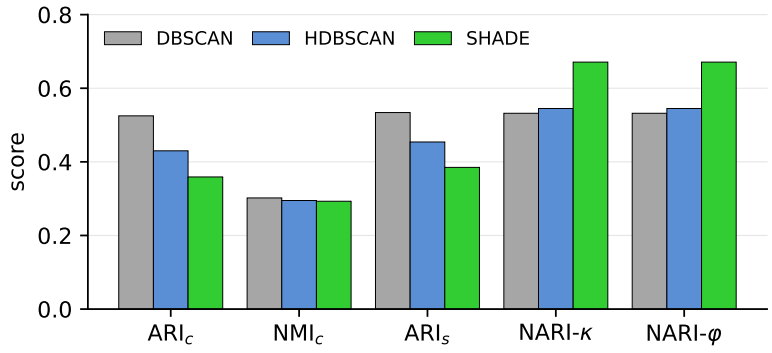


Figure 14: **Cross-method ranking inverts on htru2 between** ARI_c **and** NARI_κ**.** Same scores as Table 14; bars at the NARI_κ-picked configuration of each method. DBSCAN dominates the left pair of CVIs; SHADE dominates the right pair.

both produce one cluster covering 86% of the data, five microclusters totalling about 220 and 470 points, respectively, and $\approx 12\%$ noise — an essentially trivial "everything-in-one-bin" structure with token fragmentation. SHADE's configuration produces a two-cluster + noise structure (76%, 2.3%, 22% noise) that approximately recovers the binary GT topology. None of the three methods perfectly recovers the GT classes, but the inversion is no longer a tie requiring theoretical adjudication: ARI_c scores trivially-fragmented near-single-cluster outputs above the structurally faithful one, while NARI_κ does not. This is the P3 spurious-optima failure mode of ARI_c from Section 3 playing out on real data, verifiable directly from the cluster-size distributions of the picked configurations.

Table 15: Cluster-size structure of the ARI_c-picked configuration of each method on htru2 ($n = 17{,}898$, binary GT). DBSCAN and HDBSCAN's winning configurations under ARI_c are structurally trivial (one near-total cluster + a handful of microclusters); SHADE's is a binary partition that aligns with the GT topology.

method	picked config	largest cluster	# clusters	# noise	ARI_c	NARI_κ
DBSCAN	mp14_eps0.5276	15,409 (86%)	6	2,268	0.539	0.440
HDBSCAN	mcs43_ms1	15,326 (86%)	6	2,102	0.525	0.535
SHADE	seed2_mp5	13,598 (76%)	2	3,888	0.359	0.671

Table 16: Cross-method comparison on benchmarks with ground-truth noise: scores at the NARI_κ-picked configuration of each method. ARI_c and NARI_κ agree on the cross-method ranking on the three 2D benchmarks (CLUTO-t4.8k, CLUTO-t8.8k, HORSE). On Synth_low, as on Synth_high (Table 13), ARI_c ties SHADE with HDBSCAN at 1.000, whereas NARI_κ ranks SHADE below DBSCAN because SHADE flags far fewer points as noise than the GT contains.

dataset	method	configid	ARI_c	NMI_c	ARI_s	NARI_κ	NARI_φ
Synth_low	DBSCAN	mp11_eps0.097	0.964	0.948	0.974	0.976	0.977
	HDBSCAN	mcs10_ms1	1.000	1.000	1.000	1.000	1.000
	SHADE	seed1_mp150	1.000	1.000	0.954	0.900	0.900
CLUTO-t4.8k	DBSCAN	mp16_eps8.93	0.972	0.958	0.981	0.990	0.990
	HDBSCAN	mcs200_ms10	0.952	0.936	0.965	0.983	0.984
	SHADE	seed0_mp5	0.574	0.686	0.616	0.691	0.699
CLUTO-t8.8k	DBSCAN	mp6_eps8.16	0.942	0.914	0.947	0.952	0.953
	HDBSCAN	mcs43_ms15	0.905	0.900	0.924	0.928	0.934
	SHADE	seed1_mp5	0.932	0.908	0.942	0.947	0.948
HORSE	DBSCAN	mp15_eps0.45	0.857	0.806	0.924	0.946	0.947
	HDBSCAN	mcs200_ms5	0.774	0.736	0.856	0.911	0.912
	SHADE	seed0_mp5	0.596	0.536	0.825	0.752	0.755

