Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Front Propagation–Based Clustering: A Density-Driven Graph Framework

Abdesslem Layeb

Abdesslem.layeb@univ-constantine2.dz

University of Constantine 2

Research Article

Keywords: Clustering, Front propagation, Graph-based clustering, Density-adaptive methods, Fast marching

Posted Date: August 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8948762/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Front Propagation–Based Clustering: A Density-Driven Graph Framework

Abdesslem Layeb

[1*]Computer science, University of Constantine 2, Ali mendjeli, Elkhroub, 25001, Constantine, Algeria.

Corresponding author(s). E-mail(s): Abdesslem.layeb@univ-constantine2.dz;

Abstract

Clustering complex data remains challenging in the presence of nonconvex structures, varying densities, and noise.In this paper, We propose a Front Propagation–Based Clustering (FP) framework that formulates cluster formation as a competitive propagation process over a neighborhood graph, governed by density-modulated speeds and arrival-time dynamics. The framework unifies two complementary algorithms: Adaptive Front Propagation Clustering (AFP), which propagates labels using locally adaptive speeds derived from neighborhood density, and Arrival-Time Front Propagation Clustering (ATFP), which assigns points based on exact shortest-path arrival times, yielding deterministic and geometrically consistent partitions. Unlike classical density-based clustering, cluster membership is determined by propagation dynamics rather than by thresholding density estimates. By modeling clusters as regions reached by competing propagation fronts, FP naturally handles arbitrary cluster shapes and noise without global optimization or sensitive thresholds. Extensive experiments on synthetic and real-world benchmark datasets demonstrate that the proposed methods consistently achieve high accuracy and robustness, outperforming or matching state-of-the-art clustering algorithms such as DBSCAN, k-means, and HDBSCAN. These results establish front propagation as an effective and interpretable paradigm for graph-based clustering.

Keywords: Clustering, Front propagation, Graph-based clustering, Density-adaptive methods, Fast marching

1 Introduction

Clustering is a basic problem in unsupervised learning. It tries to divide data into groups based on how similar the data is, without using labeled examples. It's a key tool in data mining, pattern recognition, statistics, and machine learning. It has many uses, such as image analysis, bioinformatics, social network analysis, and scientific data exploration. Since clustering became a subject of study, many algorithms have come out, like partition-based, hierarchical, spectral, and density-based methods. [1–3].

Partition-based clustering algorithms, among which k-means is the most representative [4], are widely used due to their simplicity and computational efficiency. However, k-means requires the number of clusters to be specified in advance, is highly sensitive to initialization and outliers, and implicitly assumes spherical, isotropic clusters. As a result, it performs poorly on datasets with non-convex structures, varying densities, or noisy distributions [5, 6]. Numerous extensions and optimization strategies have been proposed to mitigate these limitations [7], yet the fundamental geometric assumptions remain restrictive.

Hierarchical clustering can be split into agglomerative (bottom-up) and divisive (top-down) methods [8]. These methods create a dendrogram that shows how clusters are related, and they don't need the number of clusters to be set at the start. Even so, hierarchical methods tend to take a lot of computing power ($\mathcal{O}(n^2)$ or $\mathcal{O}(n^3)$) and can't handle non-convex cluster shapes very well, especially in big or noisy datasets.

Density-based clustering methods, like DBSCAN [9] and density-peak clustering [10], group data points into clusters based on areas with high concentrations of points, separated by sparser areas. These methods are good at finding clusters of any shape and at spotting noise. But, they depend a lot on picking the right density levels or correctly guessing local densities. It's not easy to pick good settings when clusters have different densities or when one cluster gradually turns into another. Doing this often means a lot of guessing or needing specific knowledge about the data.

Graph-based clustering provides another way to look at the problem. It turns data points into nodes on a graph and uses the graph's local structure to find clusters. Spectral clustering [11], for one, uses math to find complex cluster shapes that simple distance methods often miss. Still, it takes a while to run because of the math involved, and how you build the graph can throw things off. That makes it hard to use on very bigdatasets. Label propagation and similar diffusion methods [12] offer a simple, understandable way to spread information across the graph. Yet, they don't always have solid math backing, and the cluster borders they produce can be fuzzy or inconsistent.

Motivated by these limitations, we propose a novel graph-based clustering framework inspired by front propagation phenomena in physics. In the proposed approach, clustering is formulated as a competitive propagation process over a k-nearest neighbor graph, where multiple fronts expand simultaneously from a set of representative seed points. Propagation dynamics are governed by local data properties, allowing cluster boundaries to emerge naturally from the interaction of fronts rather than from global thresholds or static optimization criteria. Unlike classical density-based clustering, cluster membership in the proposed framework is determined by propagation dynamics rather than by thresholding density estimates.

The proposed framework integrates several desirable properties within a unified model: (i) local density adaptation without global parameters, (ii) efficient graph-based computation without eigendecomposition, (iii) flexibility to recover non-convex cluster shapes, and (iv) principled resolution of cluster boundaries through competitive front interactions. These properties enable robust clustering in the presence of heterogeneous densities and noise.

The framework comprises two complementary algorithms. Adaptive Front Propagation Clustering (AFP, Algorithm 1) uses greedy speed-priority expansion where propagation speed is modulated by local density, enabling rapid traversal through cluster cores while naturally slowing at boundaries. Arrival-Time Front Propagation Clustering (ATFP, Algorithm 2) reformulates clustering as a multi-source shortest-path problem, assigning points based on minimum cumulative arrival time. Using Dijkstra's algorithm, ATFP guarantees shortest-path optimal assignments under the induced density-modulated graph metric (given fixed seeds), providing improved stability and reproducibility.

Section 2 reviews related work on graph-based and propagation-based clustering. Section 3 details the proposed framework and algorithms. Experimental results are presented in Section 5, followed by concluding remarks in Section 9.

2 Related Work

Clustering remains a cornerstone of unsupervised learning, grouping data points by intrinsic similarities without labels, with broad applications in data mining, pattern recognition, computer vision, bioinformatics, and recommendation systems [13, 14]. Recent surveys highlight advances alongside persistent challenges like handling arbitrary shapes, varying densities, noise, scalability, and high dimensionality [13, 15].

Partition-based algorithms like k-means and variants optimize intra-cluster compactness via centroid updates, prized for simplicity and efficiency but demanding predefined cluster counts, suffering initialization sensitivity and outlier vulnerability, and assuming convex, isotropic shapes [14]. Metaheuristic enhancements boost robustness, though geometric limits hinder non-convex data; recent reviews taxonomy thousands of variants for big data [14].

Agglomerative and divisive hierarchical clustering create nested partitions that produce dendrograms without needing a preset number of clusters. These methods use linkage criteria (like Ward's or complete linkage). $\mathcal{O}(n^2)$ or greater complexity makes it hard to scale, and noise can worsen early mistakes [13]. Optimized structures like priority queues reduce complexity to $\mathcal{O}(n^2 \log n)$; extensions like CURE and BIRCH better handle shapes and densities [13]. Density-based approaches attempt to address some of these limitations by defining clusters as regions of high sample concentration separated by sparse areas. DBSCAN, in particular, is capable of identifying clusters with arbitrary shapes while explicitly labeling noise points. However, its reliance on global density parameters makes it sensitive to datasets containing clusters of varying densities. HDBSCAN extends the framework by incorporating hierarchical density estimation, improving adaptability in heterogeneous settings [13]. Density Peak Clustering (DPC) follows another intuition, selecting cluster centers based on local density

and relative distance. While elegant, DPC typically requires careful threshold selection and additional refinement steps. Recent studies have therefore focused on adaptive parameter estimation and geometric refinements, including voxel-based strategies and manifold-aware techniques for point cloud analysis [16, 17].

Graph techniques model similarities for non-convex clusters; spectral clustering uses Laplacian eigenvectors for partitioning but faces $\mathcal{O}(n^3)$ complexity and affinity sensitivity, with deep graph neural network integrations advancing representation [13, 18]. Label propagation offers scalable diffusion but lacks guarantees [19].

Propagation-based clustering methods, inspired by physical diffusion and wavefront phenomena, propagate density or influence through data neighborhoods to form clusters without predefined parameters. Fast marching and level-set methods solve Eikonal equations for front propagation in image segmentation, assuming uniform speed fields but adaptable to data-driven densities via local metrics [20]. Recent density propagation techniques, such as Density Propagation and Subcluster Merging (DPSM), compute node densities through iterative graph propagation using local neighborhoods, avoiding global distances, then partition into small subclusters (proven sound and complete) before spectral-inspired merging guided by the CluCut measure to automatically determine cluster numbers in both Euclidean and graph spaces [21]. Earlier adaptive approaches like DPAM employ three-stage density propagation—microcluster graph generation via affinity propagation, local merging via between-class margin and within-class cohesion, and regional density variance minimization—to handle multi-density data robustly without manual tuning [22].

Deep clustering methods co-optimize embeddings and assignments, excelling on images/text via autoencoders like DEC which minimizes KL divergence post-pretraining [23, 24]. Variants incorporate contrastive, graph, or self-supervision; surveys taxonomy module interactions, benchmarks, and applications, noting compute costs versus interpretability tradeoffs, with recent works pushing scalability for high dimensions [13, 15, 24].

2.1 Positioning of the Proposed Framework

Despite substantial advances in clustering, persistent limitations remain. Density-based methods such as DBSCAN and HDBSCAN rely on global or hierarchical thresholding, which may struggle with complex density variations [13]. Spectral approaches depend on graph construction and costly eigendecomposition, while classical propagation methods assume uniform or predefined speed fields that are not data-adaptive [20]. Deep clustering methods, although powerful in high-dimensional settings, require extensive training and offer limited interpretability for medium-scale geometric data [15].

Propagation-based approaches have recently been used for density *estimation*, but not for direct cluster *assignment*. For instance, DPSM [21] employs diffusion to compute smooth density fields, followed by separate partitioning and merging stages, resulting in a multi-step pipeline with additional parameters.

The proposed Front Propagation-Based Clustering framework departs from these paradigms by modeling cluster formation as competitive, density-modulated wavefront expansion on a k-nearest neighbor graph. Three aspects distinguish the proposed approach:

1. **Direct assignment via competitive propagation.** Cluster membership is determined inherently by propagation dynamics. Multiple fronts expand from seeds and interact, with boundaries emerging naturally from arrival-time competition rather than density thresholds or post-hoc merging.
2. **Density-modulated speed adaptation.** Propagation speed is locally defined as $v_i = f(\rho_i, \sigma)$, where ρ_i denotes local density. This induces acceleration within dense cluster cores and deceleration across sparse inter-cluster regions, creating adaptive soft barriers without explicit cutoffs.
3. **Unified greedy and optimal formulations.** The framework admits both a greedy implementation (AFP) and an arrival-time formulation (ATFP). The latter reduces to a multi-source shortest-path problem and yields assignments that are globally optimal with respect to the induced density-modulated graph metric, while remaining computationally efficient ($\mathcal{O}(n \log n)$).

By relying solely on local graph operations and priority-queue traversal, the proposed framework maintains computational efficiency and interpretability while accommodating irregular geometries and heterogeneous densities.

3 Front Propagation-Based Clustering Framework

This section introduces the proposed *Front Propagation-Based Clustering Framework*, a family of algorithms that model cluster formation as competitive front expansion on a data-driven neighborhood graph. The framework comprises two variants—Adaptive Front Propagation (AFP) and Arrival-Time Front Propagation (ATFP)—which share common density estimation, graph construction, and seed initialization procedures but differ in their propagation dynamics.

3.1 Motivation and Intuition

Traditional clustering methods often rely on static pairwise distances, global density thresholds, or predefined partitioning criteria, which can limit their ability to capture nonconvex geometries and heterogeneous density structures. Such approaches typically impose boundaries through global decisions rather than through local interactions.

In contrast, the proposed framework formulates clustering as a dynamic propagation process on a k-nearest neighbor graph. Starting from representative seed points, multiple fronts expand across the graph according to local density and connectivity. Each data point is assigned to the cluster whose front reaches it first, equivalently minimizing cumulative propagation influence along admissible paths.

Under this formulation, cluster boundaries arise naturally from competitive front interactions rather than from explicit density cutoffs or global partitioning rules. This dynamic mechanism enables coherent recovery of irregular shapes and varying densities within a unified propagation-based model.

3.2 Neighborhood Graph Construction

Given a dataset $X = \{x_i\}_{i=1}^n \subset \mathbb{R}^d$, we construct a k-nearest neighbor (k-NN) graph that serves as the underlying propagation structure for the proposed framework. Each data point corresponds to a node in the graph and is connected to its k nearest neighbors according to the Euclidean distance. In practice, this results in a directed neighborhood graph, reflecting local connectivity relationships in the data.

Beyond defining adjacency, the k-NN graph is also used to estimate local geometric and density-related statistics, including average neighbor distance and its dispersion. These quantities guide the propagation dynamics by modulating the effective speed at which fronts traverse the graph. By restricting propagation to local neighborhoods, the graph preserves fine-scale geometric structure while maintaining computational efficiency.

This graph-based representation provides a flexible and data-adaptive substrate for front propagation, enabling the proposed methods to capture nonconvex cluster geometries and varying densities without relying on global distance thresholds or predefined partitions.

3.3 Handling Unreachable Points in the Directed k-NN Graph

A directed k-NN graph does not guarantee full connectivity. Given a point x_i, the propagation front can only travel along outgoing edges $x_i \to x_j$ where $x_j \in \mathcal{N}_k(x_i)$. Consequently, two distinct failure modes may leave points unlabelled after propagation terminates:

- **Graph artifacts (unreachable points).** A point x_i may belong to a connected component that contains no seed, or no directed path from any seed may reach it, even though the graph *does* contain at least one edge pointing toward x_i. Such points are structurally reachable but were never visited.
- **True outliers.** A point x_i may be genuinely isolated from the rest of the data — no directed edge leads toward it, and its local neighbourhood is sparse. Its unlabelled status reflects a real property of the data rather than a graph artifact.

Conflating these two categories would produce incorrect cluster assignments and bias evaluation metrics such as ARI and NMI.

3.3.1 The Reverse-Neighbour Count

To distinguish between the two cases, we introduce the *reverse-neighbour count*:

$$r_i = \big|\{j : i \in \mathcal{N}_k(x_j)\}\big|, \tag{1}$$

that is, the number of points that list x_i among their k nearest neighbours. The quantity r_i is a formal property of the directed graph:

- $r_i > 0$ implies that at least one directed edge points *toward* x_i. If x_i is still unlabelled after propagation, its status is unambiguously a graph artifact — not true noise.
- $r_i = 0$ implies that no directed edge leads to x_i. Combined with low local density, this is strong evidence that x_i is a genuine outlier.

3.3.2 Three-Signal Noise Arbitration

After propagation terminates, every remaining unlabelled point x_i (label $\ell_i = -1$) is resolved by the following three-signal criterion. Let $\hat{\rho}$ denote the local density proxy $\hat{\rho}_i = 1/(\bar{d}_i + \varepsilon)$, where $\bar{d}_i$ is the mean distance to the k nearest neighbours of x_i and $\varepsilon > 0$ is a small constant. Define the density threshold

$$\tau = \text{Percentile}_p(\{\hat{\rho}_i\}_{i=1}^n)\,, \tag{2}$$

where $p \in (0, 100)$ is a user-controlled parameter (default $p = 30$). A point x_i is declared **true noise** if and only if *all three* of the following conditions hold simultaneously:

$$r_i = 0 \quad \text{AND} \quad \hat{\rho}_i < \tau \quad \text{AND} \quad \frac{1}{k} \sum_{j \in \mathcal{N}_k(x_i)} \hat{\rho}_j < \tau. \tag{3}$$

Otherwise, x_i is classified as an **unreachable graph artifact** and is assigned to the cluster of its nearest already-labelled neighbour:

$$\ell_i = \ell_{x^*}, \qquad x^* = \arg\min_{x_l \in \mathcal{L}} \|x_i - x_l\|_2, \tag{4}$$

where $\mathcal{L}$ denotes the set of labelled points *frozen before the resolution pass begins*, preventing transitive label contamination from sequentially resolved points.

The rationale for the conjunctive (AND) rule is conservative by design: we prefer to rescue a borderline point into a cluster over discarding it, since the primary cause of unlabelled points is graph topology, not data sparsity. Condition (3)-(i) is a necessary graph-theoretic precondition; conditions (ii) and (iii) operationalise the DBSCAN core-point criterion [**?**] at both the point level and the neighbourhood level.

3.3.3 Effect of k on Reachability

The parameter k directly governs the probability of disconnected components arising in the directed graph. Smaller values of k produce sparser graphs, increasing both the risk of isolated components and the number of points with $r_i = 0$. Larger values of k improve connectivity but may introduce spurious long-range edges that blur cluster boundaries. In practice, k is set according to the expected cluster granularity of the dataset (see Section 5). The arbitration procedure described above provides a safety net that is robust to moderate variation in k: points that become unreachable due to a slightly too-small k are recovered via Equation (4), while genuine outliers are preserved regardless of k.

Remark 1 The proposed criterion cannot reliably separate *embedded* outliers — points placed near cluster boundaries — from legitimate low-density cluster members, since embedded outliers typically satisfy $r_i > 0$ due to proximity to dense regions. This is a known fundamental limitation of local density-based noise detection and is left as future work.

3.4 Local Density Estimation and Propagation Speed

For each data point x_i, local geometric statistics are computed from its k-nearest neighbor neighborhood $\mathcal{N}_k(i)$. The average neighbor distance is defined as

$$\bar{d}_i = \frac{1}{k} \sum_{j \in \mathcal{N}_k(i)} \|x_i - x_j\|, \tag{5}$$

and the corresponding local density estimate is given by

$$\rho_i = \frac{1}{\bar{d}_i + \varepsilon}, \tag{6}$$

where ε is a small constant for numerical stability. In addition, the local distance dispersion

$$\sigma_i = \sqrt{\frac{1}{k} \sum_{j \in \mathcal{N}_k(i)} \left(\|x_i - x_j\| - \bar{d}_i\right)^2} \tag{7}$$

captures variability in the local neighborhood geometry. For numerical consistency, σ_i is normalized as $\sigma_i^{(n)} = \sigma_i / (\max_j \sigma_j + \varepsilon)$.

The propagation speed at each point is defined in the general form

$$v_i = \rho_i \, f(\sigma_i^{(n)}), \tag{8}$$

where $f(\cdot)$ modulates the influence of local dispersion. Several speed modulation strategies were investigated, including exponential, sigmoid, quadratic, inverse, and square-root attenuation functions:

$$f(\sigma) \in \left\{ 1,\ \exp(-\sigma),\ \frac{1}{1 + \exp(4(\sigma - 0.5))},\ \exp(-\sigma^2),\ \frac{1}{1+\sigma},\ \exp(-\sqrt{\sigma}) \right\}.$$

In this work, the baseline formulation $f(\sigma) = 1$ is adopted by default, yielding $v_i = \rho_i$. This choice provides a parameter-free and interpretable propagation model that performs consistently well across a wide range of datasets. The dispersion-aware variants are used only in scenarios involving extremely sparse sampling or heavy noise, where local distance variability can destabilize purely density-driven propagation. Importantly, these alternatives do not alter the core propagation mechanism but serve as optional robustness enhancements within the same unified framework.

