AIMS Mathematics, 11(3): 7871–7909.
DOI:10.3934/math.2026325
Received: 14 January 2026
Revised: 10 March 2026
Accepted: 16 March 2026
Published: 25 March 2026

https://www.aimspress.com/journal/Math

Research article

A fast granular ellipsoid-based density peaks clustering algorithm for large-scale data

Shihu Liu[1,2], **Shuang Li**[1,*] **and Fusheng Yu**[3]

[1] School of Mathematics and Computer Science, Yunnan Minzu University, Kunming 650504, China

[2] Fujian Provincial Key Laboratory of Data-Intensive Computing, Quanzhou Normal University, Quanzhou 362000, China

[3] School of Mathematical Sciences, Beijing Normal University, Beijing 100875, China

* **Correspondence:** Email: 24213037570008@ymu.edu.cn; Tel: 18314435328.

Abstract: As an effective clustering approach, the density peaks clustering (DPC) has been extensively studied in recent years. However, the traditional DPC algorithm suffers from not only high computational complexity, but also a limited capability to identify non-spherical or anisotropic clusters. Therefore, we combine the concept of granular computing with ellipsoidal modeling and propose a novel algorithm termed granular-ellipsoid density peaks (GEDP). Meanwhile, we extend the granular ball model into a granular ellipsoid ($\mathcal{GE}$) model through a hierarchical splitting and fitting process guided by compactness and shape, enabling adaptive modeling of local geometry. Furthermore, the Mahalanobis distance is utilized to capture feature correlations and anisotropy, providing a more faithful description of data structure. Based on this, we define ellipsoid-level density and δ-distance in an adaptive and parameter-free manner without requiring any manually tuned thresholds or kernel widths. We further redesign the automatic cluster center identification and refinement processes using a normalized γ criterion, combined with robust label propagation and post-processing to ensure reliable clustering performance. Most importantly, comprehensive experiments on synthetic, real-world, and large-scale datasets demonstrate the effectiveness, scalability, and robustness of the proposed GEDP algorithm. The results further confirm its strong adaptability within various data distributions, particularly on large-scale datasets.

Keywords: clustering; density peaks clustering; granular computing; granular ellipsoid; large-scale data

Mathematics Subject Classification: 62H30, 68T10, 68W27

1. Introduction

The density peak clustering (DPC) algorithm [1] has emerged as an effective density-based clustering approach due to its capability of automatically identifying cluster centers without assuming predefined cluster shapes. It determines cluster centers based on two intuitive criteria: high local density and large distance from other high-density points. Owing to its simplicity and interpretability, DPC has been successfully applied in various domains such as image segmentation, bioinformatics, and anomaly detection [2, 3].

Despite its advantages, the original DPC algorithm suffers from two major limitations. First, it requires the computation of pairwise distances among all data points, resulting in a quadratic computational complexity that significantly limits its scalability on large-scale datasets [4, 5]. Modern applications in image analysis, Internet of Things (IoT) monitoring, and bioinformatics frequently involve massive datasets containing millions of samples, making scalable clustering frameworks increasingly important [6, 7]. Second, DPC treats each data point as an individual unit, which makes it sensitive to noise and local density fluctuations. To address these issues, several extensions have been proposed, including Fast-DP [4], local-density adaptive variants [8, 9], and distributed implementations [10]. Although these approaches improve efficiency or robustness to some extent, they often rely on heuristic parameter tuning or approximations that may compromise clustering accuracy [11].

To further improve scalability, the concept of granular ball computing (GBC) [12, 13] was introduced as a coarse-to-fine data abstraction framework. Instead of operating directly on individual samples, GBC represents local data regions as spherical granules, thereby significantly reducing the computational burden of clustering and classification tasks. Building on this idea, the granular-ball density peaks algorithm (GB-DP) [14] replaces sample-level operations in DPC with granule-level processing. This design reduces the computational complexity from quadratic to near-linear, making it suitable for large-scale datasets containing millions of samples. Moreover, the granule representation inherently suppresses noise and improves the stability of density estimation.

Nevertheless, existing GB-DP methods [14, 15] are fundamentally constrained by their isotropic assumption, where each granule is modeled as a hypersphere with uniform variance in all directions. This isotropic assumption limits the ability of GB-DP to represent elongated or directionally correlated clusters. Consequently, density estimation may become distorted, and cluster boundaries may be inaccurately identified when dealing with anisotropic data distributions, as illustrated in Figure 1. Figure 1(a) shows the original dataset, while Figure 1(b) presents the granular-ball representation, where clusters are approximated by circles that ignore directional information. In contrast, Figure 1(c) illustrates the proposed granular ellipsoid representation [16], which adaptively aligns its orientation and scale with the intrinsic geometry of the data, thereby providing a more faithful and compact description of anisotropic structures. In many real-world datasets, directional variance and local correlations are common characteristics that spherical granules fail to capture.

Furthermore, recent studies on improving DPC have explored several directions, including adaptive decision-graph-based clustering methods , KNN- and graph-theory-enhanced density peak clustering, and granular-sphere clustering techniques designed for non-spherical data. These approaches provide valuable insights into handling complex data structures and improving algorithm robustness while maintaining scalability for large-scale datasets.