Table 16 also shows that on the 2D shape benchmarks (CLUTO and HORSE), NARI_κ is in fact *more generous* to SHADE than ARI_c is (e.g. on HORSE: $\mathrm{ARI}_c = 0.60$ vs. $\mathrm{NARI}_\kappa = 0.75$ for SHADE). SHADE assigns substantial noise on 2D data because it operates outside its high-dimensional design regime, but the noise it assigns partially overlaps with GT noise. NARI_κ scores that overlap on its merits via the noise component; ARI_c instead absorbs all of SHADE's emitted noise into a single

cluster that misaligns with GT, depressing the score. This is consistent across all three 2D benchmarks and rules out the worry that NARI_κ is structurally biased against deep methods that emit noise.

Combined with the empirical evidence already in the paper for individual methods — Section 3.6 (DBSCAN parameter search on denseAndSparse; ARI_c assigns 0.97 to a clustering that visibly merges the sparse cluster into noise), Section H (HORSE; the Voronoi partition that ignores clusters scores $\mathrm{NMI}_{\text{LNP-I}} = 0.99$ but $\mathrm{NARI}_\kappa = 0.59$), and Section J (HDBSCAN, ruling out an algorithm-specific artifact) — this gives us empirical calibration of NARI_κ against ground truth on every benchmark where ground truth is available. Where NARI_κ disagrees with ARI_c (htru2 and the two Synth benchmarks), the property analysis in Section B shows ARI_c violating P3 on degenerate clusterings, while NARI_κ satisfies P1–P4; we therefore prefer NARI_κ's ranking on adjudication grounds.

Across the remaining six clustpy benchmarks (har, optdigits, pendigits, letterrecognition, mice_protein, tcga; see our repository), ARI_c and NARI_κ agree on the cross-method ranking but differ in absolute scores by amounts comparable to the spreads above. The sweep and aggregation scripts and the per-configuration scores are available in our repository (`https://anonymous.4open.science/r/External-Evaluation-of-Clusterings-with-Noise-Labels-F777`) at `experiments/shade/`.

M Real data with non-synthetic noise labels.

Real-world datasets with reliable, diffuse GT noise are scarce (Section 2.1). To test whether the selection failures of Section 3.6 transfer to real feature distributions and real cluster geometry, we obtain real, non-injected GT noise labels in two ways. Both experiments share the protocol of Sections 3.6 and J: features are standardized, DBSCAN is run on a 15×15 grid ($\mathrm{minPts} = 2, \ldots, 16$; $\varepsilon = f \cdot \varepsilon_k$, where ε_k is the median k-dist for $k = \mathrm{minPts}$ and f takes 15 values on an exponential scale from 0.5 to 2), HDBSCAN on a 4×4 grid (minimum cluster size $\in \{2, 9, 43, 200\}$, minimum samples $\in \{1, 5, 10, 15\}$), and every CVI–strategy combination selects the configuration that maximizes it. We report the selected configuration's shape (k_{pred} and the fraction of predicted noise), the selecting measure's own score of it (self-score), and its quality under NARI_κ and ARI_s. When a selected clustering labels every point as noise, the filtered evaluation set is empty and the standard implementation returns a score of 1.0, which we report as such; Table 11 marks the same case as undefined. Crucially, the selection pathologies of filtering and Purity depend only on the fraction of points labeled as noise, not on their spatial geometry, so both settings are valid tests of whether those pathologies transfer, even where the noise construction is artificial as a data model. An evaluation measure only sees the two label vectors; what determines its behavior is the combinatorial structure it is fed, namely the noise fraction, the cluster-size distribution, and the overlap pattern between predicted and GT noise. Both protocols supply exactly these on real data with a controlled noise set. Scripts: `experiments/rebuttal/real_noise_minority.py`, `fm_embed_cifar.py`, and `fm_noise_annotation.py`.