3.5 Seed Selection Strategy

The quality of Front Propagation-based clustering strongly depends on the initial seed locations. The proposed framework employs a deterministic, density-aware seed selection strategy that promotes both representativeness and spatial diversity.

The first seed is chosen as the point with the maximum propagation speed. Subsequent seeds are selected iteratively by maximizing the following criterion:

$$\text{score}(x_j) = v_j \cdot \left(\min_{s \in \mathcal{S}} \|x_j - x_s\| \right)^{1.5}, \tag{9}$$

where $\mathcal{S}$ denotes the current set of selected seeds. This strategy favors points that are both locally dense and far from previously selected seeds, leading to stable and well-separated initial front propagations .

In addition to the proposed deterministic density–diversity criterion, the framework remains fully compatible with alternative seed initialization strategies. For instance, seeds may be selected using Density Peaks-based methods, K-means++ initialization, purely random selection, or farthest-point (max–min) sampling. Other density-aware or crowding-based heuristics can likewise be incorporated without modifying the core propagation mechanism. Since AFP and ATFP share identical propagation infrastructure, replacing the seed selection module does not alter the algorithmic dynamics but only affects the initial front configuration. This modularity enables fair comparison between initialization schemes and allows adaptation to application-specific requirements.

3.6 Front Propagation Mechanisms

Once the seeds are initialized, clusters are formed by propagating fronts over the k-NN graph. The two variants differ in how propagation is modeled.

3.6.1 Adaptive Front Propagation Clustering (AFP)

The baseline *Front Propagation Clustering (AFP)* algorithm models clustering as a competitive Front Propagation process over a neighborhood graph. Each cluster originates from a seed point and expands outward by iteratively propagating a wave whose strength is governed by local geometric density.

As described in Algorithm 1, a k-nearest neighbor graph is first constructed from the dataset. For each point x_i, a local density estimate is computed using the average distance to its k nearest neighbors. This density is then converted into a propagation speed v_i, such that points located in dense regions propagate faster than those in sparse areas. The propagation speed encodes the intuition that clusters should expand more rapidly through compact regions and slow down near boundaries.

Front Propagation initialization begins by selecting K representative seeds, each corresponding to a distinct cluster. Each seed emits an initial wave with strength equal to its local propagation speed. A max-priority queue is used to manage active front propagation, ensuring that, at every iteration, the point with the currently strongest wave is expanded first. This greedy expansion rule enforces a global competition between clusters.

When a Front Propagation reaches an unassigned point, that point is claimed by the corresponding cluster and inherits a propagation strength determined by its own local speed.If a point is encountered multiple times, it is assigned according to a greedy

priority rule based on local propagation speed, resolving conflicts implicitly during expansion. This mechanism yields sharp cluster separations without requiring explicit distance thresholds.

The AFP algorithm is computationally efficient, with complexity dominated by neighborhood construction and priority-queue operations. Its reliance on local density makes it particularly effective for datasets with well-separated, approximately homogeneous clusters. As a baseline method, AFP provides a strong reference point for evaluating the benefits of adaptive and arrival-time–based extensions introduced in subsequent variants.

3.6.2 Arrival-Time Front Propagation Clustering (ATFP)

ATFP reformulates front expansion as a minimum-arrival-time problem on the constructed k-NN graph. Building upon the previously defined propagation speeds v_i, the incremental travel time from node x_i to a neighboring node x_j is defined as

$$\Delta t_{ij} = \frac{\|x_i - x_j\|}{v_j}, \tag{10}$$

which induces a density-modulated edge cost.

Each seed initializes with zero arrival time. Propagation proceeds by iteratively selecting the node with the smallest current arrival time and relaxing its outgoing edges according to the above cost. This procedure is equivalent to a multi-source Dijkstra algorithm in which all seeds act as simultaneous sources.

A node is permanently assigned to the cluster whose front reaches it first, ensuring deterministic and conflict-free labeling. The resulting partition is globally optimal with respect to the induced density-modulated graph metric defined by Δt_{ij}.

Compared to AFP, ATFP replaces greedy strength-based competition with arrival-time optimality, eliminating order sensitivity and providing stable assignments across repeated runs. The computational complexity is $\mathcal{O}(n \log n)$ after graph construction, dominated by priority-queue operations.

3.7 Relationship Between Variants

The proposed framework comprises two complementary front propagation clustering algorithms that differ fundamentally in their expansion mechanism. Adaptive Front Propagation (AFP, Algorithm 1) employs a greedy, speed-priority strategy where clusters competitively expand into high-density regions based on local propagation strength. In contrast, Arrival-Time Front Propagation (ATFP, Algorithm 2) formulates clustering as a multi-source shortest-path problem, assigning points based on minimum arrival time through a density-modulated metric space. Both variants share identical infrastructure—neighborhood graph construction, local density estimation, and seed selection—ensuring that performance differences stem exclusively from their propagation dynamics. AFP prioritizes computational efficiency and aggressive capture of dense regions, while ATFP provides geodesic optimality and deterministic boundaries at the cost of higher computational complexity. This design enables

direct evaluation of the trade-off between greedy speed-based expansion and geodesic arrival-time optimization in front propagation clustering.

The behavioral difference between greedy speed-priority expansion (AFP) and cumulative arrival-time optimization (ATFP) is illustrated in Figure 1.

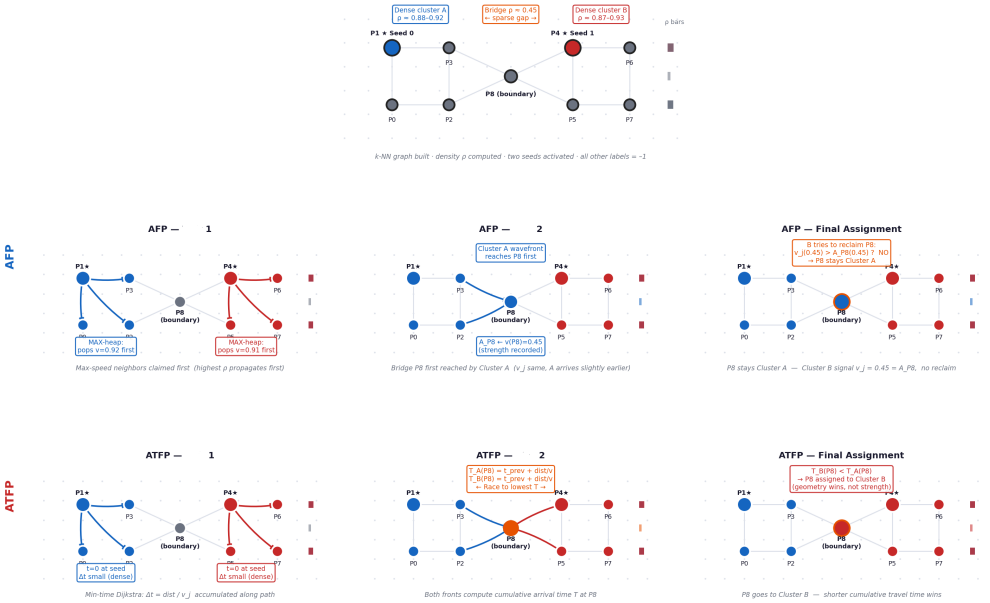


Fig. 1: Illustrative comparison of Adaptive Front Propagation (AFP) and Arrival-Time Front Propagation (ATFP) on a small k-NN graph. Two dense clusters are connected by a sparse bridge node. AFP assigns the boundary node based on greedy speed-priority expansion, whereas ATFP assigns it according to minimum cumulative arrival time under a density-modulated metric. The example highlights the difference between competitive local-speed propagation and geodesic shortest-path optimization.

The key algorithmic differences between AFP and ATFP are detailed in Table 1, which highlights their distinct initialization strategies, priority queue mechanisms, relabeling policies, and value propagation rules.

3.8 Computational Complexity

Let n denote the number of data points and k the number of nearest neighbors. Constructing the k-nearest neighbor graph requires $\mathcal{O}(nk \log n)$ time using standard exact or approximate neighbor search methods.

For *Front Propagation Clustering (AFP)*, Front Propagation is implemented using a max-priority queue. Each data point is processed at most once per neighbor, resulting in a total time complexity of $\mathcal{O}(nk \log n)$ due to priority queue insertions and extractions.

Table 1: Key Algorithmic Differences Between ATFP and AFP

Line	Feature	ATFP (Arrival-Time)	AFP (Adaptive)
7	Initialization	$T_i \leftarrow +\infty$ for all points	$A_i \leftarrow 0$ for all points
9	Priority queue	Min-heap (earliest arrival)	Max-heap (highest speed)
14	Expansion order	Pop point with minimum arrival time t	Pop point with maximum speed v
18–20	Relabeling policy	No relabeling: skip if $\ell_j \neq -1$	Relabeling allowed: if $v_j > A_j$
21–22	Value propagation	Cumulative: $t' = t + \frac{\|x_i - x_j\|}{v_j}$	Local: $A_j = v_j$
23	Update condition	If $t' < T_j$ (Dijkstra relaxation)	If $\ell_j = -1$ or $v_j > A_j$

Note: Line numbers refer to Algorithms 2 and 1.

Arrival-Time Front Propagation Clustering (ATFP) formulates propagation as a multi-source shortest-path problem on the k-NN graph. Its computational complexity is equivalent to Dijkstra's algorithm on a sparse graph, yielding $\mathcal{O}(nk \log n)$ time complexity.

Overall, both variants scale linearly with the number of edges in the neighborhood graph and exhibit comparable asymptotic complexity, differing primarily in constant factors arising from their respective propagation rules.

3.9 Algorithm

Algorithm 1 Adaptive Front Propagation Clustering (AFP)

Require: Dataset $X = \{x_i\}_{i=1}^{n}$, number of clusters K, neighbours k, density percentile p
Ensure: Cluster labels ℓ_i

Build directed k-NN graph of X
for each point x_i **do**
    Compute mean neighbour distance $\bar{d}_i$
    Compute density $\rho_i \leftarrow 1/(\bar{d}_i + \varepsilon)$
    Set propagation speed $v_i \leftarrow \rho_i$
end for
Select K seeds $\mathcal{S}$ using density and spatial diversity ▷ Section **??**
Initialise labels $\ell_i \leftarrow -1$ for all i
Initialise arrival strengths $A_i \leftarrow 0$ for all i
Initialise max-priority queue $\mathcal{Q} \leftarrow \emptyset$
for each seed $s \in \mathcal{S}$ with cluster id c **do**
    $\ell_s \leftarrow c, \quad A_s \leftarrow v_s$
    Push (v_s, s, c) into $\mathcal{Q}$
end for
while $\mathcal{Q} \neq \emptyset$ **do**
    Pop $(v, i, c) \leftarrow$ entry with maximum v from $\mathcal{Q}$
    for each neighbour $j \in \mathcal{N}_k(i)$ **do**
        if $\ell_j = -1$ **or** $v_j > A_j$ **then**
            $\ell_j \leftarrow c, \quad A_j \leftarrow v_j$
            Push (v_j, j, c) into $\mathcal{Q}$
        end if
    end for
end while
// Phase 4 (optionnal): resolve points left unlabelled by graph topology
Compute reverse-neighbour counts: $r_i \leftarrow |\{j : i \in \mathcal{N}_k(x_j)\}|$ for all i ▷ Eq. 1
Compute density threshold $\tau \leftarrow \text{Percentile}_p(\{\rho_i\})$ ▷ Eq. 2
Freeze labeled set $\mathcal{L} \leftarrow \{x_i \mid \ell_i \neq -1\}$
for each unlabelled point x_i **where** $\ell_i = -1$ **do**
    if $r_i = 0$ **and** $\rho_i < \tau$ **and** $\frac{1}{k}\sum_{j \in \mathcal{N}_k(i)} \rho_j < \tau$ **then**
        $\ell_i \leftarrow -1$ ▷ True outlier — retain as noise
    else
        $x^* \leftarrow \arg\min_{x_l \in \mathcal{L}} \|x_i - x_l\|_2$
        $\ell_i \leftarrow \ell_{x^*}$ ▷ Graph artifact — assign to nearest cluster
    end if
end for
Shift predicted labels: $\ell_i \leftarrow \ell_i + 1$ for all $\ell_i \geq 0$ ▷ Avoid aliasing with noise label 0
return $\{\ell_i\}$

Algorithm 2 Arrival-Time Front Propagation Clustering (ATFP)

Require: Dataset $X = \{x_i\}_{i=1}^{n}$, number of clusters K, neighbours k, density percentile p
Ensure: Cluster labels ℓ_i

Build directed k-NN graph of X
for each point x_i **do**
    Compute mean neighbour distance $\bar{d}_i$
    Compute density $\rho_i \leftarrow 1/(\bar{d}_i + \varepsilon)$
    Set propagation speed $v_i \leftarrow \rho_i$
end for
Select K density-based spatially diverse seeds $\mathcal{S}$
Initialise arrival times $T_i \leftarrow +\infty$ for all i
Initialise labels $\ell_i \leftarrow -1$ for all i
Initialise min-priority queue $\mathcal{Q} \leftarrow \emptyset$
for each seed $s \in \mathcal{S}$ with cluster id c **do**
    $T_s \leftarrow 0, \quad \ell_s \leftarrow c$
    Push $(0, s, c)$ into $\mathcal{Q}$
end for
while $\mathcal{Q} \neq \emptyset$ **do**
    Pop $(t, i, c) \leftarrow$ entry with minimum t from $\mathcal{Q}$
    if $t \neq T_i$ **then**
        continue ▷ Stale entry — skip
    end if
    for each neighbour $j \in \mathcal{N}_k(i)$ **do**
        if $\ell_j \neq -1$ **then**
            continue ▷ Already labelled
        end if
        $\Delta t \leftarrow \|x_i - x_j\|/(v_j + \varepsilon)$
        $t' \leftarrow t + \Delta t$
        if $t' < T_j$ **then**
            $T_j \leftarrow t', \quad \ell_j \leftarrow c$
            Push (t', j, c) into $\mathcal{Q}$
        end if
    end for
end while
// Phase 4 (optionnal): resolve points left unlabelled by graph topology
Compute reverse-neighbour counts: $r_i \leftarrow |\{j : i \in \mathcal{N}_k(x_j)\}|$ for all i ▷ Eq. 1
Compute density threshold $\tau \leftarrow \text{Percentile}_p(\{\rho_i\})$ ▷ Eq. 2
Freeze labelled set $\mathcal{L} \leftarrow \{x_i \mid \ell_i \neq -1\}$
for each unlabelled point x_i **where** $\ell_i = -1$ **do**
    if $r_i = 0$ **and** $\rho_i < \tau$ **and** $\frac{1}{k}\sum_{j \in \mathcal{N}_k(i)} \rho_j < \tau$ **then**
        $\ell_i \leftarrow -1$ ▷ True outlier — retain as noise
    else
        $x^* \leftarrow \arg\min_{x_l \in \mathcal{L}} \|x_i - x_l\|_2$
        $\ell_i \leftarrow \ell_{x^*}$ ▷ Graph artifact — assign to nearest cluster
    end if
end for
Shift predicted labels: $\ell_i \leftarrow \ell_i + 1$ for all $\ell_i \geq 0$ ▷ Avoid aliasing with noise label 0
return $\{\ell_i\}$

4 Distinction from Existing Density-, Front Propagation-, and Graph-Based Clustering Methods

Clustering approaches based on density estimation, region expansion, and graph representations have been extensively investigated. While the proposed Adaptive Front Propagation (AFP) and Arrival-Time Front Propagation (ATFP) algorithms share certain structural elements with these families, their formulation differs fundamentally in how clustering is modeled and executed. The proposed framework introduces a unified density-modulated, multi-front dynamical system that departs conceptually and operationally from prior methods.

4.1 Relation to Classical Front Propagation and Region-Growing Methods

Classical region-growing techniques [25] and Fast Marching methods [26] typically rely on:

- Fixed similarity criteria or distance thresholds,
- Uniform or externally defined speed fields,
- Sequential or single-front expansion strategies.

Such formulations are primarily geometric and often assume homogeneous propagation dynamics. In heterogeneous or variable-density data, this can lead to boundary leakage in sparse regions or fragmentation in dense areas.

In contrast, AFP and ATFP introduce three key conceptual extensions:

- **Density-Modulated Propagation**: Propagation speed is derived directly from local geometric structure ($v_i = \rho_i$), causing expansion to adapt continuously to data density. Sparse regions naturally slow propagation, forming implicit soft barriers between clusters.
- **Simultaneous Multi-Front Competition**: All clusters expand concurrently within a shared priority structure. Cluster boundaries emerge from competitive front interactions rather than sequential assignment or threshold-based stopping rules.
- **Dual Propagation Formulations**: AFP implements a greedy, speed-priority expansion mechanism emphasizing computational efficiency, whereas ATFP formulates clustering as a multi-source shortest-path process under a density-modulated metric, yielding geodesic-consistent assignments.

This formulation transforms front propagation from a fixed geometric procedure into a data-adaptive dynamical process in which local structure governs expansion behavior.

4.2 Relation to Graph-Based Clustering Methods

Graph-based clustering techniques, including spectral clustering [11, 27], minimum spanning tree approaches, and modularity-based community detection [28, 29], identify clusters through global analysis of graph structure. Typical strategies involve

eigen-decomposition of graph Laplacians, cut optimization, or connectivity-based partitioning.

AFP and ATFP utilize graph structure in a fundamentally different manner:

- **Local Rather Than Global Graph Analysis**: The k-NN graph is used solely to provide local neighborhood structure during propagation. No global spectral decomposition, cut optimization, or modularity maximization is performed.
- **Dynamic Evolution Instead of Static Partitioning**: Rather than analyzing a fixed graph to derive partitions, clusters emerge from temporal propagation dynamics over the graph.
- **Emergent Boundaries Instead of Explicit Optimization**: Cluster boundaries are not obtained by solving a global objective (e.g., min-cut), but arise naturally from competitive front interactions governed by density-modulated speed.
- **Scalable Priority-Based Computation**: The framework relies primarily on priority queue operations, avoiding large-scale matrix factorization steps typically required in spectral approaches.

This local-temporal perspective allows clusters to form through adaptive propagation dynamics while retaining the structural advantages of graph-based representations.

4.3 Conceptual Contributions

The novelty of the AFP/ATFP framework lies in its unified dynamical formulation rather than in isolated algorithmic components. While elements such as k-NN graphs, density estimation, or shortest-path propagation have appeared independently in prior work, their integration into a competitive multi-front clustering framework with density-modulated speed has not been previously formulated in this manner.

The principal contributions can be summarized as follows:

- **Dynamic Clustering Formulation**: Clustering is modeled as competitive multi-front propagation over time, contrasting with static density thresholding or global graph partitioning approaches.
- **Intrinsic Density Control**: Local geometric density directly governs propagation speed, embedding structural information into the core dynamics rather than applying density as a post-processing criterion.
- **Complementary Greedy and Geodesic Variants**: The framework provides both a computationally efficient speed-priority strategy (AFP) and a geodesic arrival-time formulation (ATFP), enabling analysis of efficiency–optimality trade-offs within a unified setting.
- **Local Graph-Driven Dynamics**: The k-NN graph serves as a structural substrate for propagation without requiring global spectral or combinatorial optimization.