In parallel, incomplete multi-view clustering has recently attracted considerable attention in scenarios where multi-view data contain missing entries. For instance, Liu et al. [17] proposed a self-guided partial graph propagation method that exploits cross-view consistency and complementarity to infer missing information. Another approach [18] reformulates the missing instance problem as a similarity graph completion task and performs discriminative representation learning through self-supervision. More recently, Liu et al. [19] introduced a latent-structure-aware view recovery framework that leverages structural information to reconstruct incomplete views. Although these methods mainly focus on view-missing scenarios rather than anisotropic clustering, their strategies for exploiting structural consistency provide complementary perspectives for designing more robust clustering frameworks.

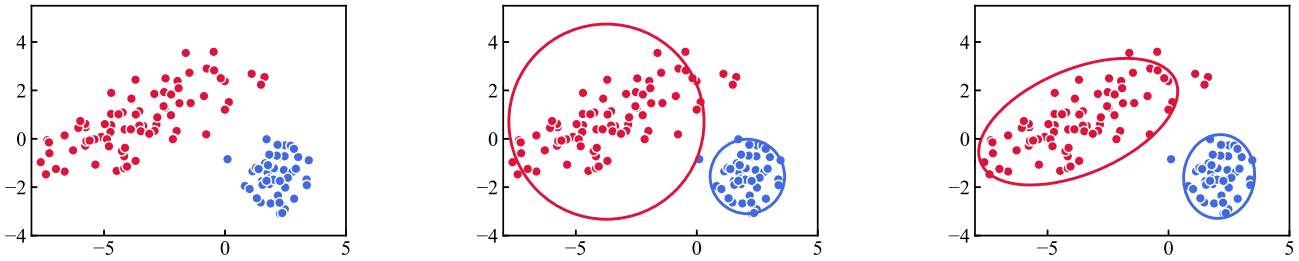


Figure 1. Conceptual comparison between granular ball and granular ellipsoid representations in capturing anisotropic cluster structures.

Motivated by the above limitations, we propose a novel framework called GEDP, which generalizes spherical granules into adaptive granular ellipsoids ($\mathcal{GE}$s). The orientations and axis lengths of these ellipsoids are estimated from the covariance of enclosed samples, enabling each granule to align with the intrinsic geometry of the data. As a result, the proposed representation can accurately capture anisotropic structures while maintaining scalability on large-scale datasets. The framework includes three key components: (i) a regularized covariance-based estimation strategy to ensure numerical stability, (ii) a hierarchical splitting mechanism driven by compactness and shape metrics for adaptive granule refinement, and (iii) an ellipsoid-level density peaks clustering procedure with a normalized γ criterion for automatic center identification. These components collectively provide geometric adaptability, robustness to noise, and high computational efficiency.

The main contributions of this work can be summarized as follows:

- **Granular ellipsoid representation:** We extend the isotropic granular-ball model to an anisotropy-aware $\mathcal{GE}$ representation that captures local geometric correlations and directional variances.
- **Adaptive multi-level granule generation:** A hierarchical splitting mechanism guided by compactness and shape metrics is proposed to construct granular ellipsoids with appropriate scales.
- **Ellipsoid-level density peaks clustering:** The DPC paradigm is extended from individual data points to granular ellipsoids using a density-distance formulation, improving both efficiency and robustness.

- **Automatic center selection and refinement:** A normalized γ-based criterion combined with a refined label-propagation strategy enables accurate and consistent point-level clustering.
- **Scalability for large-scale datasets:** By operating on coarse-to-fine granular ellipsoids, the proposed GEDP framework achieves near-linear computational complexity and demonstrates superior efficiency on large datasets.

The remainder of this paper is organized as follows: Section 2 introduces the necessary preliminaries. Section 3 presents the proposed GEDP algorithm in detail. Section 4 describes the experimental setup. Section 5 reports the experimental results and analysis. Finally, Section 6 concludes the paper and outlines future research directions.

2. Preliminaries

In this section, we introduce some notations of this paper as well as some basic concepts that are closely related to this work, such as the DPC algorithm and granular ellipsoid computing.

2.1. Notations and descriptions

Some of the necessary notations are summarized in Table 1 for an easy read.

Table 1. Notations and descriptions.

Notations	Descriptions
D	The dataset, i.e., $D = \{x_1, x_2, \cdots, x_m\}$, x_i is the ith data point in D.
$\mathcal{G}$	The set of all $\mathcal{GE}$s, i.e., $\mathcal{G} = \{\mathcal{GE}_1, \mathcal{GE}_2, \cdots, \mathcal{GE}_t\}$.
$\mathcal{GE}(P, \mathcal{H}, c)$	A $\mathcal{GE}$ with point set P, shape matrix $\mathcal{H}$, and center c.
$r, s, \kappa, \rho(\mathcal{GE}), \rho_{\text{norm}}(\mathcal{GE}), V$	The radius, shape score, compactness, density, normalized density, and volume of $\mathcal{GE}$.
$d(x, \mathcal{GE})$	The distance between a point x and $\mathcal{GE}$.
$d(\mathcal{GE}_i, \mathcal{GE}_j)$	The distance between two $\mathcal{GE}$s.
$\delta(\mathcal{GE}), \delta_{\text{norm}}(\mathcal{GE})$	The δ-distance and normalized δ-distance to the nearest higher-density $\mathcal{GE}$.
$N(\mathcal{GE})$	The nearest higher-density $\mathcal{GE}$.
$\gamma(\mathcal{GE})$	The decision value.
C	The set of all centers.
$\mathcal{L}$	The set of final point-level clustering labels.