M.1 Held-out minority classes as noise

We take four classification datasets from OpenML (Vanschoren et al., 2014): optdigits and pendigits (Alpaydin & Kaynak, 1998; Alpaydin & Alimoglu, 1996), MiceProtein (Higuera et al., 2015), and HAR (Anguita et al., 2013). We designate one or two classes as noise (classes 8 and 9 for optdigits and pendigits, the two smallest classes for MiceProtein, the smallest class for HAR), subsample the designated points so that they make up $\approx 5\%$ of the final dataset, and use the remaining classes as the GT cluster structure; rows of MiceProtein with missing values are dropped. As noted in Section 2.1, held-out classes form compact microclusters rather than diffuse noise. We use this protocol because it is the only way to obtain real, non-synthetic noise labels at scale, and we do not claim that it models how noise arises in the wild.

Table 17 shows that the failure modes identified synthetically reproduce on all four datasets: ARI_f selects degenerate clusterings with 99–100% of the points labeled as noise on every dataset, and $\mathrm{Purity}_{\text{LNP-I}}$ selects 98–100%-noise microcluster solutions, both at a perfect self-score of 1.000, while the P1–P4-satisfying measures ARI_s, AMI_s, NARI_κ, and NARI_φ select structurally sensible clusterings (e.g., $k_{pred} = 8 = k_{GT}$ on optdigits). ARI_c and NMI_c select reasonable configurations in

this experiment as well; our case against M_c rests on the spurious optima demonstrated in Section 3 and Section L, not on picker degeneracy here.

Table 17: Real data with held-out minority classes as GT noise ($\approx 5\%$): the configuration selected by each CVI–strategy combination from a DBSCAN 15×15 grid plus an HDBSCAN 4×4 sweep, its number of clusters, its fraction of predicted noise, the selecting measure's own score of that configuration (self-score), and the configuration's quality under NARI_κ and ARI_s. ε_k denotes the median k-dist with $k = \mathrm{minPts}$. Row colors: property-satisfying pickers (green) and pickers selecting degenerate, almost-all-noise clusterings at a perfect self-score (orange).