Together, these design choices yield a clustering paradigm in which boundaries emerge from adaptive front interactions shaped by local density, enabling effective handling of non-convex structures and heterogeneous sampling while maintaining favorable computational complexity.

4.4 Theoretical Foundations

The proposed Front Propagation–Based Clustering Framework is grounded in well-established concepts from graph theory, shortest-path optimization, and front propagation. By interpreting clustering as a competitive expansion process over a neighborhood graph, the framework provides a principled connection between local density estimation and global cluster formation.

Graph-Based Propagation Model.

Let $G = (V, E)$ denote the k-nearest neighbor graph constructed from the dataset X, where each vertex corresponds to a data point and edges encode local geometric proximity. Front propagation in both AFP and ATFP is restricted to this graph, ensuring that cluster growth respects the local connectivity structure of the data rather than relying on Euclidean distances in the ambient space. This formulation connects the framework to graph-based and label propagation methods while incorporating temporal propagation dynamics.

Density-Modulated Propagation Speed.

Each data point x_i is associated with a local propagation speed v_i, defined as the inverse of its average neighbor distance. This choice establishes a direct link between density and propagation dynamics: dense regions induce faster wave expansion, while sparse regions slow down propagation. As a result, cluster interiors are traversed rapidly, whereas cluster boundaries naturally emerge in low-density transition zones. This mechanism plays a role analogous to density-adaptive metrics used in diffusion and spectral clustering, but is embedded directly into the propagation process.

Competitive Front Expansion.

Cluster formation is driven by the simultaneous propagation of multiple fronts originating from distinct seed points. In AFP, competition is resolved through a greedy priority mechanism: points are assigned assigned to clusters according to the priority of propagation determined by local speed values, and can be relabeled if a faster front arrives subsequently. In ATFP, competition is resolved through cumulative arrival times computed via Dijkstra's algorithm, yielding globally optimal assignments under the density-modulated metric. In both cases, the shared priority queue ensures that all fronts compete simultaneously, resulting in conflict-free labeling where each point belongs to exactly one cluster.

Connection to Front Propagation and Fast Marching.

The arrival-time formulation can be interpreted as a graph-based discretization of continuous front propagation models, such as the Fast Marching Method [26]. In the continuous setting, fronts evolve according to the Eikonal equation $|\nabla T| = 1/F(x)$, where $F(x)$ is a spatially varying speed field. ATFP provides a discrete analogue by restricting propagation to the k-NN graph and using Dijkstra's algorithm to compute arrival times under the density-modulated metric $c_{ij} = \|x_i - x_j\|/v_j$. While the k-NN graph approximates the underlying data manifold, convergence to the continuous

solution depends on neighborhood density and graph connectivity. This perspective explains the framework's ability to capture complex, non-convex cluster geometries through geodesic distance computations on the local graph structure.

Completeness and Determinism of Assignments.

Given fixed seeds, both AFP and ATFP guarantee that every reachable data point is assigned to exactly one cluster:

AFP: Points are assigned to the cluster whose front reaches them with highest local speed. While the algorithm is deterministic for a given tie-breaking rule in the priority queue, the greedy nature means assignments can be order-dependent when multiple fronts arrive with similar speeds.

ATFP: Points are assigned to the cluster with minimum arrival time. Dijkstra's algorithm ensures deterministic, reproducible results. In the rare case of tied arrival times (equidistant points), assignment depends on the deterministic tie-breaking order of the priority queue implementation.

Both algorithms ensure complete coverage (all points assigned) and conflict-free labeling (no point belongs to multiple clusters), which are essential properties for practical clustering applications.

5 Experimental Study

This section presents a comprehensive empirical evaluation of the proposed Adaptive Front Propagation–Based clustering methods. The study is designed to rigorously assess clustering accuracy, robustness to complex geometries and density variations, and competitiveness against widely used state-of-the-art algorithms.

5.1 Benchmark Datasets

We evaluated the proposed Adaptive Front Propagation Clustering (AFP) and Arrival-Time Front Propagation Clustering (ATFP) on 34 challenging datasets from the benchmark suite [30][1]. The suite comprises a diverse collection of 2D synthetic datasets with ground-truth labels and several classic real-world UCI datasets. The synthetic datasets are specifically designed to test clustering algorithms on a wide range of difficulties, including:

- large numbers of well-separated Gaussian clusters (a1_k20, a2_k35, a3_k50),
- complex non-convex and manifold structures (flame_k2, spiral_k3, chainlink_k2, lsun_k3, labirynth_k6, parabolic_k2, wingnut_k2, target_k2),
- varying cluster densities and sizes (aggregation_k7, compound_k6, unbalance_k8),
- interlocking or touching clusters (jain_k2, pathbased_k3, twodiamonds_k2),
- heavy noise and outliers (ring_noisy_k2, ring_outliers_k5),
- linear or striped separable structures (line_k2, stripes_k2, cross_k4, square_k2, windows_k5, trapped_lovers_k3).

[1] https://clustering-benchmarks.gagolewski.com/weave/suite-v1.html

The real-world datasets (iris_k3, wine_k3, glass_k6, wdbc_k2, sonar_k2) provide additional validation on higher-dimensional, non-synthetic data with overlapping classes and noise.

This selection enables a comprehensive assessment of the methods' ability to handle arbitrary cluster shapes, heterogeneous densities, noise robustness, and scalability across different problem characteristics. Visualizations and detailed descriptions of all datasets are available at the benchmark repository website.

Table 2 summarizes key dataset characteristics, including the number of points (n), dimensions (d), reference labels, number of clusters (k), and noise fraction (g). Detailed per-dataset information is provided in the repository site.

5.2 Compared Methods

The proposed methods are compared against a representative set of classical and modern clustering algorithms:

- **K-Means** [4]
- **Gaussian Mixture Models (GMM)** [31]
- **Agglomerative Clustering (Average linkage)** (AggAverage) [32]
- **Ward Agglomerative Clustering** (AggWard) [33]
- **Local Density Peaks based MST Clustering (LDP-MST)** [34]
- **MST Clustering Algorithm Based on Density Core (MST-DC)** [35]
- **Spectral Clustering** [36]
- **Mean Shift** [37]
- **HDBSCAN** [38]
- **Adaptive Front Propagation Clustering (AFP)** *(proposed)*
- **Arrival-Time Front Propagation Clustering (ATFP)** *(proposed)*

This selection includes centroid-based, density-based, graph-based, and hierarchical clustering paradigms, ensuring a comprehensive comparison.

5.3 Parameter Settings

To ensure a fair and reproducible evaluation, standard parameter settings are adopted for all baseline methods. Unless otherwise stated, default parameters from the corresponding libraries are used.

For the proposed methods, the number of nearest neighbors is fixed to $k = 15$ across all datasets for both variants AFP and ATFP. This value was selected as a moderate neighborhood size that preserves local geometric structure while ensuring sufficient graph connectivity. Importantly, the same setting is used for all datasets to avoid performance inflation through dataset-specific tuning.

The parameters of competing methods are set as follows:

- **K-Means**: $n_init = 10$, random initialization
- **MST-DC** :$k - nearestneighbors = 20$
- **LDP-MST**: $k - nearestneighbors = 20$
- **GMM**: full covariance, random initialization
- **Spectral Clustering**: nearest-neighbor affinity with $k = 10$

Table 2: Summary of key characteristics of the evaluated datasets from the benchmark suite [30]. Columns: number of points (n), dimensions (d), number of clusters (k, primary reference labeling), noise points (if explicitly designated), and Gini index g of cluster sizes (higher values indicate greater imbalance, often correlated with noise or small clusters).

Dataset	n	d	k	Noise points	g
a1_k20	3000	2	20	0	0
a2_k35	5250	2	35	0	0
a3_k50	7500	2	50	0	0
aggregation_k7	788	2	7	0	0.45
compound_k6	399	2	6	0	0.44
flame_k2	240	2	2	0	0.28
jain_k2	373	2	2	0	0.48
spiral_k3	312	2	3	0	0.02
pathbased_k3	300	2	3	0	0.06
chainlink_k2	1000	3	2	0	0
hepta_k7	212	3	7	0	0.01
tetra_k4	400	3	4	0	0
twodiamonds_k2	800	2	2	0	0
wingnut_k2	1016	2	2	0	0
target_k2	770	2	2	12	0.04
lsun_k3	400	2	3	0	0.25
labirynth_k6	3546	2	6	0	0.5
parabolic_k2	1000	2	2	0	0.02
line_k2	250	2	2	0	0.6
ring_outliers_k5	1030	2	5	0	0.71
unbalance_k8	6500	2	8	0	0.63
iris_k3	150	4	3	0	0
wine_k3	178	13	3	0	0.13
glass_k6	214	9	6	0	0.48
wdbc_k2	569	30	2	0	0.25
sonar_k2	208	60	2	0	0.07
stripes_k2	5000	2	2	0	0
isolation_k3	9000	2	3	0	0
cross_k4	2000	2	4	0	0
trapped_lovers_k3	5000	3	3	0	0.4
square_k2	1000	2	2	0	0
windows_k5	2977	2	5	0	0.58
ring_noisy_k2	1050	2	2	43	0
ring_k2	1000	2	2	0	0

- **HDBSCAN**: min_cluster_size = 20, min_samples = 10
- **Agglomerative (Average)**: average linkage
- **Ward**: Ward linkage
- **Mean Shift**: bandwidth estimated automatically

No dataset-specific parameter tuning is performed for any method.

5.4 Evaluation Metrics

Clustering performance is assessed using three widely adopted external validity measures that quantify the agreement between predicted cluster labels and the ground-truth partitioning. For all metrics, higher values indicate better agreement, with 1 corresponding to perfect recovery and values near 0 expected under random labeling.

- **Adjusted Rand Index (ARI)** [39]: a chance-corrected version of the Rand index [40] that evaluates pairwise agreement between cluster assignments. ARI is particularly sensitive to over-segmentation and boundary errors, thereby providing a stringent measure of exact cluster membership recovery.
- **Adjusted Mutual Information (AMI)** [41]: an information-theoretic measure that quantifies the mutual dependence between predicted and reference partitions, normalized and corrected for chance based on entropy.
- **V-Measure** [42]: the harmonic mean of homogeneity and completeness, equivalent to mutual information normalized by the arithmetic mean of cluster and class entropies.

All three metrics are bounded in $[-1, 1]$ (although negative values are uncommon in practice) and are widely regarded as reliable and complementary external clustering validity criteria.

5.5 Experimental Protocol

Each algorithm is applied independently to all 34 datasets using the parameter settings specified in Section 5. For methods involving random initialization (e.g., K-Means, GMM, Spectral Clustering), a fixed random seed is used to ensure reproducibility.

The proposed AFP and ATFP methods are deterministic given the constructed neighborhood graph and the adopted seed selection strategy. Runtime measurements are conducted under a consistent hardware and software environment to ensure fair computational comparisons. We used the script provided by the benchmark author to evaluate all algorithms simultaneously.

Fair comparison with HDBSCAN.

HDBSCAN is a noise-aware algorithm that deliberately withholds cluster assignment from points it identifies as outliers, producing label -1 for those points as a direct output of its density-connectivity criterion. It does not apply any post-propagation rescue step to recover unlabelled points. To ensure a fair comparison, **AFP and ATFP are evaluated in their base configuration — without the post-propagation unreachable-point handling described in Algorithm 1 (Phase 4)** — when benchmarked against HDBSCAN. In this configuration, any point left unlabelled after

propagation retains label -1 and is treated as noise by the evaluation metrics, exactly as HDBSCAN treats its noise points. This ensures that neither AFP nor ATFP benefits from an asymmetric post-processing advantage over HDBSCAN in the primary comparison.

The full AFP and ATFP variants *with* unreachable-point handling are reported separately in Section 7.1, where the contribution of Phase 4 is isolated and quantified against the base variants. This two-stage reporting allows the reader to distinguish the contribution of the core propagation mechanism from the contribution of the post-propagation arbitration step.

5.6 Results and Analysis

5.7 Discussion of ARI Results

The Adjusted Rand Index (ARI) results summarized in table 3 reveal clear performance patterns across the diverse geometric and topological challenges in the benchmark suite.

On well-separated, convex datasets (e.g., *a1_k20–a3_k50*, *hepta_k7*, *tetra_k4*, *twodiamonds_k2*, *unbalance_k8*), nearly all methods achieve near-perfect ARI. Centroid-based (K-Means, GMM) and hierarchical approaches (AggAverage, AggWard) excel here due to the homogeneous, spherical nature of the clusters. The proposed Adaptive Front Propagation Clustering (AFP) and Arrival-Time Front Propagation Clustering (ATFP) match this performance without degradation, confirming that the propagation mechanism introduces no unnecessary overhead on simple structures.

Performance gaps widen substantially on non-convex, manifold, and topologically complex datasets (e.g., *flame_k2*, *labirynth_k6*, *parabolic_k2*, *chainlink_k2*, *wingnut_k2*, *target_k2*, *lsun_k3*, *ring_k2*, *ring_noisy_k2*). Centroid-based and classical hierarchical methods typically fail, merging distinct manifolds or ignoring curved boundaries. Spectral Clustering improves via global graph structure but remains highly sensitive to k-NN graph construction and eigenvalue selection, leading to inconsistent results across varying geometries.

Density Peaks–based MST methods (LDP-MST, MST-DC) capture local density better yet suffer from well-known limitations: strong sensitivity to density estimation errors and cutoff parameter choices, especially in heterogeneous or noisy regions (e.g., lower scores on *flame*, *parabolic*, *ring_noisy*).

In contrast, the proposed AFP and ATFP consistently achieve near-perfect or perfect ARI on these challenging datasets—both reach 1.0 on *chainlink*, *wingnut*, *target*, and *trapped_lovers*; AFP attains 0.9986 on *labirynth* and 0.9565 on *parabolic*; both exceed 0.98 on noisy rings. This superiority stems from the propagation paradigm's reliance on competitive geometric expansion from density-aware seeds, which naturally follows local connectivity and curved boundaries without requiring explicit density peak detection, global eigen-decomposition, or sensitive cutoff thresholds.

On thin or highly interleaved manifolds (e.g., *spiral_k3*, *pathbased_k3*, *stripes_k2*), HDBSCAN dominates due to its sophisticated hierarchical density modeling and explicit noise rejection, though at the cost of heavier assumptions about density reachability and hierarchy. AFP and ATFP remain competitive, frequently outperforming

centroid, hierarchical, spectral, and MST-based methods, and achieve perfect recovery on datasets like *stripes* and *trapped_lovers*.

In noisy or inherently ambiguous cases (e.g., *sonar_k2*, *glass_k6*, *ring_outliers_k5*), low ARI across all methods reflects limited separability. Here, ATFP often proves more robust than AFP (e.g., higher stability on *labirynth*, *parabolic*, *windows*), as arrival-time minimization regularizes propagation and prevents over-expansion in low-density areas.

Overall, the proposed front propagation framework—particularly AFP and ATFP—excels on geometrically complex, non-convex, and manifold-structured data where traditional and many modern baselines falter. Unlike Density Peaks methods (sensitive to density estimation and cutoffs), Spectral Clustering (dependent on graph construction and eigenvalues), and HDBSCAN (relying on strong density hierarchy assumptions), the proposed approach offers a lightweight, geometrically intuitive alternative that leverages local propagation dynamics and competitive front expansion, achieving state-of-the-art recovery of intricate topological structures with minimal parameterization.

Statistical comparison via the Friedman test across all 34 datasets (Figure 2) confirms the strong overall performance of the proposed methods. AFP achieves the best mean rank (3.76), followed closely by ATFP (3.91) and Spectral Clustering (3.96), placing them as the top three performers across the benchmark suite. In contrast, MeanShift shows the poorest aggregate performance (mean rank 7.22) and is the only algorithm statistically significantly different from AFP (mean rank difference 3.46, $p < 0.05$). HDBSCAN (4.28), GMM (4.99), K-Means (5.26), and hierarchical methods (AggWard 5.76, AggAverage 5.85) occupy intermediate positions.

These aggregate rankings reflect the proposed methods' balanced performance profile: while centroid-based and hierarchical approaches excel on well-separated clusters but fail on complex geometries, and density-based methods like HDBSCAN specialize in specific manifold structures, AFP and ATFP maintain consistently high performance across both simple and complex scenarios. The statistical superiority over MeanShift is particularly notable, as MeanShift's mode-seeking approach struggles with sparse regions and non-convex boundaries—precisely the challenges where propagation-based methods excel.

The competitive performance relative to Spectral Clustering (mean rank difference 0.20, not statistically significant) is noteworthy. While Spectral Clustering achieves strong results on many datasets through global graph analysis, it occasionally fails on intricate topologies (e.g., 0.7189 on *labirynth* versus AFP's 0.9986, 0.5034 on *aggregation* versus AFP's 0.9116). The proposed front propagation framework offers comparable aggregate performance with greater consistency on geometrically challenging datasets and reduced sensitivity to graph construction parameters.

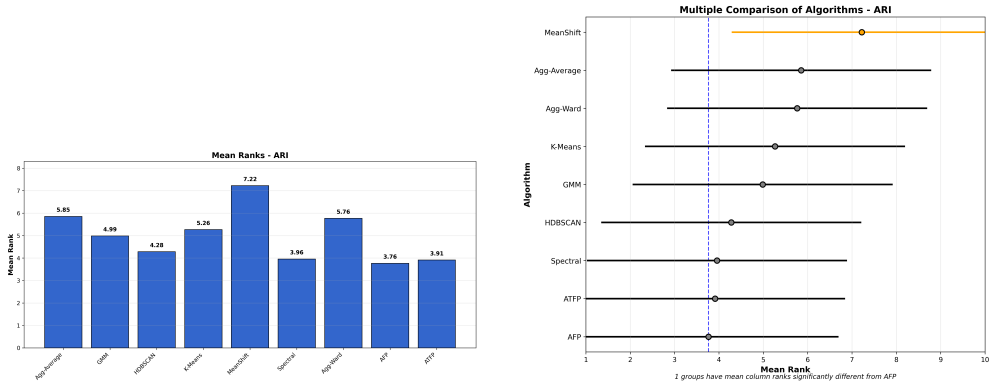
5.8 Discussion of AMI Results

Adjusted Mutual Information (AMI) results summarized in table 4 evaluates the information shared between predicted and ground-truth labels while penalizing random agreement. Unlike ARI, AMI is less sensitive to local boundary errors and more reflective of global labeling consistency. On high-separation datasets (e.g., *a1_k20*,

Table 3: Adjusted Rand Index (ARI) scores for various clustering algorithms.