2.2. The DPC algorithm

The density peaks clustering (DPC) algorithm consists of two main stages: (i) the identification of cluster centers and (ii) the assignment of the remaining data points. A data point $x_i \in D$ is considered as a cluster center if it simultaneously satisfies two criteria: (1) it possesses a relatively high local density ρ_i, and (2) it is located at a large distance from any other point with higher density.

The local density ρ_i of a point x_i can be computed by the equation:

$$\rho_i = \sum_{j \neq i} \chi(d_{ij} - d_c), \tag{2.1}$$

where d_{ij} denotes the distance between points x_i and x_j, d_c is a cutoff distance [20], and $\chi(\cdot)$ is an indicator function defined as

$$\chi(x) = \begin{cases} 1, & \text{if } d_{ij} < d_c, \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

Moreover, the value of ρ_i can also be computed by equation:

$$\rho_i = \sum_{j \neq i} \exp\left(-\frac{d_{ij}^2}{d_c^2}\right). \tag{2.3}$$

Equation (2.1) is referred to as the cutoff kernel method, while Eq (2.3) is called the Gaussian kernel method. Regardless of which method is chosen, the value of ρ_i is highly sensitive to the parameter d_c.

The value of δ_i can be computed by the following equation:

$$\delta_i = \begin{cases} \min\limits_{j:\rho_j > \rho_i} d_{ij}, & \text{if } \rho_i \neq \max(\rho), \\ \max\limits_{j} d_{ij}, & \text{otherwise.} \end{cases} \tag{2.4}$$

After constructing a decision graph with ρ_i on the x-axis and δ_i on the y-axis, points appearing in the upper-right region are typically regarded as cluster centers [21]. Quantitatively, a higher γ_i value indicates a greater likelihood that x_i serves as a cluster center, where γ_i is defined as

$$\gamma_i = \rho_i \, \delta_i. \tag{2.5}$$

After determining the cluster centers $\{c_1, c_2, \ldots, c_k\}$, each remaining point is assigned to the same cluster as its nearest neighbor with higher density, completing the clustering.

2.3. Granular ellipsoid computing

Granular computing (GrC) [12] is a computational paradigm that focuses on the efficient representation and processing of complex data by constructing information granules at multiple levels of abstraction. The classical granular-ball (GB) model [13] employs hyperspheres as basic granules, providing simplicity, robustness, and computational efficiency for clustering and classification tasks.

A granular ball is characterized by a center c and a radius r, formally defined as

$$GB(c, r) = \{x \in \mathbb{R}^n \mid \|x - c\|_2 \leq r\}. \tag{2.6}$$

Let P denote the set of data points enclosed by a granular ball. The center c is typically computed as the mean of all points in P:

$$c = \frac{1}{|P|} \sum_{x_i \in P} x_i. \tag{2.7}$$

The radius r can be defined in different ways. One common choice is the average distance from the points in P to the center,

$$r = \frac{1}{|P|} \sum_{x_i \in P} \|x_i - c\|_2, \tag{2.8}$$

while another option is the maximum distance,

$$r = \max_{x_i \in P} \|x_i - c\|_2. \tag{2.9}$$

Equation (2.9) reflects the compactness of the granule and provides an intuitive measure of its spatial extent. However, due to the isotropic assumption of uniform spread in all directions, the granular ball model encounters limitations when representing anisotropic or elongated data distributions.

Recent studies on ellipsoidal granular computing models [16] have demonstrated improved capability in capturing anisotropic structures and enhancing classification performance, providing theoretical motivation for the granular ellipsoid model adopted in this work.

A granular ellipsoid is defined as the set of points determined by a center $c \in \mathbb{R}^n$ and a symmetric positive definite shape matrix $H \in \mathbb{R}^{n\times n}$ [16]:

$$GE(H, c) = \{x \in \mathbb{R}^n \mid (x - c)^\top H(x - c) \leq 1\}. \tag{2.10}$$

Note that $GE(H, c)$ represents a continuous region in $\mathbb{R}^n$ serving as a geometric approximation of the enclosed data distribution. It does not coincide with the discrete dataset D or the subset P; instead, the points in P are contained within or approximated by the ellipsoidal region.

The eigenvectors of H^{-1} determine the orientations of the ellipsoid's principal axes, while the eigenvalues control their corresponding lengths. This property enables the granular ellipsoid to align with the principal directions of data variation.

For a data subset $P = \{x_1, x_2, \ldots, x_m\}$ enclosed by a granular ellipsoid, the shape matrix H is commonly estimated as the inverse of the sample covariance matrix [16]:

$$H = \left(\frac{1}{|P|} \sum_{x \in P} (x - c)(x - c)^\top \right)^{-1}. \tag{2.11}$$

Here, $\frac{1}{|P|} \sum_{x\in P}(x-c)(x-c)^\top$ denotes the empirical covariance matrix. This formulation characterizes the anisotropic structure of the enclosed data and provides a geometric description consistent with the Mahalanobis distance metric.