picking measure	selected configuration	k_{pred}	pred. noise	self-score	NARI_κ of pick	ARI_s of pick
optdigits ($n = 4741$, $d = 64$, $k_{GT} = 8$; GT noise: classes 8, 9)						
ARI_c	HDBSCAN, mcs= 200, ms= 1	7	23.2%	0.545	0.715	0.667
NMI_c	HDBSCAN, mcs= 43, ms= 1	8	27.8%	0.739	0.775	0.709
ARI_s	HDBSCAN, mcs= 43, ms= 1	8	27.8%	0.709	0.775	0.709
AMI_s	HDBSCAN, mcs= 200, ms= 1	7	23.2%	0.657	0.715	0.667
ARI_f	DBSCAN, $\mathrm{minPts} = 4$, $\varepsilon = 0.552\,\varepsilon_k$	1	99.0%	1.000	0.012	0.002
ARI_p	HDBSCAN, mcs= 43, ms= 1	8	27.8%	0.715	0.775	0.709
$\mathrm{NMI}_{\text{LNP-I}}$	HDBSCAN, mcs= 200, ms= 1	7	23.2%	0.752	0.715	0.667
$\mathrm{Purity}_{\text{LNP-I}}$	DBSCAN, $\mathrm{minPts} = 2$, $\varepsilon = 0.500\,\varepsilon_k$	11	99.4%	1.000	0.001	0.000
NARI_κ	HDBSCAN, mcs= 43, ms= 1	8	27.8%	0.775	0.775	0.709
NARI_φ	HDBSCAN, mcs= 43, ms= 1	8	27.8%	0.806	0.775	0.709
pendigits ($n = 9349$, $d = 16$, $k_{GT} = 8$; GT noise: classes 8, 9)						
ARI_c	HDBSCAN, mcs= 43, ms= 1	15	12.0%	0.729	0.788	0.774
NMI_c	HDBSCAN, mcs= 43, ms= 1	15	12.0%	0.804	0.788	0.774
ARI_s	HDBSCAN, mcs= 43, ms= 1	15	12.0%	0.774	0.788	0.774
AMI_s	HDBSCAN, mcs= 43, ms= 1	15	12.0%	0.764	0.788	0.774
ARI_f	DBSCAN, $\mathrm{minPts} = 11$, $\varepsilon = 0.500\,\varepsilon_k$	2	99.8%	1.000	0.003	0.000
ARI_p	HDBSCAN, mcs= 43, ms= 1	15	12.0%	0.749	0.788	0.774
$\mathrm{NMI}_{\text{LNP-I}}$	HDBSCAN, mcs= 43, ms= 1	15	12.0%	0.818	0.788	0.774
$\mathrm{Purity}_{\text{LNP-I}}$	DBSCAN, $\mathrm{minPts} = 2$, $\varepsilon = 0.500\,\varepsilon_k$	81	98.1%	1.000	0.003	0.000
NARI_κ	HDBSCAN, mcs= 200, ms= 1	9	16.2%	0.794	0.794	0.766
NARI_φ	HDBSCAN, mcs= 200, ms= 1	9	16.2%	0.804	0.794	0.766
MiceProtein ($n = 471$, $d = 77$, $k_{GT} = 6$; GT noise: two smallest classes)						
ARI_c	DBSCAN, $\mathrm{minPts} = 5$, $\varepsilon = 1.346\,\varepsilon_k$	13	11.5%	0.375	0.425	0.387
NMI_c	HDBSCAN, mcs= 9, ms= 1	24	8.5%	0.636	0.351	0.311
ARI_s	DBSCAN, $\mathrm{minPts} = 6$, $\varepsilon = 1.219\,\varepsilon_k$	13	15.3%	0.389	0.432	0.389
AMI_s	HDBSCAN, mcs= 9, ms= 1	24	8.5%	0.621	0.351	0.311
ARI_f	DBSCAN, $\mathrm{minPts} = 2$, $\varepsilon = 0.500\,\varepsilon_k$	0	100.0%	1.000	0.000	0.000
ARI_p	DBSCAN, $\mathrm{minPts} = 6$, $\varepsilon = 1.219\,\varepsilon_k$	13	15.3%	0.395	0.432	0.389
$\mathrm{NMI}_{\text{LNP-I}}$	HDBSCAN, mcs= 9, ms= 1	24	8.5%	0.701	0.351	0.311
$\mathrm{Purity}_{\text{LNP-I}}$	DBSCAN, $\mathrm{minPts} = 2$, $\varepsilon = 0.500\,\varepsilon_k$	0	100.0%	1.000	0.000	0.000
NARI_κ	DBSCAN, $\mathrm{minPts} = 6$, $\varepsilon = 1.219\,\varepsilon_k$	13	15.3%	0.432	0.432	0.389
NARI_φ	DBSCAN, $\mathrm{minPts} = 6$, $\varepsilon = 1.219\,\varepsilon_k$	13	15.3%	0.439	0.432	0.389
HAR ($n = 9361$, $d = 561$, $k_{GT} = 5$; GT noise: smallest class)						
ARI_c	HDBSCAN, mcs= 200, ms= 1	2	17.6%	0.307	0.326	0.319
NMI_c	HDBSCAN, mcs= 200, ms= 1	2	17.6%	0.459	0.326	0.319
ARI_s	HDBSCAN, mcs= 200, ms= 1	2	17.6%	0.319	0.326	0.319
AMI_s	DBSCAN, $\mathrm{minPts} = 5$, $\varepsilon = 1.219\,\varepsilon_k$	18	17.0%	0.448	0.326	0.316
ARI_f	DBSCAN, $\mathrm{minPts} = 3$, $\varepsilon = 0.500\,\varepsilon_k$	0	100.0%	1.000	0.000	0.000
ARI_p	HDBSCAN, mcs= 200, ms= 1	2	17.6%	0.266	0.326	0.319
$\mathrm{NMI}_{\text{LNP-I}}$	HDBSCAN, mcs= 200, ms= 1	2	17.6%	0.566	0.326	0.319
$\mathrm{Purity}_{\text{LNP-I}}$	DBSCAN, $\mathrm{minPts} = 2$, $\varepsilon = 0.500\,\varepsilon_k$	10	99.8%	1.000	0.001	0.000
NARI_κ	HDBSCAN, mcs= 200, ms= 1	2	17.6%	0.326	0.326	0.319
NARI_φ	HDBSCAN, mcs= 200, ms= 1	2	17.6%	0.341	0.326	0.319