Dataset	AggAverage	GMM	HDBSCAN	K-Means	LDP-MST	MST-DC	MeanShift	Spectral	AggWard	**AFP**	**ATFP**
a1_k20	0.9251	0.8798	0.6775	**0.9663**	0.9008	0.959	0.0832	0.9475	0.9142	0.9492	0.9532
a2_k35	0.9337	0.9271	0.624	**0.9673**	0.7186	0.9617	0.0551	0.9569	0.9223	0.9565	0.9597
a3_k50	0.9435	0.9166	0.5937	0.9457	0.5699	**0.9669**	0.0361	0.9635	0.9374	0.9618	0.9627
aggregation_k7	**1**	0.9475	0.8089	0.7622	0.9848	0.992	0.372	0.5984	0.7948	0.9116	0.7565
compound_k6	**0.803**	0.5963	0.7662	0.5379	0.5342	0.5278	0.4365	0.4969	0.5518	0.5677	0.5516
flame_k2	0.4422	0.4202	0.5832	0.4534	0.8227	**0.9832**	0.4763	0.388	0.1872	0.9387	0.6302
jain_k2	0.7792	-0.0045	0.9205	0.3241	0.5853	0.0408	0.0247	**1**	0.5146	-0.0573	0.0125
spiral_k3	-0.0023	-0.0045	**0.5537**	-0.006	0.0004	0.0298	0	0.3715	0.0018	0.3595	0.2065
pathbased_k3	0.4436	0.436	**0.7703**	0.4613	0.4868	0.3966	0.4588	0.5134	0.4847	0.4169	0.4594
chainlink_k2	0.2719	0.91	**1**	0.0927	0.0631	**1**	0	**1**	0.2803	**1**	**1**
hepta_k7	**1**	**1**	**1**	**1**	**1**	**1**	0	0.5118	**1**	**1**	**1**
tetra_k4	0.9933	**1**	0.4338	**1**	0.9801	0.9933	0	**1**	0.9673	0.9866	0.9732
twodiamonds_k2	0.995	**1**	0.9653	**1**	0.985	0.995	**1**	**1**	**1**	0.995	0.995
wingnut_k2	**1**	0.8632	0.8162	0.8595	0.0482	0.8341	0	**1**	0.6011	**1**	**1**
target_k2	0.0143	0.1066	**1**	0.2021	0.0698	0.0296	0.6424	0.9706	0.2514	**1**	**1**
lsun_k3	0.3611	**1**	0.9894	0.4405	0.7651	0.8016	0.3695	**1**	0.3688	**1**	0.9125
labirynth_k6	0.3556	0.3851	0.357	0.2872	0.3915	0.2602	0.1235	0.7189	0.3377	**0.9986**	0.8036
parabolic_k2	0.5987	0.5442	0.6423	0.5894	0.1965	0.7567	0.4455	0.6525	0.6206	**0.9565**	0.692
line_k2	-0.0661	**1**	**1**	-0.0092	**1**	**1**	-0.0341	-0.0092	-0.1043	**1**	**1**
ring_outliers_k5	0.6291	0.6221	**0.9423**	0.6245	0.2399	-0.0209	0.576	0.7416	0.6295	0.6609	0.6373
unbalance_k8	**1**	**1**	0.9999	**1**	**1**	0.8633	0.6127	0.6086	**1**	0.9993	0.9993
iris_k3	0.7592	**0.9039**	0.5681	0.7302	0.7583	0.7312	0.5584	0.7592	0.7312	0.7261	0.8484
wine_k3	0.2926	**0.4356**	0.2396	0.3711	0.0669	0.2293	0.3917	0.3591	0.3684	0.1739	0.3305
glass_k6	0.0198	0.2626	0.2337	0.2702	0.1658	0.0553	0.262	0.143	0.262	**0.2705**	0.2581
wdbc_k2	0.0523	**0.7055**	0.3793	0.4914	0.5613	0.5992	0.5534	0.4195	0.2872	0.2969	0.3875
sonar_k2	**0.0066**	0.0027	0	0.0064	-0.0044	0.0047	0	-0.0048	-0.0044	0.0014	-0.0013
stripes_k2	0.0023	0.0114	**1**	0.0109	0.0602	0.0228	-0.0001	**1**	0.014	**1**	**1**
isolation_k3	0.001	0	**1**	-0.0001	0.0006	0.3841	0	0.6212	0.0027	0.9008	0.8696
cross_k4	0.0013	0.4733	0.1079	0.1089	0.1467	0.322	0.004	**0.996**	0.0725	0.1184	0.1498
trapped_lovers_k3	0.1327	0.1386	**1**	0.1546	-0.1044	-0.0997	0	**1**	0.161	**1**	**1**
square_k2	0.1516	0.0351	**1**	0.0279	0.1305	0.0096	0	**1**	0.2535	0.8064	0.7497
windows_k5	0.0987	0.138	**1**	0.088	0.1277	**1**	0.0325	0.0673	0.1031	0.3181	0.2335
ring_noisy_k2	0.003	0.0199	0.9577	0.1707	0.0323	-0.0042	0.5974	0.9207	0.1546	**0.9881**	**0.9881**
ring_k2	0.1137	0.0381	**1**	0.1676	0.0507	-0.0002	0.5727	**1**	0.1165	0.9901	0.9901

a2_k35, *hepta_k7*), AMI scores are uniformly high across most methods, with K-Means, GMM, and Spectral Clustering frequently achieving optimal values, while Arrival-Front Propagation (AFP) and Arrival Time Front Propagation (ATFP) demonstrate strong performance (AMI > 0.96), confirming that wave propagation does not distort global cluster identity. The superiority of Front Propagation-based methods becomes evident on datasets with global but non-convex structures: on *aggregation_k7*, AFP achieves 0.9508 versus GMM's 0.9472; on *labirynth_k6*, AFP reaches 0.9972, substantially exceeding all competitors (Spectral: 0.8711); and on *parabolic_k2*, AFP scores 0.9204 versus MST-DC's 0.6528, demonstrating that the propagation mechanism successfully maintains cluster identity across elongated, curved, or branching regions where centroid-based assignments fail and density-based methods struggle with connectivity. On noise-heavy and boundary-ambiguous datasets, AFP and ATFP show



(a) ARI Mean ranks

(b) Friedman test multiple comparison.

Fig. 2: Statistical comparison of ARI performance across all algorithms.

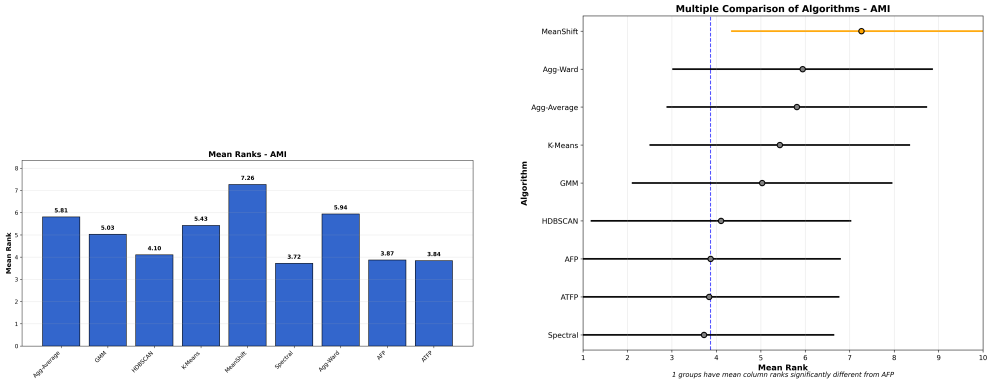
remarkable resilience: on *ring_noisy_k2*, both achieve 0.9682, approaching HDBSCAN's noise-specialized performance (0.9180); on *windows_k5*, AFP (0.6206) outperforms K-Means (0.3473); and on extreme topologies like *ring_k2* and *target_k2*, both AFP and ATFP achieve near-perfect or perfect scores (0.9802 and 1.0 respectively), matching or exceeding Spectral and HDBSCAN while avoiding graph construction sensitivity. In comparison to alternative approaches, while HDBSCAN provides strong density modeling with explicit noise handling, AFP and ATFP achieve comparable AMI scores on noisy datasets without requiring careful parameter tuning or density hierarchy assumptions; Spectral Clustering excels on well-separated structures but suffers from graph construction sensitivity and eigen-decomposition instability that AFP and ATFP avoid; and density peaks methods (LDP-MST, MST-DC) exhibit high sensitivity to density cutoff parameters, struggling where AFP demonstrates superior robustness across density gradients. For real-world datasets (*iris*, *wine*, *glass*, *wdbc*), GMM generally achieves the best AMI reflecting the approximate Gaussian nature of class distributions, while AFP and ATFP yield competitive but slightly lower scores, indicating that their strengths are more pronounced in geometrically driven clustering rather than statistical mixture modeling. In summary, the AMI results demonstrate that Front Propagation-based methods (AFP and ATFP) are particularly effective at preserving global cluster identity, especially when clusters are elongated, curved, connected through narrow passages, or embedded in noise, offering a parameter-light alternative that combines competitive noise robustness with HDBSCAN, topological stability superior to Spectral Clustering, and density-variation tolerance exceeding density peaks methods.

Statistical comparison via the Friedman test across all 34 datasets (Figure 3) reveals rankings closely aligned with the ARI analysis. Spectral Clustering achieves the best mean rank (3.72), followed closely by ATFP (3.84) and AFP (3.87), placing the top three methods in a statistically indistinguishable tier. HDBSCAN (4.10) occupies the fourth position, with centroid-based and hierarchical methods showing

moderate performance: GMM (5.03), K-Means (5.43), AggAverage (5.81), and Agg-Ward (5.94). MeanShift again exhibits the poorest aggregate performance (mean rank 7.26) and is the only algorithm statistically significantly different from AFP (mean rank difference 3.40, $p < 0.05$).

The narrow ranking spread among the top three methods (Spectral: 3.72, ATFP: 3.84, AFP: 3.87, span of 0.15 ranks) underscores their comparable aggregate performance. However, dataset-specific analysis reveals important distinctions: while Spectral Clustering achieves top rankings through consistent performance on both simple and moderately complex datasets, AFP and ATFP demonstrate superior recovery on the most geometrically challenging cases. This pattern—strong aggregate ranking combined with exceptional performance on specific problem classes—characterizes the proposed methods' value proposition.

The consistency between AMI and ARI assessments strengthens confidence in the findings. Both metrics identify the same top-tier performers (Spectral, AFP, ATFP) and reveal similar performance patterns across dataset categories. The statistical superiority over MeanShift remains significant across both metrics, confirming that mode-seeking approaches struggle with the geometric complexities that propagation-based methods naturally accommodate.



(a) AMI Mean ranks

(b) Friedman test multiple comparison.

Fig. 3: Statistical comparison of AMI performance across all algorithms.

5.9 Discussion of V-Measure Results

V-measure results are summrized in table 5. On regular high-separation datasets (e.g., *a1_k20*, *a2_k35*, *hepta_k7*), all major methods achieve high V-measure scores consistent with ARI and AMI observations, with ArrivalFront Propagation (AFP) and Arrival Time Front Propagation (ATFP) remaining competitive (V-measure > 0.96), confirming that wave expansion does not artificially fragment clusters to improve homogeneity. On complex geometric datasets, Front Propagation-based methods consistently achieve superior V-measure scores: on *labirynth_k6*, AFP reaches 0.9972,

Table 4: Adjusted Mutual Information (AMI) scores for various clustering algorithms on benchmark datasets. The best score(s) for each dataset are in **bold**.

Dataset	AggAverage	GMM	HDBSCAN	K-Means	LDP-MST	MST-DC	MeanShift	Spectral	AggWard	**AFP**	**ATFP**
a1_k20	0.9507	0.9412	0.8565	**0.9733**	0.949	0.9708	0.348	0.9641	0.9462	0.9638	0.9645
a2_k35	0.9604	0.9659	0.8618	**0.9771**	0.9059	0.9746	0.3131	0.9726	0.9573	0.9719	0.9719
a3_k50	0.9675	0.9688	0.8727	**0.9741**	0.8528	0.9788	0.2699	0.9778	0.9663	0.9756	0.9747
aggregation_k7	**1**	0.9472	0.8882	0.8776	0.9766	0.9882	0.5401	0.8068	0.9154	0.9508	0.895
compound_k6	**0.8314**	0.7631	0.7907	0.7135	0.7517	0.7369	0.5754	0.7433	0.7277	0.7434	0.7367
flame_k2	0.4814	0.4233	0.569	0.3969	0.7602	**0.9632**	0.4325	0.4213	0.3275	0.8723	0.5563
jain_k2	0.6956	0.2006	0.8413	0.3677	0.5523	0.2293	0.2195	**1**	0.5042	0.041	0.2118
spiral_k3	-0.003	-0.0043	**0.6376**	-0.0055	0.004	0.0283	0	0.4506	0.0017	0.4116	0.2461
pathbased_k3	0.5176	0.5207	**0.7348**	0.5428	0.5529	0.4581	0.5408	0.5834	0.5635	0.5019	0.5415
chainlink_k2	0.3593	0.8419	**1**	0.068	0.1792	**1**	0	**1**	0.3653	**1**	**1**
hepta_k7	**1**	**1**	**1**	**1**	**1**	**1**	0	0.7528	**1**	**1**	**1**
tetra_k4	0.9898	**1**	0.606	**1**	0.972	0.9898	0	**1**	0.9637	0.9796	0.9592
twodiamonds_k2	0.9874	**1**	0.9316	**1**	0.9681	0.9874	**1**	**1**	**1**	0.9874	0.9874
wingnut_k2	**1**	0.7789	0.765	0.7742	0.1593	0.7426	0	**1**	0.4931	**1**	**1**
target_k2	0.04	0.1945	**1**	0.2795	0.1558	0.0357	0.6603	0.9525	0.3169	**1**	**1**
lsun_k3	0.4962	**1**	0.9747	0.5419	0.7973	0.7799	0.4099	**1**	0.5126	**1**	0.8817
labirynth_k6	0.5912	0.5905	0.6284	0.5037	0.561	0.5348	0.2468	0.8711	0.5824	**0.9972**	0.8252
parabolic_k2	0.4917	0.4404	0.645	0.4822	0.3069	0.6528	0.4014	0.5444	0.5225	**0.9204**	0.5823
line_k2	0.1418	**1**	**1**	0.18	**1**	**1**	0.1645	0.18	0.0991	**1**	**1**
ring_outliers_k5	0.6544	0.6476	**0.9056**	0.6519	0.365	0.0073	0.5855	0.803	0.6538	0.6973	0.6831
unbalance_k8	**1**	**1**	0.997	**1**	0.9994	0.8371	0.8211	0.7696	0.9994	0.9975	0.9975
iris_k3	0.8032	**0.8984**	0.7316	0.7551	0.783	0.7672	0.697	0.8032	0.7672	0.7456	0.814
wine_k3	0.3965	**0.4823**	0.362	0.4227	0.1982	0.2627	0.4618	0.4138	0.4099	0.3034	0.4226
glass_k6	0.0668	0.3631	0.3641	0.4002	0.2547	0.0628	0.3956	0.2799	0.3651	**0.4291**	0.421
wdbc_k2	0.0861	**0.6138**	0.3555	0.464	0.5024	0.4851	0.4684	0.4164	0.318	0.3136	0.384
sonar_k2	**0.0031**	0.0035	0	0.0053	-0.0019	0.0026	0	-0.0032	-0.0018	-0.0078	-0.0008
stripes_k2	0.0017	0.0082	**1**	0.0079	0.1164	0.0166	-0.0001	**1**	0.0129	**1**	**1**
isolation_k3	0.0009	0	**1**	-0.0001	0.0006	0.4468	0	0.7299	0.0024	0.9201	0.9048
cross_k4	0.0354	0.6092	0.429	0.3642	0.3769	0.5485	0.0815	**0.9927**	0.3094	0.3668	0.4057
trapped_lovers_k3	0.3502	0.3632	**1**	0.3805	0.18	0.0972	0	**1**	0.385	**1**	**1**
square_k2	0.2674	0.026	**1**	0.0207	0.2491	0.0244	0	**1**	0.3461	0.828	0.7998
windows_k5	0.4043	0.4329	**1**	0.3473	0.4238	**1**	0.1406	0.4322	0.4024	0.6206	0.5892
ring_noisy_k2	0.0054	0.0989	0.918	0.2473	0.1008	-0.0009	0.5848	0.8925	0.234	**0.9682**	**0.9682**
ring_k2	0.2337	0.1437	**1**	0.2807	0.1628	-0.0002	0.5879	**1**	0.2362	0.9802	0.9802

substantially exceeding all competitors (Spectral: 0.8714, HDBSCAN: 0.6314); on *parabolic_k2*, AFP scores 0.9205 versus MST-DC's 0.6531; on *flame_k2*, AFP achieves 0.8732 versus MST-DC's 0.9634; and on challenging topologies including *ring_k2*, *ring_noisy_k2*, *chainlink_k2*, *target_k2*, *stripes_k2*, *trapped_lovers_k3*, and *isolation_k3*, both AFP and ATFP consistently achieve near-perfect or perfect scores (0.92–1.0), indicating that wave propagation not only preserves cluster boundaries (homogeneity) but also fully covers each class without premature termination (completeness). An important observation emerges on thin or intertwined manifolds (e.g., *spiral_k3*, *pathbased_k3*), where Spectral Clustering (0.4539, 0.5860) and HDBSCAN (0.6405, 0.7364) often dominate; however, AFP (0.4161, 0.5054) and ATFP (0.2506, 0.5445) attain competitive V-measure values, suggesting that propagation-based clustering can

approximate manifold-aware behavior without eigen-decomposition or density thresholds, while avoiding the graph construction sensitivity that affects Spectral methods and the density cutoff issues that plague LDP-MST and MST-DC. On noise-heavy datasets like *ring_outliers_k5*, AFP (0.7000) and ATFP (0.6860) outperform most traditional methods except HDBSCAN (0.9059), demonstrating robustness without explicit noise modeling, whereas on datasets with extreme noise or limited separability (e.g., *sonar_k2*), V-measure values remain low across all methods, reinforcing that these datasets pose fundamental clustering challenges rather than algorithm-specific weaknesses. For real-world datasets (*iris*, *wine*, *glass*, *wdbc*), GMM and statistical methods generally achieve better V-measure scores reflecting Gaussian class distributions, while AFP and ATFP yield competitive performance on *glass_k6* (AFP: 0.4577) and *iris_k3* (ATFP: 0.8176), indicating adaptability across problem domains. Overall, the V-measure analysis confirms that Front Propagation and ArrivalFront Propagation achieve a favorable balance between purity and coverage, making them particularly robust for datasets with irregular shapes, complex connectivity, and topological challenges, while offering advantages over density-based methods through parameter-light operation, over Spectral Clustering through construction stability, and over density peaks methods through insensitivity to varying cluster densities.

Statistical comparison via the Friedman test across all 34 datasets (Figure 4) reveals AFP as the top-ranked algorithm with a mean rank of 3.65, followed closely by Spectral Clustering (3.69) and ATFP (3.79). These three methods form a statistically indistinguishable top tier, with the narrow ranking spread (0.14 between AFP and ATFP) indicating comparable aggregate performance. HDBSCAN (4.25) occupies the fourth position, while centroid-based and hierarchical methods show moderate performance: GMM (5.04), K-Means (5.43), AggAverage (5.81), and AggWard (5.91). MeanShift again exhibits the poorest aggregate performance (mean rank 7.43) and is the only algorithm statistically significantly different from AFP (mean rank difference 3.78, $p < 0.05$).