From a theoretical perspective, the granular ellipsoid model is related to the minimum volume enclosing ellipsoid (MVEE) problem [22–24], which seeks the smallest ellipsoid enclosing all points in the dataset D. The canonical MVEE formulation is given by

$$\begin{aligned} \min_{H,c} \quad & -\ln\det(H) \\ \text{s.t.} \quad & (x_i - c)^\top H(x_i - c) \leq 1, \quad \forall x_i \in D, \\ & H \succ 0. \end{aligned} \tag{2.12}$$

Minimizing $-\ln\det(H)$ corresponds to minimizing the volume of the ellipsoid while ensuring that all data points are enclosed. This convex optimization problem provides a rigorous geometric foundation for ellipsoidal representation [25]. Although solving the exact MVEE can be computationally expensive, the covariance-based approximation in Eq (2.11) offers a practical and widely adopted alternative [26].

Let $GE = \{S_1, S_2, \ldots, S_k\}$ denote a granular ellipsoid containing samples from k classes, where S_i represents the subset belonging to class i ($i = 1, 2, \ldots, k$). The label of GE is determined by the majority class [16]:

$$\text{Label}(GE) = \text{Label}\left(\arg\max_{1\le i\le k} |S_i|\right), \tag{2.13}$$

and its purity is defined as [16]:

$$T(GE) = \frac{\max_{1\le i\le k} |S_i|}{\sum_{i=1}^{k} |S_i|}. \tag{2.14}$$

The distance from a point x to a granular ellipsoid is defined as [16]

$$d(x, GE(H, c)) = (x - c)^\top H(x - c), \tag{2.15}$$

which corresponds to the squared Mahalanobis distance and naturally incorporates both feature correlations and scale normalization.

Through iterative splitting and ellipsoid fitting, this process constructs a hierarchical collection of granular ellipsoids, forming a multi-scale geometric representation of the dataset. Compared to spherical granular balls, granular ellipsoids offer greater geometric flexibility and superior representation of anisotropic data structures.

3. The proposed GEDP method

Usually, GB-based DPC algorithms are not skilled at handling anisotropic data, in which case clustering performance is greatly affected. Typically, this performance degradation is caused by the assumption of isotropy. To address this, we propose a new clustering method to deal with anisotropic data by taking granular ellipsoid into consideration [16]. The proposed method consists of three main components: (i) generation of granular ellipsoids, (ii) a granular-ellipsoid-based DPC algorithm, and (iii) point assignment and post-processing.

3.1. Generation of granular ellipsoid

As described in Subsection 2.3, a granular ellipsoid is characterized by its center c and shape matrix H. The center c represents the weighted geometric centroid of the points within the ellipsoid and varies as the ellipsoid evolves. The shape matrix H can be obtained as the inverse of the covariance matrix of the points in the ellipsoid. However, this direct inversion may fail when the covariance matrix is singular or ill-conditioned. To address this issue, we propose a regularized reconstruction of the shape matrix that ensures numerical stability.

Multi-level granular computing methods process high-dimensional data through hierarchical structures, offering a foundation for our hierarchical splitting mechanism. Additionally, adaptive-radius granular sphere methods provide valuable insights for our parameter selection through their handling of non-spherical data.

Definition 3.1 (Shape matrix)**.** *The reconstructed shape matrix, taking $\mathcal{H}$ for example, can be computed by the following formula:*

$$\mathcal{H} = \Sigma^{-1} = \left(\frac{1}{|P|}\sum_{x\in P}(x - c)(x - c)^\top + \epsilon I\right)^{-1}, \tag{3.1}$$

where $\epsilon > 0$ is a parameter with condition $\epsilon \in [10^{-8}, 10^{-3}]$, $\top$ is transpose, and I is the identity matrix.

Comparing Eq (3.1) with Eq (2.11), we can find that the term ϵI not only prevents the singularity of the covariance matrix, but also guarantees the positive definiteness of covariance matrix.

With this, the reconstructed representation of granular ellipsoid can be expressed as

$$\mathcal{GE}(P, \mathcal{H}, c) = \{\, x \in \mathbb{R}^n \;\mid\; (x - c)^\top \mathcal{H}(x - c) \leq 1 \,\}, \tag{3.2}$$

where $P \subseteq D$, the c is the same as that of Eq (2.7).

Evidently, the eigenvalues of the regularized covariance matrix quantify the variance of the granular ellipsoid along the direction of each eigenvector. A larger eigenvalue signifies greater data dispersion in that direction, leading to a longer corresponding axis of the ellipsoid, and vice versa. Therefore, the concept "shape score", i.e., the ratio of the smallest eigenvalue of Σ to the largest, represents the numerical description of a granular ellipsoid's shape.