M.2 Foundation-model-annotated noise on CIFAR-10

As a strictly more realistic labeling source than a minority-class rule, we let a foundation model annotate the noise. We embed the 10,000 CIFAR-10 test images (Krizhevsky, 2009) with CLIP (ViT-B/32) (Radford et al., 2021) and score every image against the ten class prompts by zero-shot classification (zero-shot accuracy 0.862). Images that CLIP cannot assign to any class with sufficient

confidence, i.e., whose maximum zero-shot probability falls below the 5th percentile (0.35, against a median of 0.92), are labeled as GT noise; the ten real classes form the cluster structure. We repeat everything at the 10th percentile (10% noise, threshold 0.46). Clusterings are computed on the 50 leading principal components of the CLIP embeddings with the grids described above.

The picture is the same (Table 18): ARI_f and $\mathrm{Purity}_{\mathrm{LNP\text{-}I}}$ certify degenerate everything-is-noise clusterings (99.9–100% predicted noise) with a perfect self-score of 1.000, while the noise-aware measures select the same non-degenerate HDBSCAN clustering ($k_{pred} = 9$) as the strong baselines. At 10% noise, $\mathrm{NMI}_{\mathrm{LNP\text{-}I}}$ additionally becomes unstable and selects a weak $k_{pred} = 3$ configuration ($\mathrm{NARI}_\kappa = 0.134$ where 0.377 is achievable). Notably, the flagged images are not geometric outliers in embedding space: their mean distance to the 10th nearest neighbor is 0.259, versus 0.264 for the remaining images. This noise is semantic ambiguity rather than low density, i.e., a genuinely different noise origin than held-out minority classes, and the measures behave identically under both. Changing the annotation source from a minority-class rule to a foundation model changes no conclusion.

Table 18: CIFAR-10 test set ($n = 10{,}000$, $k_{GT} = 10$) with foundation-model-annotated GT noise: images whose maximum CLIP (ViT-B/32) zero-shot class probability falls below the 5th (10th) percentile are labeled as GT noise. Clusterings are computed on the 50 leading principal components of the CLIP embeddings with the same DBSCAN and HDBSCAN grids as in Table 17; columns as there.

picking measure	selected configuration	k_{pred}	pred. noise	self-score	NARI_κ of pick	ARI_s of pick
CIFAR-10, 5% FM noise						
ARI_c	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.133	0.364	0.330
NMI_c	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.475	0.364	0.330
ARI_s	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.330	0.364	0.330
AMI_s	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.315	0.364	0.330
ARI_f	DBSCAN, minPts $= 3$, $\varepsilon = 0.500\,\varepsilon_k$	1	100.0%	1.000	0.000	0.000
ARI_p	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.356	0.364	0.330
$\mathrm{NMI}_{\mathrm{LNP\text{-}I}}$	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.567	0.364	0.330
$\mathrm{Purity}_{\mathrm{LNP\text{-}I}}$	DBSCAN, minPts $= 2$, $\varepsilon = 0.500\,\varepsilon_k$	7	99.9%	1.000	0.000	0.000
NARI_κ	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.364	0.364	0.330
NARI_φ	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.375	0.364	0.330
CIFAR-10, 10% FM noise						
ARI_c	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.132	0.377	0.343
NMI_c	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.476	0.377	0.343
ARI_s	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.343	0.377	0.343
AMI_s	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.333	0.377	0.343
ARI_f	DBSCAN, minPts $= 3$, $\varepsilon = 0.500\,\varepsilon_k$	1	100.0%	1.000	0.000	0.000
ARI_p	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.349	0.377	0.343
$\mathrm{NMI}_{\mathrm{LNP\text{-}I}}$	DBSCAN, minPts $= 16$, $\varepsilon = 1.000\,\varepsilon_k$	3	15.8%	0.631	0.134	0.117
$\mathrm{Purity}_{\mathrm{LNP\text{-}I}}$	DBSCAN, minPts $= 2$, $\varepsilon = 0.500\,\varepsilon_k$	7	99.9%	1.000	0.001	0.000
NARI_κ	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.377	0.377	0.343
NARI_φ	HDBSCAN, mcs$=43$, ms$=1$	9	50.9%	0.397	0.377	0.343