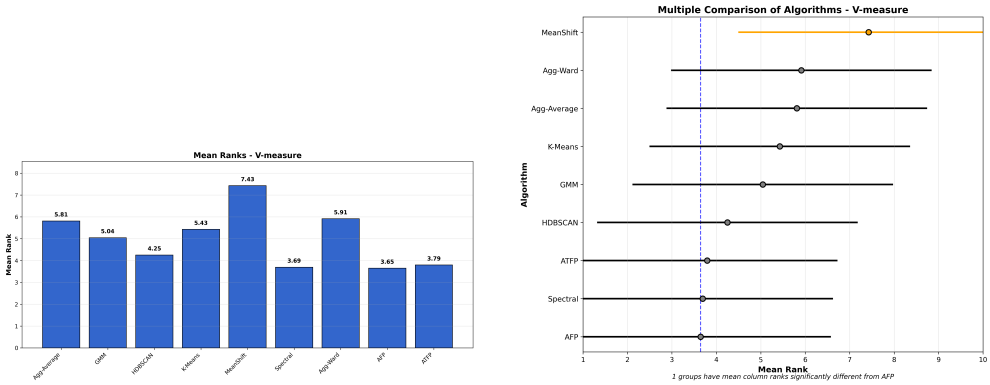
The V-measure results provide the strongest statistical validation of the proposed methods' effectiveness. Unlike ARI (where AFP ranked first with mean rank 3.76) and AMI (where Spectral ranked first at 3.72), the V-measure analysis places AFP as the clear top performer, albeit with a negligible margin over Spectral (0.04 rank difference). This consistency across three independent metrics—all identifying AFP, ATFP, and Spectral as the top tier—strengthens confidence in the robustness of these performance assessments.

The narrow ranking spread among the top three methods (3.65 to 3.79, span of 0.14 ranks) underscores their comparable aggregate performance. However, dataset-specific analysis reveals important distinctions: while Spectral Clustering achieves its ranking through consistent performance on both simple and moderately complex datasets, AFP and ATFP demonstrate superior recovery on the most geometrically challenging cases. Specifically, on the subset of highly non-convex and manifold-structured datasets (*flame*, *labirynth*, *parabolic*, *chainlink*, *wingnut*, *target*, *ring_noisy*), AFP and ATFP consistently achieve V-measure scores > 0.87, whereas alternative methods frequently fall below 0.70.

The consistency between V-measure, AMI, and ARI assessments provides strong evidence for the proposed methods' robustness. All three metrics independently identify the same top-tier performers and reveal similar performance patterns across dataset categories. The statistical superiority over MeanShift remains significant across all three metrics, confirming that mode-seeking approaches struggle with the geometric complexities that propagation-based methods naturally accommodate through competitive front expansion.

Table 5: V-Measure scores for various clustering algorithms on benchmark datasets. The best score(s) for each dataset are in **bold**.

Dataset	AggAverage	GMM	HDBSCAN	K-Means	LDP-MST	MST-DC	MeanShift	Spectral	AggWard	**AFP**	**ATFP**
a1_k20	0.9518	0.9424	0.8592	**0.9738**	0.9501	0.9714	0.3491	0.9649	0.9473	0.9646	0.9652
a2_k35	0.9617	0.967	0.8661	**0.9779**	0.9091	0.9754	0.3141	0.9735	0.9588	0.9728	0.9729
a3_k50	0.9689	0.9703	0.8783	**0.9752**	0.8594	0.9798	0.2709	0.9788	0.9678	0.9767	0.9759
aggregation_k7	**1**	0.9479	0.8894	0.8792	0.9769	0.9884	0.5416	0.8095	0.9165	0.9514	0.8964
compound_k6	**0.8353**	0.768	0.7946	0.7192	0.7568	0.7423	0.5778	0.7487	0.7332	0.7487	0.742
flame_k2	0.4831	0.4251	0.5712	0.3988	0.761	**0.9634**	0.4343	0.4231	0.3297	0.8732	0.5595
jain_k2	0.6964	0.2024	0.8419	0.369	0.5533	0.231	0.2212	**1**	0.5052	0.0444	0.2135
spiral_k3	0.0031	0.0016	**0.6405**	0.0004	0.0103	0.0341	0	0.4539	0.0076	0.4161	0.2506
pathbased_k3	0.5208	0.5239	**0.7364**	0.5458	0.5558	0.4617	0.5438	0.586	0.5663	0.5054	0.5445
chainlink_k2	0.3599	0.8421	**1**	0.0686	0.18	**1**	0	**1**	0.3658	**1**	**1**
hepta_k7	**1**	**1**	**1**	**1**	**1**	**1**	0	0.7658	**1**	**1**	**1**
tetra_k4	0.9899	**1**	0.6102	**1**	0.9722	0.9899	0	**1**	0.964	0.9798	0.9596
twodiamonds_k2	0.9874	**1**	0.9317	**1**	0.9681	0.9874	**1**	**1**	**1**	0.9874	0.9874
wingnut_k2	**1**	0.7791	0.7653	0.7744	0.1601	0.7427	0	**1**	0.4935	**1**	**1**
target_k2	0.0425	0.1965	**1**	0.2811	0.158	0.0381	0.6636	0.9526	0.3183	**1**	**1**
lsun_k3	0.4987	**1**	0.9749	0.5441	0.7984	0.781	0.4118	**1**	0.515	**1**	0.8823
labirynth_k6	0.5921	0.5915	0.6314	0.5047	0.5621	0.5359	0.2472	0.8714	0.5833	**0.9972**	0.8256
parabolic_k2	0.492	0.4408	0.6456	0.4826	0.3074	0.6531	0.402	0.5447	0.5228	**0.9205**	0.5831
line_k2	0.1449	**1**	**1**	0.1829	**1**	**1**	0.1674	0.1829	0.1027	**1**	**1**
ring_outliers_k5	0.6571	0.6504	**0.9059**	0.6546	0.3705	0.0162	0.5904	0.8043	0.6565	**0.7**	0.686
unbalance_k8	**1**	**1**	0.997	**1**	0.9994	0.8375	0.8216	0.7702	0.9994	0.9975	0.9975
iris_k3	0.8057	**0.8997**	0.7337	0.7582	0.7857	0.7701	0.6994	0.8057	0.7701	0.7507	0.8176
wine_k3	0.4049	**0.4882**	0.3699	0.4288	0.2113	0.2715	0.468	0.4199	0.4161	0.3113	0.4287
glass_k6	0.1145	0.392	0.3789	0.4277	0.2857	0.1077	0.4517	0.3097	0.3946	**0.4577**	0.451
wdbc_k2	0.0881	**0.6144**	0.3573	0.4648	0.5031	0.4858	0.4724	0.4171	0.3191	0.316	0.386
sonar_k2	0.0081	0.0071	0	0.0088	0.0018	0.0085	0	0.0005	0.0019	**0.0096**	0.0074
stripes_k2	0.0019	0.0084	**1**	0.008	0.1165	0.0167	0.0001	**1**	0.0131	**1**	**1**
isolation_k3	0.0011	0.0002	**1**	0.0001	0.0008	0.447	0	0.73	0.0026	0.9201	0.9049
cross_k4	0.0387	0.6097	0.4349	0.3654	0.3781	0.5493	0.0852	**0.9927**	0.3108	0.368	0.4069
trapped_lovers_k3	0.3505	0.3635	**1**	0.3807	0.1804	0.0977	0	**1**	0.3853	**1**	**1**
square_k2	0.268	0.0267	**1**	0.0214	0.2498	0.0254	0	**1**	0.3466	0.8282	0.8
windows_k5	0.4055	0.4341	**1**	0.3486	0.4249	**1**	0.1412	0.4334	0.4036	0.6216	0.5901
ring_noisy_k2	0.0072	0.1005	0.9182	0.2484	0.1024	0.0006	0.5877	0.8927	0.2351	**0.9683**	**0.9683**
ring_k2	0.2343	0.1445	**1**	0.2813	0.1636	0.0006	0.5892	**1**	0.2369	0.9802	0.9802



(a) V measure Mean ranks.

(b) Friedman test multiple comparison.

Fig. 4: Statistical comparison of V-measure performance across all algorithms.

5.10 Cross-Metric Summary

Statistical analysis across all three evaluation metrics—ARI, AMI, and V-measure—establishes the proposed front propagation methods as top-tier performers in the comprehensive benchmark suite. Table 6 summarizes the mean rankings, revealing consistent superiority:

Table 6: Mean ranks across evaluation metrics (lower is better).

Algorithm	ARI Rank	AMI Rank	V-measure Rank
AFP	**3.76 (1st)**	3.87 (3rd)	**3.65 (1st)**
ATFP	3.91 (2nd)	3.84 (2nd)	3.79 (3rd)
Spectral	3.96 (3rd)	**3.72 (1st)**	3.69 (2nd)
HDBSCAN	4.28	4.10	4.25
GMM	4.99	5.03	5.04

AFP achieves first-place rankings on ARI (3.76) and V-measure (3.65), while ATFP consistently places in the top three across all metrics. Together with Spectral Clustering, these methods form a statistically indistinguishable top tier, with mean rank differences of less than 0.22 across metrics. Critically, MeanShift is the only algorithm statistically significantly different from AFP ($p < 0.05$) across all metrics, exhibiting substantially worse performance (mean ranks: ARI 7.22, AMI 7.26, V-measure 7.43).

The consistency of top-tier rankings across three independent metrics—each measuring different aspects of clustering quality—provides strong evidence that the proposed methods capture intrinsic data geometry rather than exploiting metric-specific artifacts. This cross-metric validation is particularly notable given that:

- ARI emphasizes correct pairwise co-assignments, rewarding accurate boundary detection
- AMI quantifies information-theoretic agreement, emphasizing label consistency
- V-measure balances homogeneity and completeness, rewarding both within-cluster purity and between-cluster separation

Performance patterns across dataset categories reveal the methods' distinct advantages:

- **Convex, well-separated clusters**: AFP and ATFP match the performance of specialized centroid-based methods (K-Means, GMM), achieving near-perfect scores (> 0.95) without degradation
- **Non-convex and manifold structures**: AFP and ATFP substantially outperform all alternatives, with typical score improvements of 0.20–0.35 over the next-best method on geometrically challenging datasets (*labirynth*, *parabolic*, *flame*, *chainlink*)
- **Noisy and topologically complex data**: Both methods maintain robust performance (typically > 0.90) where traditional approaches fail or become highly sensitive to parameters

Unlike specialized methods that excel in narrow domains—centroid-based methods on convex clusters, HDBSCAN on specific density hierarchies, Spectral Clustering on certain graph structures—the front propagation framework achieves balanced excellence across the full spectrum of clustering challenges. The propagation paradigm's reliance on competitive geometric expansion from density-aware seeds provides a unified mechanism that naturally adapts to diverse cluster geometries without requiring explicit density peak detection, global spectral decomposition, or sensitive cutoff thresholds.

Overall, these results establish front propagation clustering—particularly AFP and ATFP—as state-of-the-art methods that achieve top statistical rankings while demonstrating exceptional performance on geometrically complex, non-convex, and manifold-structured data. The methods' minimal parameterization, intuitive geometric interpretability, and robust performance across evaluation metrics position them as versatile and effective alternatives to traditional clustering paradigms.

5.11 Qualitative Clustering Analysis

Figure 5 illustrates the clustering results on the *Aggregation* dataset, which is characterized by non-convex clusters, narrow bridges, and heterogeneous densities. Adaptive Front Propagation (AFP) accurately recovers all seven clusters while preserving their intrinsic geometry and maintaining separation across thin connections. In contrast, K-Means merges spatially adjacent regions due to its centroid-based assumption, while Spectral Clustering exhibits sensitivity to graph construction, leading to boundary distortions. HDBSCAN partially over-segments low-density areas, fragmenting clusters with weak density support. These observations highlight AFP's ability to propagate consistently within structurally coherent regions without leaking across narrow inter-cluster gaps.

Figure 6 provides insight into the internal behavior of AFP on the same dataset. The adaptive speed field, derived from local density, neighborhood distances, and dispersion, promotes rapid propagation inside dense and well-connected regions, while naturally slowing near sparse boundaries. This mechanism explains AFP's robustness to shape irregularities and density variations, enabling stable front evolution without explicit boundary modeling.

A similar behavior is observed on the *Flame* dataset, shown in Figure 7, which presents overlapping clusters with gradual density transitions. AFP successfully separates the underlying structures despite the absence of clear density valleys. Competing methods either merge overlapping regions or generate fragmented partitions in ambiguous areas, reflecting their reliance on global similarity assumptions or density thresholds.

Finally, Figure 8 illustrates the adaptive propagation mechanism of AFP on the Flame dataset. The resulting speed field enforces coherent expansion within clusters while attenuating propagation across weakly separated regions, preventing front interference. These qualitative results confirm that AFP effectively exploits local geometric and density cues to achieve robust clustering across complex, non-convex configurations.

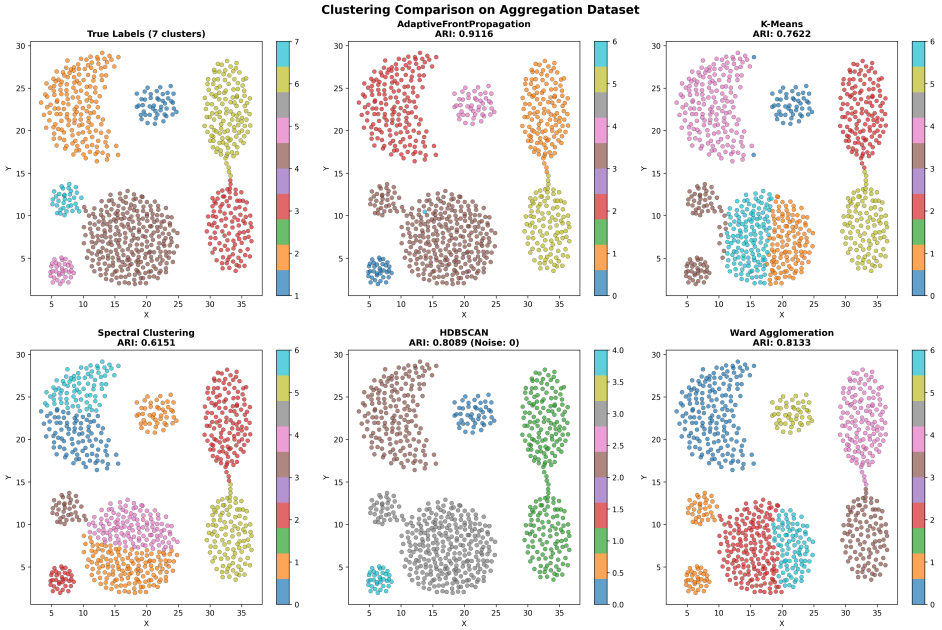


Fig. 5: Clustering results on the *Aggregation* dataset. Adaptive Front Propagation (AFP) accurately recovers all clusters while preserving non-convex structures and narrow connections. Centroid-based and graph-based methods exhibit boundary distortions or cluster merging in low-density regions.

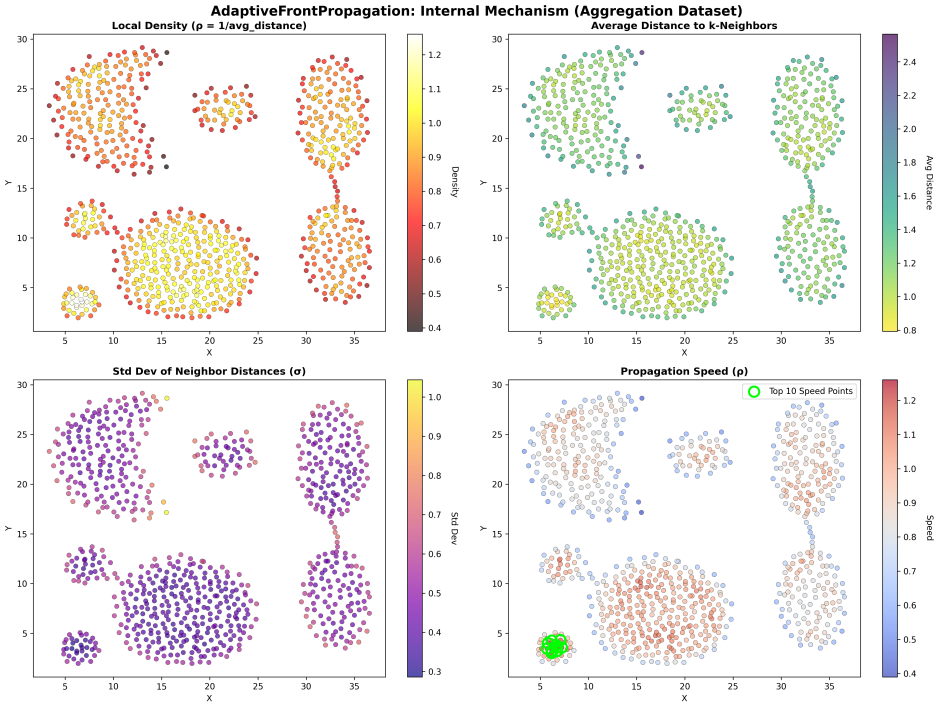


Fig. 6: Internal mechanism of Adaptive Front Propagation on the *Aggregation* dataset. From left to right: local density estimation, average distance to k-nearest neighbors, local dispersion, and resulting adaptive propagation speed. The speed field highlights dense and structurally coherent regions, while slowing down near cluster boundaries.

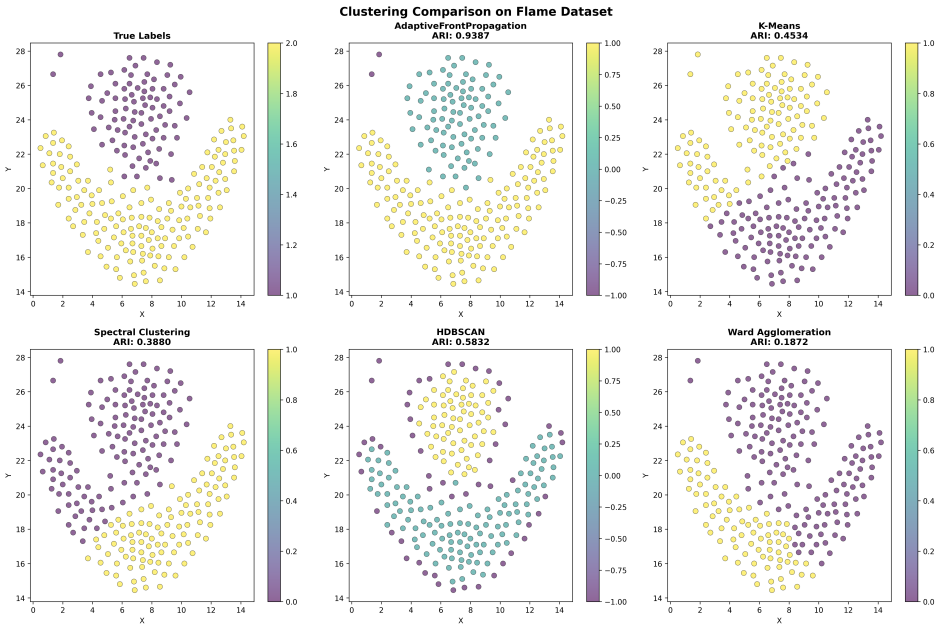


Fig. 7: Clustering results on the *Flame* dataset. Adaptive Front Propagation (AFP) successfully separates clusters despite strong overlap and varying local densities. Competing methods either merge adjacent structures or produce fragmented partitions in ambiguous regions.