Definition 3.2 (Shape score)**.** *Given that $\mathcal{GE}(P, \mathcal{H}, c) \in \mathcal{G}$ is a granular ellipsoid, its shape score can be determined by the following formula:*

$$s = \frac{\lambda_{min}(\Sigma)}{\lambda_{max}(\Sigma)}, \tag{3.3}$$

where, λ denotes the eigenvalues of the regularized covariance matrix Σ.

Property 3.1. *Given that $\mathcal{GE}(P, \mathcal{H}, c) \in \mathcal{G}$ is a granular ellipsoid with regularized covariance matrix Σ, the shape score s satisfies:*

$$0 < s \leq 1. \tag{3.4}$$

Proof. According to Eq (3.1), the regularized covariance matrix is expressed as

$$\Sigma = \frac{1}{|P|} \sum_{x \in P} (x - c)(x - c)^\top + \epsilon I,$$

where $\epsilon > 0$, and $|P|$ denotes the number of points in P. The addition of the term ϵI ensures that Σ is positive definite, implying that all its eigenvalues are strictly positive.

Hence,

$$\lambda_{\min}(\Sigma) > 0, \quad \lambda_{\max}(\Sigma) > 0,$$

and, consequently,

$$s = \frac{\lambda_{\min}(\Sigma)}{\lambda_{\max}(\Sigma)} > 0.$$

Furthermore,

$$\lambda_{\min}(\Sigma) \leq \lambda_{\max}(\Sigma),$$

which directly gives

$$s = \frac{\lambda_{\min}(\Sigma)}{\lambda_{\max}(\Sigma)} \leq 1.$$

Therefore, the shape score satisfies $0 < s \leq 1$.

This completes the proof. □

When $s = 1$, i.e., $\lambda_{\min}(\Sigma) = \lambda_{\max}(\Sigma)$, the granular ellipsoid degenerates to a sphere.

As illustrated in Figure 2, the shape score s provides an intuitive geometric interpretation of a granular ellipsoid's shape. When s is close to 1, the granular ellipsoid is nearly spherical; as it approaches 0, the ellipsoid becomes increasingly elongated or flattened. This quantitative measure directly guides our strategy for splitting granules of different shapes. While the shape score effectively captures anisotropy and informs the selection of the splitting method, it alone is insufficient to determine whether subdivision is necessary. Even a highly elongated ellipsoid may correspond to either a single cohesive cluster or multiple disjoint substructures; therefore, relying solely on the shape score is insufficient for making subdivision decisions.

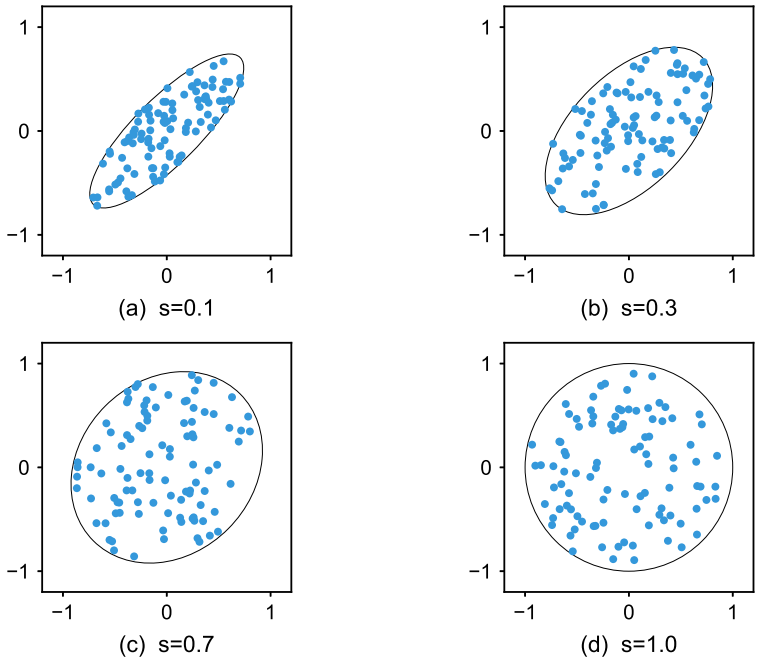


Figure 2. The granular ellipsoids with different s.

To complement the geometric information captured by the shape score, we introduce a compactness metric to measure how tightly data points are distributed within each granular ellipsoid relative to its spatial extent. Before formally defining compactness, we first introduce the notion of the ellipsoid radius, which characterizes its geometric scale.

Definition 3.3 (Radius)**.** *Given that $\mathcal{GE}(P, \mathcal{H}, c) \in \mathcal{G}$ is a granular ellipsoid, its radius can be determined as follows:*

$$r = \max_{x \in P} d_M(x, c), \tag{3.5}$$

where $d_M(x, c) = \sqrt{(x - c)^\top \mathcal{H}(x - c)}$ represents the Mahalanobis distance between data point x and center c.