N Overview of External CVIs.

The metrics that are adapted from binary classification are applied to pairs of points. A pair is "positive" if both points are in the same cluster, and "negative" otherwise. The first measures are based on counting true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN).

- $\mathrm{RI} = \dfrac{TP + TN}{TP + FP + FN + TN}$ (Rand Index)
- $\mathrm{ARI} = \dfrac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{1 - \mathbb{E}[\mathrm{RI}]}$
- $\mathrm{FM} = \sqrt{\dfrac{TP}{TP + FP} \cdot \dfrac{TP}{TP + FN}}$ (Fowlkes-Mallows)
- $\mathrm{F1} = \dfrac{2TN}{2TN + FP + FN}$
- $\mathrm{J} = \dfrac{TN}{TN + FP + FN}$ (Jaccard index)
  - For F1 and J, our implementation takes the pairs that are separated in both labelings (TN) as the positive class, so both measure agreement on which pairs are separated. With the co-clustered pairs as the positive class (TP in place of TN), both would instead decrease with k_{pred} in the experiment of Figure 2, like FM, and would thus still violate (P1).
- $\mathrm{Purity} = \dfrac{1}{n} \sum_{y \in Y} \max_{x \in X} |x \cap y|$
  - The sum runs over the predicted clusters $y \in Y$ and the maximum over the GT clusters $x \in X$; n is the size of the dataset.
  - This performs poorly on unbalanced datasets, and is biased for more predicted clusters.
- $\mathrm{NMI} = \dfrac{2 \cdot (H(Y) - H(Y|X))}{H(Y) + H(X)}$ (arithmetic-mean normalization, the default of scikit-learn; Strehl & Ghosh (2002) normalize by the geometric mean)
  - $H(Y)$ is the entropy, $H(Y|X)$ is the conditional entropy.
- $\mathrm{AMI} = \dfrac{I(X;Y) - \mathbb{E}[I(X;Y)]}{\frac{1}{2}\big(H(X) + H(Y)\big) - \mathbb{E}[I(X;Y)]}$, with the expectation of the mutual information I under the permutation model (Vinh et al., 2009)

Our proposed CVIs are (ARI computed on the points labeled as cluster objects by both labelings, κ and φ on the binary noise indicators of all n points):

$$\mathrm{NARI}_\kappa = \frac{\#data \cdot \mathrm{ARI} + \#noise \cdot \kappa}{n}$$

$$\mathrm{NARI}_\varphi = \frac{\#data \cdot \mathrm{ARI} + \#noise \cdot \varphi}{n}$$

CVIs from the LNP-I paper:

- $\mathrm{Purity}_{\text{LNP-I}}$: Purity with filtered noise,
- $\mathrm{NMI}_{\text{LNP-I}}$: NMI with singleton noise, where ground-truth noise points are also given singleton labels in the prediction.

Finally, the different noise handling strategies:

- M_c: noise points are considered a **cluster**,
- M_s: each noise point is a **singleton** cluster,
- M_f: noise points are **filtered**, i.e. not included in clustering evaluation,
- M_p: noise points are filtered, and the score is multiplied by a **penalty** term.