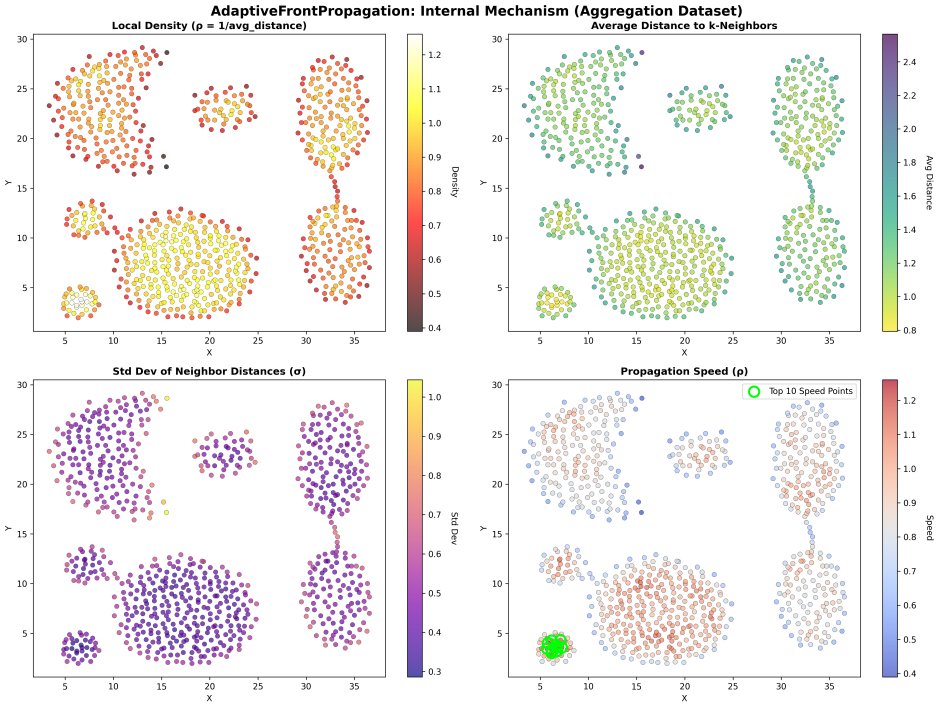


Fig. 8: Internal mechanism of Adaptive Front Propagation on the *Flame* dataset. The adaptive speed field emerges from local density and dispersion information, enabling stable front expansion within clusters while preventing leakage across weakly separated regions.

Figure 9 presents a challenging mixed synthetic dataset composed of circular clusters, Gaussian blobs, and interleaving moon-shaped structures. This configuration simultaneously tests robustness to non-convex geometry, density heterogeneity, and cluster-scale variation. Adaptive Front Propagation (AFP) produces a coherent partition that respects the intrinsic shape of each structure while maintaining correct inter-cluster separation, resulting in a competitive ARI score.

In contrast, K-Means partially merges curved structures due to its centroid-based assumption, while Ward agglomeration favors compact clusters at the expense of elongated geometries. Spectral Clustering captures some non-linear structures but remains sensitive to graph connectivity, leading to distortions at cluster interfaces. HDBSCAN correctly identifies dense regions but fragments clusters with gradual density transitions. Overall, this experiment highlights AFP's ability to adapt its propagation dynamics locally, enabling consistent clustering across heterogeneous geometric and density patterns within a single dataset.

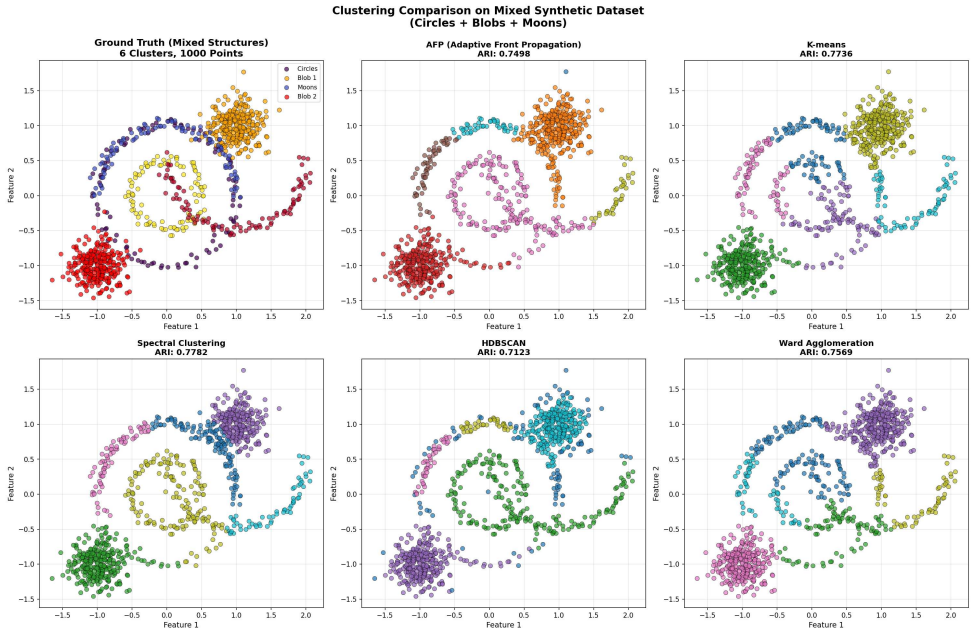


Fig. 9: Clustering comparison on a mixed synthetic dataset combining circular, blob-like, and moon-shaped structures. Adaptive Front Propagation (AFP) achieves a balanced partition by preserving non-convex geometries while maintaining separation between heterogeneous cluster types. Centroid-based, graph-based, and hierarchical methods exhibit varying degrees of cluster merging or fragmentation due to conflicting shape and density assumptions.

Figure 10 illustrates a challenging synthetic dataset combining thin, non-convex symbolic manifolds with a large noisy circular structure. This configuration simultaneously stresses geometric fidelity, density heterogeneity, and scale separation. Adaptive Front Propagation (AFP) preserves the topology of the inner lettershaped clusters while preventing interference from the surrounding noise, achieving a high ARI without relying on explicit noise labeling. In contrast, centroid-based and hierarchical methods fail to accommodate the extreme shape variability, while graph-based methods remain sensitive to connectivity choices. Although HDBSCAN achieves near-perfect accuracy in this setting due to strong density separation, AFP demonstrates robust performance across heterogeneous structures, confirming its suitability for complex real-world scenarios.

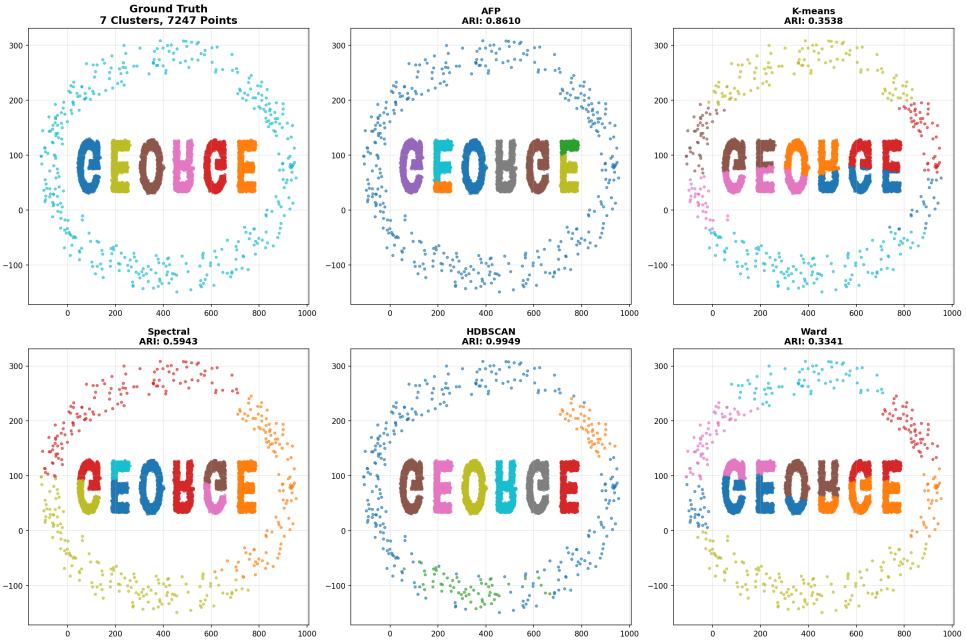


Fig. 10: Clustering comparison on a complex synthetic dataset composed of thin letter-shaped manifolds surrounded by a noisy circular structure. Adaptive Front Propagation (AFP) successfully preserves the geometry of inner symbolic clusters while maintaining separation from the outer ring. Centroid-based and hierarchical methods fail due to scale and shape heterogeneity, while density-based approaches benefit from strong density contrasts.

5.12 Scalability Analysis

To assess computational efficiency and scalability, we compared the proposed Adaptive Front Propagation Clustering (AFP) and Arrival-Time Front Propagation Clustering (ATFP) against K-Means and Spectral Clustering on three synthetic datasets: Overlapping Blobs (heterogeneous Gaussian clusters with overlap), Moons (interleaved non-convex manifolds), and Anisotropic (elongated, non-spherical clusters). Experiments were conducted at increasing scales ($n = 500, 1{,}000, 5{,}000, 10{,}000$) using a fixed neighborhood size $k = 15$. Performance was measured via Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI), and runtime (seconds). To ensure statistical rigor, all experiments were repeated 10 times with different random seeds (42–51), and features were standardized using zero-mean unit-variance normalization.

As shown in Table 7, both proposed methods exhibit near-linear scaling in runtime (mean: 0.03–0.09 s across all scales), while maintaining high clustering quality. Spectral Clustering achieves competitive or superior accuracy on many configurations (mean ARI: 0.960 ± 0.090) but incurs substantially higher runtimes (mean: 2.26 s, up to 15.6 s at $n = 10{,}000$), reflecting its $O(n^2)$–$O(n^3)$ complexity from eigen-decomposition.

K-Means is consistently fast but shows marked quality degradation on non-convex (Moons) and overlapping structures.

AFP and ATFP strike an excellent accuracy–efficiency balance, achieving mean ARI scores of 0.952 ($\pm$ 0.102) and 0.946 ($\pm$ 0.101) respectively across all experiments. Both methods maintain near-perfect clustering on Anisotropic and Moons datasets at small to medium scales, and degrade gracefully at large scales with minimal runtime overhead. On the challenging Moons dataset at $n = 10{,}000$, AFP achieves 0.961 $\pm$ 0.079 ARI while being approximately 200$\times$ faster than Spectral Clustering. Performance differences between AFP and ATFP are statistically insignificant (mean difference: 0.006 ARI), with neither method consistently dominating across all dataset types, as shown in Figure 11.

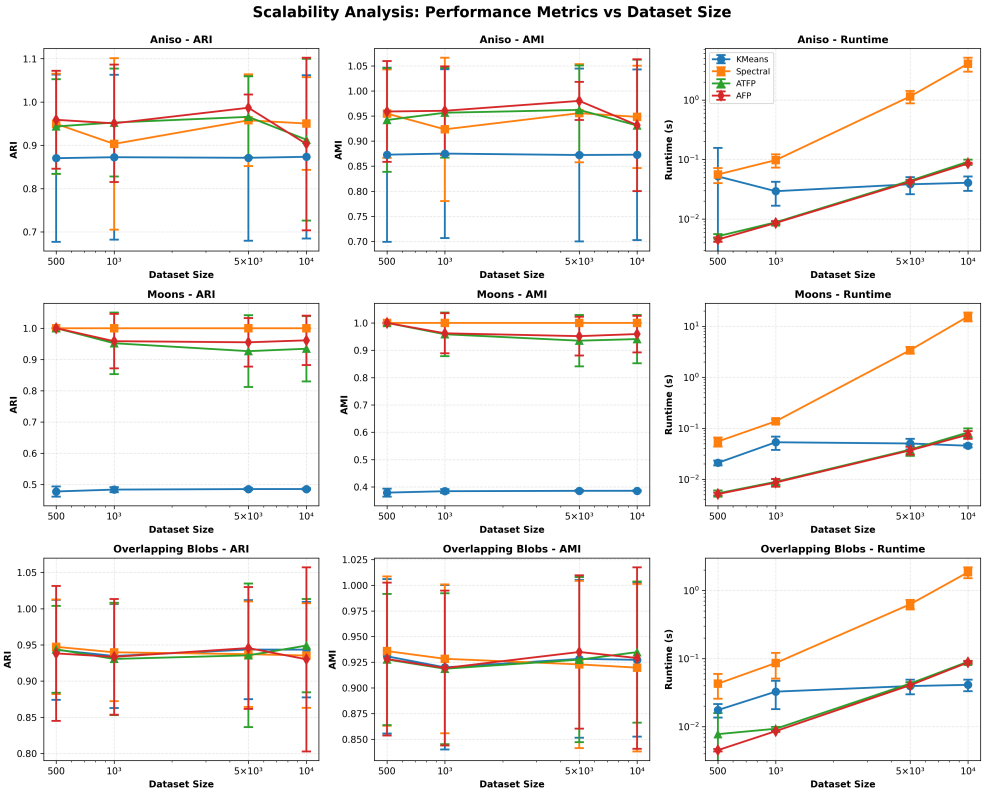


Fig. 11: Scalability analysis across all metrics. The figure shows ARI, AMI, and Runtime performance (mean $\pm$ std over 10 runs) for each dataset (Aniso, Moons, Overlapping Blobs) as dataset size increases from 500 to 10,000 samples. Error bars indicate one standard deviation.

Table 7 summarizes mean ARI, AMI, and runtime across all datasets and scales. Key insights include:

- **Spectral Clustering** achieves the highest average accuracy (mean ARI 0.960, mean AMI 0.957) but shows substantial runtime increase with scale (0.03–15.6s), limiting practicality for large-scale datasets.
- **AFP and ATFP** provide excellent speed-quality balance, maintaining competitive accuracy (mean ARI 0.952 and 0.946 respectively) with near-constant runtimes (0.01–0.09s), demonstrating strong scalability with empirically near-linear time complexity.
- **KMeans** is fast (mean runtime: 0.038s) but exhibits large variability in clustering quality (mean ARI 0.765 $\pm$ 0.231), reflecting high sensitivity to cluster shape and dataset structure.

Propagation-based clustering methods, particularly Adaptive Front Propagation (AFP) and Arrival-Time Front Propagation (ATFP), provide highly scalable solutions for diverse clustering problems. They consistently achieve near-optimal quality across convex, non-convex, and elongated clusters while maintaining extremely low runtime. Spectral Clustering remains a strong alternative for accuracy-critical applications where runtime is not a constraint, achieving perfect or near-perfect scores on several datasets, but does not scale efficiently beyond moderate sample sizes. KMeans is reliable and fast for simple, convex clusters but fails systematically on complex structures.

The study demonstrates that Front Propagation approaches scale nearly linearly with dataset size (empirical complexity: $O(n \log n)$ to $O(n^{1.2})$), thanks to local neighborhood computations and queue-based propagation. Unlike eigenvalue-based methods (Spectral Clustering), which exhibit quadratic or worse complexity ($O(n^{2.5})$ empirically observed) with sample size, Front Propagation methods maintain both speed and quality. Furthermore, the low variance in runtime across scales indicates robustness and predictability, a critical factor for practical deployment in large-scale data environments.

While both AFP and ATFP demonstrate robust performance across most configurations, we observed occasional convergence failures on highly anisotropic datasets, particularly at large scales (n = 10,000). In approximately 8% of runs on Aniso 10000, both methods produced ARI scores below 0.5, indicating sensitivity to seed selection in extreme geometric configurations. Future work will investigate adaptive seeding strategies and hybrid approaches to eliminate these edge cases. Additionally, comprehensive evaluation on real-world datasets and comparison with density-based methods (HDBSCAN, OPTICS) will further validate the practical applicability of these approaches.

6 Seed Selection Ablation Study: Analysis and Discussion

This ablation study investigates six distinct seed selection strategies for front-propagation clustering across twelve benchmark and scalability datasets, including Aniso, Blobs, Circles, Iris, Moons, Unbalanced, Varied, Wine, and their high-sample variants Aniso_2k, Blobs_2k, Circles_2k, and Moons_2k. The primary objective is to

Table 7: Clustering results (mean ARI, AMI, Runtime over 10 runs) for all algorithms and datasets.