Definition 3.4 (Compactness)**.** *Given that $\mathcal{GE}(P, \mathcal{H}, c) \in \mathcal{G}$ is a granular ellipsoid, its compactness κ can be defined as*

$$\kappa = \frac{1}{r \cdot |P|} \sum_{x \in P} \sqrt{(x - c)^\top \mathcal{H}(x - c)}. \tag{3.6}$$

Property 3.2. *Given that $\mathcal{GE}(P, \mathcal{H}, c) \in \mathcal{G}$ is a granular ellipsoid, one has that $0 \leq \kappa \leq 1$.*

Proof. From Eq (3.5), we have

$$\sqrt{(x-c)^\top \mathcal{H}(x-c)} \geq 0,$$

and

$$r = \max_{x \in P} \sqrt{(x-c)^\top \mathcal{H}(x-c)} > 0.$$

Then, it follows that:

$$\frac{1}{|P|} \sum_{x \in P} \sqrt{(x-c)^\top \mathcal{H}(x-c)} \geq 0, \quad r > 0,$$

which implies $\kappa \geq 0$.

Furthermore, by the definition of r, for every $x \in P$, we have

$$\sqrt{(x-c)^\top \mathcal{H}(x-c)} \leq r,$$

Summing over all data points gives

$$\sum_{x \in P} \sqrt{(x-c)^\top \mathcal{H}(x-c)} \leq |P| r,$$

and dividing both sides by $|P|$ yields

$$\frac{1}{|P|} \sum_{x \in P} \sqrt{(x-c)^\top \mathcal{H}(x-c)} \leq r,$$

Therefore,

$$\kappa = \frac{\frac{1}{|P|} \sum_{x \in P} \sqrt{(x-c)^\top \mathcal{H}(x-c)}}{r} \leq 1.$$

Hence, the compactness of a granular ellipsoid satisfies $0 \leq \kappa \leq 1$.

This completes the proof. □

A value of κ close to 0 indicates a tight, homogeneous distribution concentrated near the center, whereas κ approaching 1 suggests significant dispersion or heterogeneity. If κ exceeds a chosen threshold, the $\mathcal{GE}$ may be a candidate for further partitioning.

As illustrated in Figure 3, the compactness κ serves as a key criterion for subdivision decisions, complementing the shape score s. Although both granular ellipsoids shown in the figure exhibit the same shape ($s = 0.3$), their compactness values differ significantly. The low-compactness ellipsoid contains tightly clustered points, and thus represents a homogeneous region requiring no further splitting. In contrast, the high-compactness ellipsoid shows dispersed points, revealing potential internal substructures that justify subdivision.

In the GEDP algorithm, subdivision decisions are jointly guided by two criteria. The compactness metric κ first assesses whether further partitioning is necessary; once subdivision is triggered, the shape score s determines the appropriate splitting mechanism. Granular ellipsoids with $s < 0.3$, indicating pronounced anisotropy, are divided using a Gaussian mixture model (GMM) [27–29], while those with $s \geq 0.3$, corresponding to near-spherical shapes, are split using the bisecting k-means algorithm [30].

The empirical threshold of 0.3, established through extensive experiments across diverse datasets, provides an effective trade-off between geometric fidelity and computational efficiency.

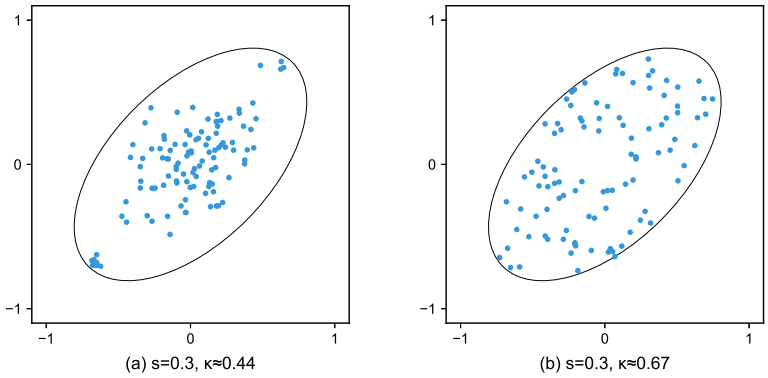


Figure 3. Granular ellipsoids with the same s but different compactness κ.

Furthermore, to maintain scalability and prevent over-partitioning, we incorporate a hierarchical depth control mechanism. Each granular ellipsoid is assigned a level index, and subdivision proceeds only when the current level is below the maximum depth $l_{\max}$. This mechanism preserves the structural hierarchy and ensures convergence of the splitting process. The complete pseudocode for granular ellipsoid generation is provided in Algorithm 1.

Algorithm 1 Generation of granular ellipsoid.

Require: Dataset D, $m_{\min}$, $l_{\max}$, τ_c
Ensure: $\mathcal{G}$

Initialize root ellipsoid $\mathcal{GE}_0$ with $P = D$, $l = 0$
$\mathcal{G} \leftarrow \emptyset$, $Q \leftarrow \{\mathcal{GE}_0\}$
while $Q \neq \emptyset$ **do**
    $\mathcal{GE} \leftarrow Q.pop()$
    if $|P(\mathcal{GE})| > m_{\min}$ **and** $l(\mathcal{GE}) < l_{\max}$ **and** $\kappa(\mathcal{GE}) > \tau_c$ **then**
        Compute $s(\mathcal{GE}) \leftarrow \lambda_{\min}(\Sigma)/\lambda_{\max}(\Sigma)$
        if $s(\mathcal{GE}) < 0.3$ **then**
            $\{P_1, P_2\} \leftarrow$ SplitWithGMM($P(\mathcal{GE})$)
        else
            $\{P_1, P_2\} \leftarrow$ SplitWithKMeans($P(\mathcal{GE})$)
        end if
        Create sub-ellipsoids $\mathcal{GE}_1$, $\mathcal{GE}_2$ from P_1, P_2
        $l(\mathcal{GE}_1), l(\mathcal{GE}_2) \leftarrow l(\mathcal{GE}) + 1$
        $Q.push(\mathcal{GE}_1)$, $Q.push(\mathcal{GE}_2)$
    else
        $\mathcal{G}.add(\mathcal{GE})$
    end if
end while
return $\mathcal{G}$