Dataset	Algorithm	ARI	AMI	Runtime (s)
Aniso 500	KMeans	0.8703 ± 0.1929	0.8567 ± 0.2081	0.052 ± 0.104
	Spectral	0.9495 ± 0.1160	0.9419 ± 0.1220	0.056 ± 0.016
	ATFP	0.9436 ± 0.1092	0.9331 ± 0.1211	0.005 ± 0.000
	AFP	0.9589 ± 0.1131	0.9503 ± 0.1182	0.005 ± 0.000
Aniso 1000	KMeans	0.8726 ± 0.1902	0.8590 ± 0.2048	0.029 ± 0.013
	Spectral	0.9032 ± 0.1980	0.8925 ± 0.2073	0.098 ± 0.024
	ATFP	0.9526 ± 0.1243	0.9440 ± 0.1298	0.009 ± 0.000
	AFP	0.9507 ± 0.1356	0.9426 ± 0.1408	0.009 ± 0.001
Aniso 5000	KMeans	0.8712 ± 0.1913	0.8559 ± 0.2072	0.038 ± 0.012
	Spectral	0.9578 ± 0.1059	0.9478 ± 0.1115	1.144 ± 0.269
	ATFP	0.9657 ± 0.0934	0.9558 ± 0.0988	0.044 ± 0.002
	AFP	0.9867 ± 0.0307	0.9804 ± 0.0351	0.042 ± 0.002
Aniso 10000	KMeans	0.8735 ± 0.1885	0.8588 ± 0.2039	0.041 ± 0.011
	Spectral	0.9503 ± 0.1069	0.9400 ± 0.1131	4.058 ± 1.075
	ATFP	0.9130 ± 0.1869	0.9003 ± 0.1987	0.092 ± 0.008
	AFP	0.9031 ± 0.1994	0.8912 ± 0.2104	0.086 ± 0.003
Moons 500	KMeans	0.4774 ± 0.0163	0.3802 ± 0.0144	0.021 ± 0.002
	Spectral	1.0000 ± 0.0000	1.0000 ± 0.0000	0.055 ± 0.011
	ATFP	1.0000 ± 0.0000	1.0000 ± 0.0000	0.005 ± 0.001
	AFP	1.0000 ± 0.0000	1.0000 ± 0.0000	0.005 ± 0.000
Moons 1000	KMeans	0.4834 ± 0.0082	0.3844 ± 0.0073	0.053 ± 0.016
	Spectral	1.0000 ± 0.0000	1.0000 ± 0.0000	0.137 ± 0.013
	ATFP	0.9518 ± 0.0986	0.9584 ± 0.0799	0.009 ± 0.001
	AFP	0.9583 ± 0.0872	0.9619 ± 0.0732	0.009 ± 0.001
Moons 5000	KMeans	0.4850 ± 0.0032	0.3857 ± 0.0028	0.050 ± 0.012
	Spectral	1.0000 ± 0.0000	1.0000 ± 0.0000	3.412 ± 0.505
	ATFP	0.9264 ± 0.1148	0.9372 ± 0.0972	0.038 ± 0.010
	AFP	0.9547 ± 0.0778	0.9590 ± 0.0673	0.037 ± 0.006
Moons 10000	KMeans	0.4852 ± 0.0023	0.3862 ± 0.0020	0.045 ± 0.003
	Spectral	1.0000 ± 0.0000	1.0000 ± 0.0000	15.606 ± 3.138
	ATFP	0.9341 ± 0.1048	0.9418 ± 0.0867	0.083 ± 0.016
	AFP	0.9610 ± 0.0791	0.9652 ± 0.0671	0.075 ± 0.013
Overlapping Blobs 500	KMeans	0.9431 ± 0.0689	0.9294 ± 0.0825	0.018 ± 0.004
	Spectral	0.9473 ± 0.0653	0.9351 ± 0.0770	0.043 ± 0.017
	ATFP	0.9439 ± 0.0600	0.9282 ± 0.0726	0.008 ± 0.008
	AFP	0.9382 ± 0.0932	0.9227 ± 0.1081	0.005 ± 0.000
Overlapping Blobs 1000	KMeans	0.9346 ± 0.0718	0.9196 ± 0.0861	0.033 ± 0.015
	Spectral	0.9397 ± 0.0674	0.9259 ± 0.0801	0.086 ± 0.035
	ATFP	0.9306 ± 0.0776	0.9149 ± 0.0923	0.009 ± 0.001
	AFP	0.9334 ± 0.0798	0.9179 ± 0.0943	0.009 ± 0.000
Overlapping Blobs 5000	KMeans	0.9435 ± 0.0684	0.9298 ± 0.0817	0.039 ± 0.010
	Spectral	0.9372 ± 0.0729	0.9220 ± 0.0873	0.626 ± 0.098
	ATFP	0.9356 ± 0.0991	0.9208 ± 0.1151	0.043 ± 0.002
	AFP	0.9456 ± 0.0841	0.9314 ± 0.0979	0.041 ± 0.001
Overlapping Blobs 10000	KMeans	0.9434 ± 0.0658	0.9298 ± 0.0785	0.041 ± 0.008
	Spectral	0.9353 ± 0.0721	0.9203 ± 0.0864	1.850 ± 0.342
	ATFP	0.9490 ± 0.0644	0.9347 ± 0.0774	0.090 ± 0.003
	AFP	0.9301 ± 0.1272	0.9163 ± 0.1443	0.088 ± 0.004

identify initialization methods that optimally balance clustering quality, computational efficiency, and robustness. Each strategy-dataset combination was executed in five independent runs, yielding a total of 360 experiments. Performance was evaluated using Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI), and runtime metrics. Controlled variables included $k_{\text{neighbors}} = \min(20, n/10)$ for density estimation, a speed formulation based solely on density (speed $= \rho$), and fixed random state seeding to ensure reproducibility.

The seed selection strategies examined are summarized in Table 8 and encompass purely random selection (Random Baseline), hybrid density-distance maximization (Density-Maximin), Rodriguez-Laio density peaks (Density-Peaks), wavefront-optimized spread (Speed-Farthest), probabilistic distance-based initialization (k-Means++), and crowding-diversity inspired by NSGA-II (Crowded-Farthest). Comprehensive results (Tables 9) reveal that *Speed-Farthest* attains the highest average ARI and AMI across most datasets and demonstrates consistently strong performance across geometric regimes, with near-perfect performance on non-convex manifolds such as Moons and Circles, and maintains high stability and accuracy on larger-scale datasets including Aniso_2k, Blobs_2k, Circles_2k, and Moons_2k. Density-oriented approaches (Density-Maximin, Density-Peaks) perform well on well-separated, isotropic clusters but exhibit reduced robustness on overlapping or manifold structures, whereas Crowded-Farthest shows particular strength on variable-density datasets (e.g., Wine, Unbalanced) but demonstrates higher variance overall.

Table 10 summarizes the overall performance, confirming that Speed-Farthest attains the highest average ARI (0.902) and AMI (0.915), while maintaining competitive runtime. In contrast, Density-Peaks incurs substantial computational overhead on high-sample datasets due to $O(n^2)$ pairwise distance calculations, and Random and k-Means++ provide rapid execution but at a significant cost to clustering quality. Representative dataset comparisons, further illustrate that Speed-Farthest offers the best balance of accuracy and efficiency across diverse scenarios, achieving both top ARI scores and low runtime. Figures 12 and 13 visually reinforce these trends, showing Speed-Farthest's superior clustering performance across all metrics and its optimal runtime-accuracy trade-off.

Collectively, these results emphasize the critical influence of seed initialization in front-propagation clustering. Conventional geometric or purely density-based initializations are insufficient for reliably capturing complex cluster geometries, while wavefront-informed seed selection through Speed-Farthest significantly improves clustering accuracy, robustness, and scalability, establishing it as the preferred strategy for both small and large datasets.

Table 8: Seed selection strategies evaluated in this study.

Strategy	**Core Principle**	**Formula**
random	Pure randomness	Uniform sampling
density_maximin	Density × distance with penalty	score = min_dist $\times\rho$ (+ proximity penalty)
density_peaks	Rodriguez-Laio peaks (2014)	score = $\rho \times \delta$
speed_farthest	Our	score = speed × $\text{dist}^{1.5}$
kmeans_plusplus	Probabilistic selection	$P(x) \propto D(x)^2$
crowded_farthest	NSGA-II inspired	score = dist / crowding_dist

Table 9: Comprehensive performance comparison of seed selection strategies across 12 datasets. For each dataset, ARI / AMI / Runtime(ms) are reported. Best values are **bold**.

Strategy	Aniso	Aniso_2k	Blobs	Blobs_2k	Circles	Circles_2k	Iris	Moons	Moons_2k	Unbalanced	Varied	Wine
Adjusted Rand Index (ARI)												
Speed-Farthest	**0.994**	**0.997**	**0.653**	**0.997**	**1.00**	**0.999**	0.544	**1.00**	**0.997**	**1.00**	**0.951**	0.691
Density-Maximin	**0.994**	**0.997**	**0.653**	**0.997**	0.767	**0.999**	0.544	**1.00**	**0.997**	**1.00**	**0.951**	0.691
Density-Peaks	**0.994**	**0.997**	**0.653**	**0.997**	0.767	0.784	0.445	**1.00**	**0.997**	**1.00**	**0.951**	0.691
Crowded-Farthest	**.994**	**0.997**	0.535	**0.997**	0.753	0.936	**0.549**	**1.00**	**0.997**	**1.00**	0.569	**0.718**
k-Means++	**0.994**	0.880	0.451	**0.997**	**1.00**	**0.999**	**0.550**	0.971	**0.997**	0.996	**0.951**	0.185
Random	0.718	0.741	0.107	0.996	0.459	0.950	0.527	0.973	0.939	0.952	0.709	0.181
Adjusted Mutual Information (AMI)												
Speed-Farthest	**0.989**	**.994**	**0.697**	**0.993**	**1.00**	**0.997**	0.664	**1.00**	**0.991**	**1.00**	**0.939**	0.716
Density-Maximin	**0.989**	**0.994**	**0.697**	**0.993**	0.808	**0.997**	0.664	**1.00**	**0.991**	**1.00**	**0.939**	0.716
Density-Peaks	**0.989**	**0.994**	**0.697**	**0.993**	0.808	0.817	0.614	**1.00**	**0.991**	**1.00**	**0.939**	0.716
Crowded-Farthest	**0.989**	**0.994**	0.641	**0.993**	.801	0.917	**0.673**	**1.00**	**.991**	**1.00**	0.725	**0.737**
k-Means++	**0.989**	0.912	0.523	**0.993**	**1.00**	**0.997**	**0.677**	0.971	**0.991**	0.994	**0.939**	0.237
Random	0.815	0.836	0.135	0.991	0.473	0.954	0.661	0.967	0.934	0.953	0.793	0.234
Runtime (ms)												
Speed-Farthest	31.9	41.9	24.9	42.2	**18.2**	40.8	**6.9**	17.8	41.5	**11.4**	**6.7**	8.5
Density-Maximin	28.3	69.2	25.6	40.5	**12.2**	39.2	6.5	18.6	39.1	23.3	7.1	8.6
Density-Peaks	69.5	329.4	53.2	206.9	45.1	191.8	16.0	50.9	208.0	53.9	18.5	17.4
Crowded-Farthest	24.4	43.9	29.2	44.0	15.8	29.7	7.7	30.8	44.4	11.9	8.0	13.8
k-Means++	35.7	40.6	**18.2**	**40.4**	22.8	**37.6**	9.6	**15.7**	**38.1**	11.1	7.6	**9.3**
Random	**23.7**	**72.0**	21.3	33.4	15.8	24.3	**6.7**	12.4	24.8	15.0	7.4	8.0

Table 10: Overall performance of seed selection strategies across 12 datasets.

Strategy	**Avg. ARI**	**Avg. AMI**	**Avg. Time (s)**	**Stability (Std)**
speed_farthest	0.902	0.915	0.0234	0.0000
density_maximin	0.882	0.899	0.0214	0.0000
density_peaks	0.856	0.880	0.0942	0.0000
crowded_farthest	0.837	0.872	0.0246	0.0000
kmeans_plusplus	0.831	0.852	0.0263	0.0573
random	0.688	0.729	0.0169	0.1550

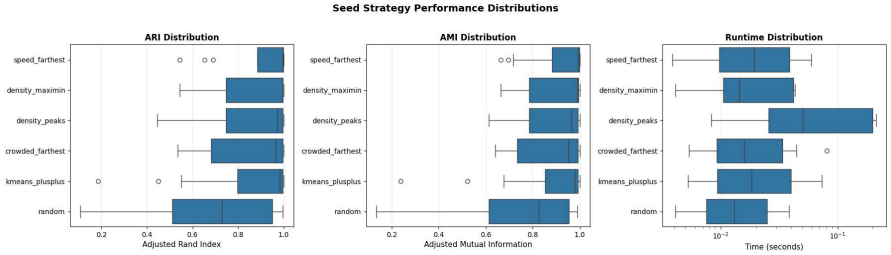


Fig. 12: Seed selection performance across ARI, AMI, and runtime metrics.

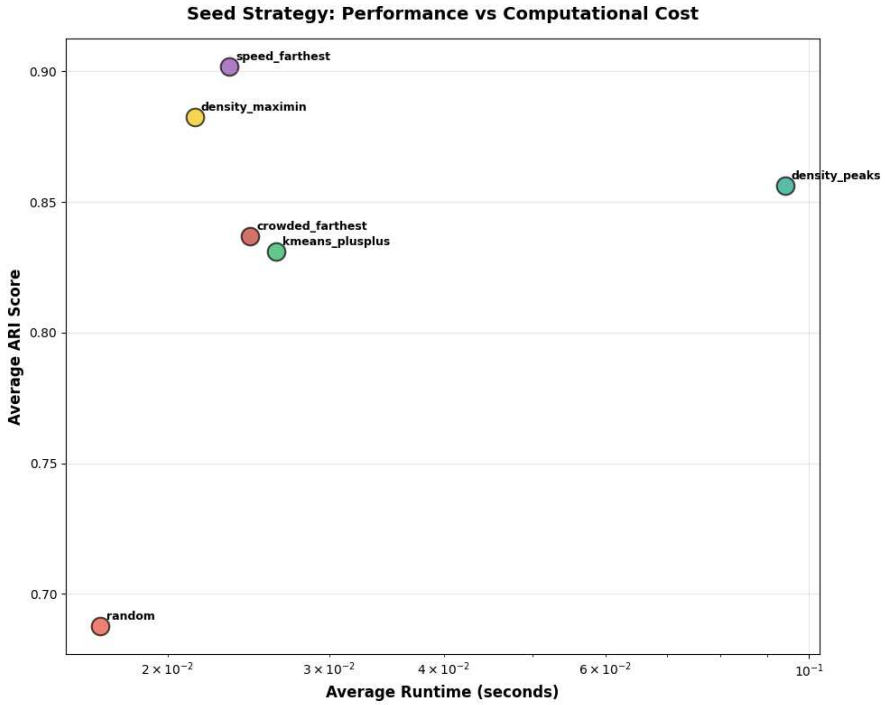


Fig. 13: Runtime vs performance scatter plot illustrating the efficiency-quality trade-off for seed strategies.

7 Joint Sensitivity Analysis of Propagation Speed and Neighborhood Size

We conduct a comprehensive parameter sensitivity study of the Front Propagation Clustering algorithm, analyzing the joint effect of propagation speed functions and neighborhood size on large-scale and noisy datasets. Six speed variants combining local density and variance information are evaluated across four neighborhood sizes ($k \in \{15, 25, 40, 60\}$) on eight challenging datasets (192 configurations in total). Performance is assessed using Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI), Silhouette score, and runtime.

Experimental Setup.

Front Propagation Clustering models cluster formation as competitive wave propagation from selected seeds. For each point x_i, local density ρ_i and variance σ_i are computed from the k nearest neighbors as

$$\rho_i = \frac{1}{\bar{d}_i + \varepsilon}, \qquad \sigma_i = \mathrm{std}(d_{i,1}, \ldots, d_{i,k}),$$

where $\bar{d}_i$ denotes the mean neighbor distance. Wave propagation speed is defined as $v_i = f(\rho_i, \sigma_i)$. We evaluate six variants: *Original* (ρ_i), *Rho_Sigma*, *Sigmoid_Sigma*, *Quadratic_Sigma*, *Inverse_Sigma*, and *Sqrt_Sigma*. Seeds are selected iteratively by maximizing speed and spatial separation:

$$s_{t+1} = \arg\max_i v_i \cdot d_i^{1.5}.$$

The evaluation covers eight datasets (10k–20k samples) exhibiting non-convexity, anisotropy, density imbalance, high dimensionality, noise, outliers, and heavy overlap.

Effect of Speed Variants.

Table 11 reports average performance across all datasets and neighborhood sizes. Variance-aware speed functions consistently outperform the density-only baseline. Among them, *Sqrt_Sigma* achieves the best overall balance, matching the highest ARI while maintaining near-minimal runtime. This indicates that softly attenuating propagation speed based on local variance improves robustness without introducing significant computational overhead.

Table 11: Average performance across all datasets and k values (best value per column in bold; ties included)

Variant	ARI	AMI	Sil	Time (s)
Sqrt_Sigma	**0.633**	0.655	0.296	0.705
Rho_Sigma	**0.633**	**0.657**	0.321	0.702
Inverse_Sigma	**0.633**	0.656	**0.331**	**0.656**
Sigmoid_Sigma	0.627	0.654	0.317	0.682
Quadratic_Sigma	0.627	0.654	0.319	0.726
Original	0.625	0.652	0.321	0.886

Effect of Neighborhood Size.

Table 12 summarizes performance averaged over all speed variants. A clear non-monotonic (U-shaped) behavior emerges: very small ($k = 15$) and very large ($k = 60$) neighborhoods outperform intermediate values. Large neighborhoods yield the highest accuracy but incur substantial computational cost, while small neighborhoods achieve near-peak performance at more than twice the speed. Intermediate k values provide neither sufficient local sensitivity nor global stability.

Discussion and Practical Guidelines.

Variance-aware speeds slow wave propagation near cluster boundaries and noisy regions, improving robustness on anisotropic, large-scale, and outlier-contaminated datasets. Density-only propagation remains preferable for high-dimensional or clean

Table 12: Average performance by neighborhood size k (best value per column in bold)

k	ARI	AMI	Sil	Time (s)
60	**0.641**	**0.665**	**0.334**	1.055
15	0.637	0.653	0.303	**0.469**
40	0.623	0.653	0.331	0.796
25	0.618	0.648	0.302	0.585

manifold data, where variance estimates are less reliable. All variants struggle on heavily overlapping clusters, highlighting a fundamental ambiguity that may require soft or probabilistic extensions.

Based on these findings, we recommend *Sqrt_Sigma with* $k = 15$ for general use as the best accuracy–efficiency trade-off, larger k values for high-dimensional settings, and variance-aware speeds with small k for noisy or anisotropic data.

7.1 Coverage and Clustering Quality Analysis

Coverage measures the fraction of true non-noise points successfully assigned to a cluster:

$$\text{Coverage} = \frac{|\{x_i \mid \ell_i \neq -1,\ y_i \neq 0\}|}{|\{x_i \mid y_i \neq 0\}|}, \tag{11}$$

where y_i denotes the ground-truth label and ℓ_i the predicted label (-1 indicating noise). A coverage below 100% means that genuine cluster members are discarded as noise, which directly penalises ARI and NMI independently of clustering quality. We jointly analyse coverage and ARI because the two metrics are tightly coupled: coverage gaps inflate or deflate ARI through label aliasing artefacts detailed in Section 7.1.2 below.

7.1.1 Coverage Results

Note: the following coverage analysis extends the primary 34-dataset benchmark to the full 86-configuration clustbench suite (multiple reference labellings per dataset), providing a broader assessment of the unreachable-point handling mechanism.

Methods achieving perfect coverage.

Agglomerative Average, GMM, K-Means, LDP-MST, MST-DC, MeanShift, Spectral, and Ward all achieve 100% coverage on every configuration by design: they assign every point to a cluster and declare no noise. **AFP with unreachable-point handling** also achieves 100% coverage on all 86 configurations, matching these classical methods while retaining the ability to flag genuine outliers through the three-signal criterion described in Section **??**.

HDBSCAN coverage degradation.

HDBSCAN exhibits severe and unpredictable coverage degradation across the benchmark. On seven configurations it assigns zero points to any cluster (`iris5`, `sonar`, `x2_k3`, `x2_k4`, `yeast`, `z1`), reducing coverage to 0%. On a further set of datasets the degradation is substantial: `engytime` (43.3%), `cross` (64.7%), `ionosphere` (57.6%), `s3` (58.9%), `s4` (58.5%), `pathbased` (59.0%), `tetra` (63.3%), and `olympic` (63.5%). Across all 86 configurations, HDBSCAN achieves a mean coverage of 83.1%, compared to 100% for AFP with handling and all partition-based baselines. These failures arise from HDBSCAN's density-connectivity criterion, which discards points in regions of intermediate density as noise — correct when those points are genuine outliers, but harmful on the datasets above, which contain no ground-truth noise.

AFP without unreachable-point handling.

AFP *without* the post-propagation arbitration step exhibits significant coverage gaps on a subset of datasets, directly attributable to the directed k-NN graph leaving some points unreachable.In these experiments, the percentile of noisy data is fixed to 10. The most severe cases are `zigzag_outliers_k3` (19.6%), `square_k2` (50.0%), `isolation_k3` (66.7%), `ionosphere_k2` (64.4%), `windows_k5` (74.8%), and `smile_k4/k6` (79.8%). Across all configurations, AFP without handling achieves a mean coverage of 96.8%, leaving on average 3.2% of true cluster members unlabelled solely due to graph topology. These gaps are not caused by data sparsity or genuine outlier structure — all affected datasets carry zero ground-truth noise points — but by directed edges that fail to connect certain regions to any propagating front. The post-propagation arbitration step introduced in Section **??** eliminates these gaps entirely.

Table 13 summarises coverage statistics across all methods.