It is worth noting that a unimodal but heavy-tailed distribution may yield a compactness value κ that is relatively low due to distant outliers, potentially exceeding the splitting threshold τ_c. In such cases, splitting the granule would be inappropriate because the data still form a single cluster. However, empirical observations on typical heavy-tailed distributions in real-world data indicate that the average distance from points to the center remains substantially smaller than the maximum distance; consequently, κ rarely exceeds τ_c (default 0.7). Moreover, even if a heavy-tailed granule is inadvertently split, the subsequent point-level assignment and cluster merging step (Section 3.4) can effectively recombine over-segmented sub-clusters by merging adjacent centers whose distance falls below τ_m. As a result, the overall clustering quality remains robust, as demonstrated by the strong performance of GEDP across diverse datasets (Section 5). The current design thus strikes a practical balance between simplicity and effectiveness, without requiring an explicit multimodality test.

3.2. Density of granular ellipsoid and distance between granular ellipsoids

In physics, density describes the amount of matter concentrated within a unit volume. Analogously, in the GEDP algorithm, the density of a granular ellipsoid measures the compactness of the data points it encloses. Based on this analogy, we define the density of a granular ellipsoid as follows.

Definition 3.5 (Density of $\mathcal{GE}$)**.** *Given that $\mathcal{GE}(P, \mathcal{H}, c) \in \mathcal{G}$ is a granular ellipsoid, its density ρ can be expressed as*

$$\rho = \frac{|P|}{V}, \tag{3.7}$$

where V denotes the volume of the $\mathcal{GE}$, given by

$$V = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)} \cdot \frac{1}{\sqrt{\det(\mathcal{H})}}, \tag{3.8}$$

where $\Gamma(\cdot)$ denotes the Gamma function, and n represents the dimensionality of the data space.

To enable a meaningful comparison of densities among granules of different scales, let $\mathcal{G} = \{\mathcal{GE}_1, \mathcal{GE}_2, \cdots, \mathcal{GE}_t\}$ denote the set of granular ellipsoids, and let $\rho(\mathcal{GE}_i)$ represent the density of $\mathcal{GE}_i$ for $i = 1, 2, \cdots, t$. The normalized density is defined as

$$\rho_{\text{norm}}(\mathcal{GE}_i) = \frac{\rho(\mathcal{GE}_i)}{\max\limits_{1 \le j \le t} \rho(\mathcal{GE}_j)}. \tag{3.9}$$

This normalization maps all density values to the range [0, 1], preserves their relative ordering, and removes scale effects among granules of different sizes. As a result, it highlights relative compactness and enables consistent identification of cluster centers in the subsequent density-peak clustering stage. The combined decision measure for each granule is defined as

$$\gamma(\mathcal{GE}_i) = \rho_{\text{norm}}(\mathcal{GE}_i) \cdot \delta_{\text{norm}}(\mathcal{GE}_i), \tag{3.10}$$

which depends on the scale-invariant quantities obtained through normalization. The detailed definition of $\delta_{\text{norm}}(\mathcal{GE}_i)$ and its role in ellipsoid-level density-peak clustering are provided in Section 3.3.

Definition 3.6 (Distance between $\mathcal{GE}$s)**.** *Given that $\mathcal{GE}_i(P_i, \mathcal{H}_i, c_i)$ and $\mathcal{GE}_j(P_j, \mathcal{H}_j, c_j)$ are two granular ellipsoids, the distance $d(\mathcal{GE}_i, \mathcal{GE}_j)$ between them can be computed as follows:*

$$d(\mathcal{GE}_i, \mathcal{GE}_j) = 1.5 \cdot (1 + |s_i - s_j|) \cdot \sqrt{(c_i - c_j)^\top \mathcal{H}_{avg}(c_i - c_j)}, \tag{3.11}$$

where

$$\mathcal{H}_{avg} = (\mathcal{H}_i + \mathcal{H}_j)/2, \tag{3.12}$$

denotes the average shape matrix, and s_i, s_j are their shape scores, respectively.

The average shape matrix $\mathcal{H}_{\text{avg}}$ symmetrizes the Mahalanobis distance, ensuring equal consideration of both ellipsoids' geometries. The penalty term $(1 + |s_i - s_j|)$ captures shape dissimilarity, reflecting that ellipsoids with divergent anisotropies should be considered more distant.

The constant factor 1.5 in Eq (3.11) enhances separation between ellipsoids. A sensitivity analysis (Section 5.7) shows that varying it within $[1.0, 2.0]$ yields ACC and NMI fluctuations below ± 0.01, confirming robustness. Hence, 1.5 serves as a robust default across diverse datasets.