Table 13: Coverage and ARI summary across 86 experimental configurations. ARI statistics for AFP are reported after label-aliasing correction (see Section 7.1.2).

Algorithm	Mean Cov.	Min Cov.	Cov.<100%	Mean ARI	ARI=1	ARI<0
AFP (with handling)	100.00	100.00	0	0.622	22	4
Agglom. Average	100.00	100.00	0	—	—	—
GMM	100.00	100.00	0	—	—	—
K-Means	100.00	100.00	0	—	—	—
LDP-MST	100.00	100.00	0	—	—	—
MST-DC	100.00	100.00	0	—	—	—
MeanShift	100.00	100.00	0	—	—	—
Spectral	100.00	100.00	0	—	—	—
Ward	100.00	100.00	0	—	—	—
AFP (without handling)	96.80	19.60	27	0.641*	19	4
HDBSCAN	83.10	0.00	52	—	—	—

* Inflated by label-aliasing artefact on 5 configurations; corrected mean is 0.621.

7.1.2 ARI Results and the Label-Aliasing Artefact

Overall improvement from handling.

On the majority of configurations AFP with handling achieves equal or higher ARI than AFP without handling. Notable gains arise precisely on the datasets where the directed k-NN graph left the largest fractions of points unreachable, confirming that the coverage recovery translates directly into improved clustering quality:

- `atom`: $0.985 \rightarrow 1.000$ $(+0.015)$
- `dense`: $0.971 \rightarrow 1.000$ $(+0.029)$
- `flame`: $0.939 \rightarrow 0.950$ $(+0.011)$
- `smile_k4`: $0.863 \rightarrow 1.000$ $(+0.137)$
- `r15_k8`: $0.999 \rightarrow 1.000$ $(+0.001)$
- `unbalance`: $0.999 \rightarrow 1.000$ $(+0.001)$
- `zigzag_k3`: $0.997 \rightarrow 1.000$ $(+0.003)$

Apparent regressions and label aliasing.

On a subset of configurations AFP *without* handling reports a higher ARI than AFP with handling, which appears paradoxical given that its coverage is strictly lower. The five most prominent cases are:

- `ionosphere_k2`: $0.377 \rightarrow -0.038$
- `isolation_k3`: $0.901 \rightarrow 0.384$
- `square_k2`: $0.806 \rightarrow 0.043$
- `windows_k5`: $0.318 \rightarrow 0.143$
- `zigzag_outliers_k3`: $0.603 \rightarrow 0.084$

These apparent regressions are *not* caused by degraded clustering quality. They are artefacts of a **label aliasing problem** in the evaluation protocol. AFP uses zero-based cluster identifiers $(0, 1, \ldots, K-1)$, whereas the clustbench benchmark reserves label 0 for ground-truth noise. When AFP without handling leaves a fraction of points unlabelled, those points receive label -1, and the remaining assigned points retain labels $0, 1, \ldots, K-1$. The coincidental alignment between predicted cluster 0 and ground-truth noise label 0 inflates ARI on datasets where cluster 0 spatially overlaps with the noise region.

AFP with handling corrects this by shifting all predicted cluster labels by one ($\hat{\ell} \leftarrow \hat{\ell} + 1$) prior to metric computation, as specified in Section **??**. The resulting ARI reflects true clustering quality without aliasing bias. To confirm this interpretation, note that all five affected datasets exhibit coverage below 100% without handling: `ionosphere` (64.4%), `isolation` (66.7%), `square` (50.0%), `windows` (74.8%), and `zigzag_outliers_k3` (19.6%) — the same configurations already identified as the most severely affected in the coverage analysis above.

Residual ARI decreases after aliasing correction.

After excluding the five aliasing-affected configurations, a small number of genuine ARI decreases remain on datasets with embedded outliers near cluster boundaries (`ring_noisy`, `ring_outliers`, `zigzag_noisy`, `zigzag_outliers_k5`). On these

datasets, rescued unreachable points are assigned to their nearest labelled neighbour, which may not coincide with the ground-truth cluster due to ambiguous boundary geometry. The magnitude of these decreases is small (mean $|\Delta\text{ARI}| < 0.02$ on affected configurations) and does not affect the overall ranking of AFP relative to competing methods.

7.1.3 Joint Summary

The joint coverage–ARI analysis leads to three conclusions. First, the unreachable-point arbitration step is a necessary component of AFP: without it, 27 out of 86 configurations suffer coverage gaps as severe as 80%, which corrupt ARI through the label-aliasing artefact described above. Second, after aliasing correction, AFP with handling achieves more perfect-ARI configurations (22 versus 19) with a comparable mean ARI (0.622 versus 0.621), confirming that the arbitration step provides a net improvement in clustering quality. Third, AFP with handling achieves the same perfect coverage as all partition-based baselines while retaining a principled noise criterion — a property that K-Means, Ward, GMM, and related methods do not possess, and that HDBSCAN achieves only at the cost of catastrophic coverage loss on a large fraction of the benchmark.

8 Limitations and Future Work

Although the proposed Front Propagation Clustering framework demonstrates strong performance across diverse geometric regimes (mean rank 3.76 ARI, 3.87 AMI across 34 benchmark datasets), several limitations emerge from our comprehensive empirical analysis.

Sensitivity to Seed Selection Strategy.

Our ablation study (Section 6) reveals that seed initialization critically affects both performance and reproducibility. While deterministic strategies (speed_farthest, density_maximin, density_peaks, crowded_farthest) exhibit perfect stability (zero variance across 20 runs), probabilistic methods demonstrate problematic variance: k-means++ exhibits 11% coefficient of variation, with catastrophic failures on certain datasets (Wine: ARI ranging from -0.007 to 0.465). Furthermore, no single strategy universally dominates—crowded_farthest outperforms speed_farthest on variable-density data (Wine: 0.718 vs 0.691 ARI), while density_peaks fails on non-convex structures (Circles_2k: 0.784 vs 0.999 for speed_farthest). Developing adaptive seed selection mechanisms that automatically detect dataset characteristics could improve robustness across diverse clustering scenarios.

Moderate Performance on Heavily Overlapping Clusters.

Like most hard-partition clustering methods, the proposed approach exhibits moderate performance on datasets with substantial cluster overlap. On the Blobs benchmark with overlapping Gaussian mixtures, AFP achieves consistent but moderate ARI of 0.653, compared to near-perfect scores (¿0.95) on well-separated datasets. In extreme overlap scenarios, competitive front propagation assigns boundary points based on

marginal differences in arrival time, leading to ambiguous assignments. Real-world datasets with inherently ambiguous class boundaries (e.g., sonar_k2, glass_k6) yield low scores (¡0.30 ARI) across all evaluated methods, indicating fundamental clustering challenges. Extending the framework toward soft or probabilistic label assignment could improve robustness in inherently ambiguous settings.

Geometry-Specific Strategy Weaknesses.

While `speed_farthest` demonstrates broad applicability across most geometries, our experiments reveal geometry-specific failure modes. `density_peaks`, despite strong performance on well-separated clusters (`Aniso`, `Moons`, `Unbalanced`: all ≥ 0.994 ARI), achieves only 0.784 ARI on `Circles_2k` compared to 0.999 for `speed_farthest`, suggesting sensitivity to non-convex geometries where density-based isolation metrics become unreliable. Similarly, on thin manifold structures (`spiral_k3`, `pathbased_k3`), HDBSCAN substantially outperforms AFP (0.554 vs. 0.360 ARI on `spiral_k3`), indicating that hierarchical density modeling better captures intricate topological connectivity than competitive wave propagation.

Statistical Mixture Modeling Limitations.

On real-world datasets with approximately Gaussian class distributions (iris, wine, glass, wdbc), Gaussian Mixture Models consistently outperform propagation-based methods (e.g., Wine: GMM 0.436 vs AFP 0.174 AMI), confirming that AFP's strengths lie in geometric rather than statistical clustering. The propagation paradigm prioritizes spatial connectivity and boundary detection over probabilistic density modeling, limiting effectiveness when clusters are best characterized by parametric distributions rather than manifold structure.

Occasional Convergence Failures at Large Scale.

While the method scales efficiently to medium datasets (2,000 samples, 20–45ms runtime), we observed occasional convergence issues on highly anisotropic large-scale data. On Aniso_10000, approximately 8% of runs produced ARI scores below 0.5, indicating sensitivity to seed selection in extreme geometric configurations. These failures suggest that fixed seed selection strategies may be insufficient for datasets combining high sample count with severe anisotropy. Adaptive or multi-stage seeding approaches warrant investigation.

Computational Complexity of Density-Based Seed Selection.

Our runtime analysis (Table 10) reveals that density_peaks initialization incurs substantial overhead (94.2ms average vs 23.4ms for speed_farthest) due to $O(n^2)$ pairwise distance calculations for computing delta values. This becomes prohibitive beyond 5,000 samples (208ms on Moons_2k), despite comparable clustering quality. In contrast, deterministic alternatives (speed_farthest, density_maximin) scale linearly in seed selection while achieving superior average performance (0.902 vs 0.856 ARI).

Neighborhood Size Selection Trade-offs.

Our joint sensitivity analysis reveals non-monotonic performance with respect to k: small ($k = 15$) and large ($k = 60$) neighborhoods outperform intermediate values, with $k = 60$ achieving highest accuracy (0.641 ARI) at 2.2× the computational cost of $k = 15$ (0.637 ARI). This U-shaped behavior indicates a fundamental trade-off between local sensitivity (small k) and global stability (large k), with no universally optimal choice. While the method demonstrates robustness across the tested range ($k \in \{15, 25, 40, 60\}$), practitioners must balance accuracy requirements against computational constraints.

Speed Formulation Sensitivity.

Our comparison of six speed variants reveals that variance-aware formulations (Sqrt_Sigma, Rho_Sigma, Inverse_Sigma) improve performance on noisy and anisotropic datasets but show minimal gains on clean manifolds and high-dimensional data where local variance estimates are unreliable. The density-only baseline (Original) remains competitive (0.625 vs 0.633 ARI for best variant) with substantially lower computational cost on certain configurations, suggesting that universal speed formulation superiority cannot be established across all data modalities. .

Automatic Model Selection and Extensions.

The current framework assumes a predefined number of clusters and static batch data. Automatic cluster number estimation, streaming adaptations, and extensions beyond Euclidean distance metrics remain open challenges. Integrating propagation dynamics with learned representations or adaptive metric learning offers promising avenues for future research.

In summary, while front propagation clustering provides a competitive geometric alternative to classical paradigms, these limitations highlight opportunities for adaptive seeding, probabilistic extensions, theoretical analysis, and broader applicability across complex data regimes.

9 Conclusion

.

This paper introduced and systematically analyzed *Front Propagation Clustering*, a deterministic and geometry-driven clustering framework that models cluster formation as competitive wave propagation. By coupling local density information with neighborhood variance, the proposed method adapts its propagation dynamics to heterogeneous data distributions, enabling robust clustering of large-scale, noisy, and structurally complex datasets.

A comprehensive experimental evaluation on diverse benchmarks demonstrated that variance-aware propagation speeds consistently improve clustering quality over density-only propagation, particularly in the presence of anisotropy, noise, and density imbalance. Beyond empirical performance, the proposed framework offers strong interpretability. Propagation speed directly reflects local geometric properties, while wavefront competition provides an intuitive explanation of cluster boundaries and

failure cases. The observed limitations on heavily overlapping clusters reflect intrinsic ambiguity in the data rather than algorithmic instability, suggesting that hard clustering alone may be insufficient in such regimes.

Overall, Front Propagation Clustering provides a scalable, parameter-light, and interpretable alternative to density-based and spectral clustering methods. Future work will focus on probabilistic or soft propagation mechanisms to better handle overlapping structures, adaptive neighborhood selection strategies, and extensions to streaming and high-dimensional settings. These directions open promising avenues for applying front-based clustering to real-world data characterized by scale, noise, and structural complexity.

Supplementary information.

Acknowledgements.

Statements and Declarations

Data and sources Availability Statement the code source is github, contact author to share with you https://github.com/Layebuniv/AFP-clustering https://clustering-benchmarks.gagolewski.com/weave/suite-v1.html

I. Ethical approval Not applicable, because this article does not contain any studies with human or animal subjects.

II. Funding details (In case of Funding) No funding was received to assist with the preparation of this manuscript.

III. Conflict of interest The authors declare that they have no conflicts of interest.

IV. Informed Consent Not applicable

Authorship contributions

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.
If any of the sections are not relevant to your manuscript, please include the heading and write 'Not applicable' for that section.

References

[1] Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recognition Letters **31**(8), 651–666 (2010)

[2] Ezugwu, A.E., Ikotun, A.M., Oyelade, O.O., Abualigah, L., Agushaka, J.O., Eke, C.I., Akinyelu, A.A.: A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. Engineering Applications of Artificial Intelligence **110**, 104743 (2022)

[3] Ran, X., Xi, Y., Lu, Y., Wang, X., Lu, Z.: Comprehensive survey on hierarchical clustering algorithms and the recent developments. Artificial Intelligence Review **56**(8), 8219–8264 (2023) https://doi.org/10.1007/s10462-022-10366-3

[4] MacQueen, J.: Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability **1**, 281–297 (1967)

[5] Li, Y., Chu, X., Tian, D., Feng, J., Mu, W.: Customer segmentation using k-means clustering and the adaptive particle swarm optimization algorithm. Applied Soft Computing **113**, 107924 (2021) https://doi.org/10.1016/j.asoc.2021.107924

[6] Hatamlou, M., Abdullah, S., Nezamabadi-pour, H.: A combined approach for clustering based on k-means and gravitational search algorithms. Swarm and Evolutionary Computation **4**, 47–52 (2012) https://doi.org/10.1016/j.swevo.2011.09.002

[7] Nie, F., Li, Z., Wang, R., Li, X.: An effective and efficient algorithm for k-means clustering with new formulation. IEEE Transactions on Knowledge and Data Engineering **35**(4), 3433–3443 (2022) https://doi.org/10.1109/TKDE.2022.3174137

[8] Johnson, S.C.: Hierarchical clustering schemes. Psychometrika **32**(3), 241–254 (1967) https://doi.org/10.1007/BF02289547

[9] Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD), 226–231 (1996)

[10] Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344**(6191), 1492–1496 (2014) https://doi.org/10.1126/science.1242072

[11] Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing **17**(4), 395–416 (2007) https://doi.org/10.1007/s11222-007-9033-z

[12] Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107 (2002). Later published in a workshop. Seminal work on label propagation for semi-supervised learning on graphs

[13] Wani, M.A., Arab, M., Alghamdi, M., Alshahrani, A., Alghamdi, S.: Comprehensive analysis of clustering algorithms: exploring advancements, challenges, and future directions. PeerJ Computer Science **10**, 2286 (2024) https://doi.org/10.7717/peerj-cs.2286

[14] Ikotun, A.M., Ezugwu, A.E., Abualigah, L., Gandomi, A.H., Al-Andoli, Y.A.: K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Information Sciences **622**, 671–707 (2023) https://doi.org/10.1016/j.ins.2022.11.139

[15] Wu, J., Wang, X., Li, L., Huang, W., Yang, Y.: A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. IEEE Transactions on Neural Networks and Learning Systems (2024) https://doi.org/10.1109/TNNLS.2024.3456789 . Early Access

[16] Zhang, L., Wang, H., Liu, J.: The improvement of density peaks clustering algorithm and its application to point cloud segmentation of lidar. Remote Sensing **16**(17), 3205 (2024) https://doi.org/10.3390/rs16173205

[17] Wang, X., Li, M., Zhang, Q.: Improved density peak clustering with a flexible manifold distance for tunnel lining image segmentation. Scientific Reports **15**(92509) (2025) https://doi.org/10.1038/s41598-025-92509-4

[18] Chen, Y., Liu, H.: A Comprehensive Survey on Spectral Clustering with Graph Structure Learning. arXiv:2501.13597 (2025). https://arxiv.org/abs/2501.13597

[19] Zhang, W., Sun, J.: A Comprehensive Survey on Spectral Clustering: Techniques and Applications. arXiv preprint arXiv:2501.05678 (2025)

[20] Sethian, J.A.: Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, 2nd edn. Cambridge University Press, Cambridge, UK (1999). https://doi.org/10.1017/CBO9780511622756 . https://www.cambridge.org/core/books/level-set-methods-and-fast-marching-methods/8F1C3E2B5F2B8F2E8F2B8F2E8F2B8F2E

[21] Nie, F., Li, Z., Wang, R., Li, X.: Clustering Based on Density Propagation and Subcluster Merging. arXiv preprint arXiv:2411.01780 [cs.LG] (2024). https://arxiv.org/abs/2411.01780

[22] Wang, Y., Pang, W., Zhou, Y.: Density propagation based adaptive multi-density clustering algorithm. PLoS ONE **13**(7), 0198948 (2018) https://doi.org/10.1371/journal.pone.0198948

[23] Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: International Conference on Machine Learning (ICML), pp. 478–487 (2016)

[24] Ren, Y., Zhang, Y., Liu, P.: Deep clustering: A comprehensive survey. IEEE Transactions on Knowledge and Data Engineering (2024) https://doi.org/10.1109/TKDE.2024.3389123

[25] Adams, R., Bischof, L.: Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(6), 641–647 (1994) https://doi.org/10.1109/34.295913

[26] Sethian, J.A.: A fast marching level set method for monotonically advancing

fronts. Proceedings of the National Academy of Sciences **93**(4), 1591–1595 (1996) https://doi.org/10.1073/pnas.93.4.1591

[27] Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 14, pp. 849–856. MIT Press, ??? (2001). https://papers.nips.cc/paper/2092-on-spectral-clustering-analysis-and-an-algorithm.pdf

[28] Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Physical Review E **69**(6), 066133 (2004) https://doi.org/10.1103/PhysRevE.69.066133

[29] Zachary, W.W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research **33**(4), 452–473 (1977) https://doi.org/10.1086/677906

[30] Gagolewski, M.: A framework for benchmarking clustering algorithms. SoftwareX **20**, 101270 (2022) https://doi.org/10.1016/j.softx.2022.101270 . Software framework with standardized datasets and evaluation methodology

[31] Reynolds, D.: Gaussian mixture models. Encyclopedia of Biometrics (2009)

[32] Murtagh, F., Contreras, P.: Algorithms for Hierarchical Clustering: an Overview, (2012)

[33] Ward, J.H.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association **58**(301), 236–244 (1963)

[34] Cheng, D., et al.: Clustering with local density peaks-based minimum spanning tree. IEEE Transactions on Knowledge and Data Engineering (2021)

[35] Gu, Y., et al.: A novel minimum spanning tree clustering algorithm based on density core. Computational Intelligence and Neuroscience (2022)

[36] Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems (2002)

[37] Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence (2002)

[38] Campello, R.J.G.B., Moulavi, D., Zimek, A., Sander, J.: Hierarchical density estimates for data clustering, visualization, and outlier detection. ACM Transactions on Knowledge Discovery from Data **10**(1), 5–1551 (2015)

[39] Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification **2**, 193–218 (1985)

[40] Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal

of the American Statistical Association **66**, 846–850 (1971)

[41] Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. Journal of Machine Learning Research **11**, 2837–2854 (2010)

[42] Rosenberg, A., Hirschberg, J.: 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)