A detailed rationale for this distance design, including its conceptual advantages over probabilistic distances and its computational efficiency, is provided in Section 3.3.1.

3.3. Granular ellipsoid based DPC algorithm

The fundamental assumption of the DPC algorithm is that cluster centers possess two key properties: they are surrounded by neighbors with lower local density, and they are relatively far from any point with higher local density. To quantify these properties, the original DPC algorithm defines a density measure ρ and a δ-distance for each data point. In adapting this principle to the granular ellipsoid level, we extend the concept of δ-distance accordingly. Specifically, for the GEDP algorithm, we define the δ-distance for each $\mathcal{GE}$ based on the distance metric $d(\mathcal{GE}_i, \mathcal{GE}_j)$ between granular ellipsoids. Formally, the δ-distance for a given $\mathcal{GE}_i$ is defined as follows.

Definition 3.7 (The δ-distance of $\mathcal{GE}$)**.** *Given that $\mathcal{GE}_i(P_i, \mathcal{H}_i, c_i) \in \mathcal{G}$ is a granular ellipsoid, its δ-distance and the nearest higher-density $\mathcal{GE}$ are computed as follows:*

$$\delta(\mathcal{GE}_i) = \begin{cases} \min\limits_{\mathcal{GE}_j \in \mathcal{G},\ \rho(\mathcal{GE}_j) > \rho(\mathcal{GE}_i)} \mathrm{d}(\mathcal{GE}_i, \mathcal{GE}_j), & \text{if } \rho(\mathcal{GE}_i) < \max \rho, \\ \max\limits_{\mathcal{GE}_j \in \mathcal{G}} \delta(\mathcal{GE}_j), & \text{if } \rho(\mathcal{GE}_i) = \max \rho, \end{cases} \tag{3.13}$$

$$N(\mathcal{GE}_i) = \begin{cases} \arg \min\limits_{\mathcal{GE}_j \in \mathcal{G},\ \rho(\mathcal{GE}_j) > \rho(\mathcal{GE}_i)} \mathrm{d}(\mathcal{GE}_i, \mathcal{GE}_j), & \text{if } \rho(\mathcal{GE}_i) < \max \rho, \\ \mathcal{GE}_i, & \text{if } \rho(\mathcal{GE}_i) = \max \rho, \end{cases} \tag{3.14}$$

where $\rho(\mathcal{GE}_j)$ is the density of $\mathcal{GE}_j$ and $\max \rho$ *is the maximum density value among the $\mathcal{GE}$s. Equation* (3.13) *means that for the $\mathcal{GE}$ with density less than* $\max \rho$*, its δ-distance is the distance between it and its nearest $\mathcal{GE}$s having larger density, and for the $\mathcal{GE}$ having the largest density, its δ-distance is set to the maximum δ-distance.*

To ensure comparability across different scales and to facilitate consistent identification of cluster centers, the δ-distance of each granular ellipsoid is normalized as

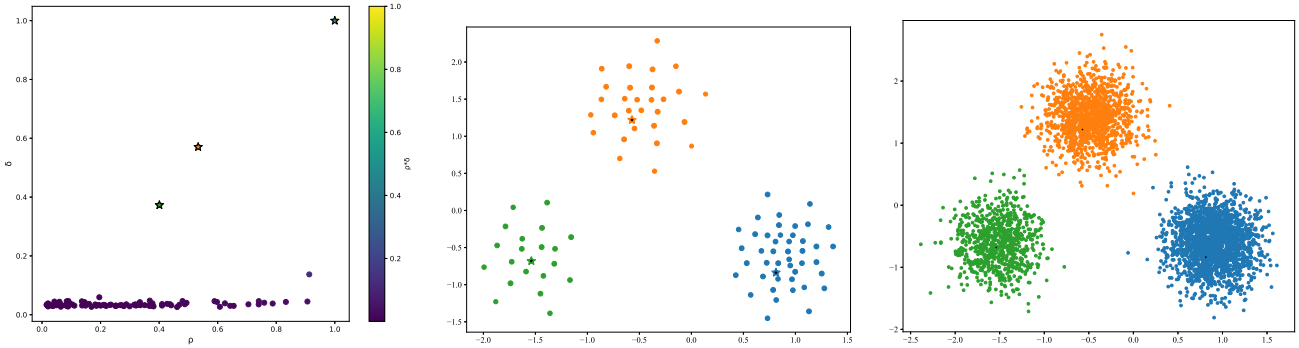
$$\delta_{norm}(\mathcal{GE}_i) = \frac{\delta(\mathcal{GE}_i)}{\max\limits_{1 \le j \le t} \delta(\mathcal{GE}_j)}, \tag{3.15}$$

where $\delta(\mathcal{GE}_i)$ denotes the δ-distance of the i-th granular ellipsoid and t is the total number of granules. This normalization maps all δ-values to the range $[0, 1]$, ensuring scale invariance and balanced contribution when combined with $\rho_{norm}(\mathcal{GE}_i)$ for cluster center determination.

Based on the density and δ-distance of each $\mathcal{GE}$, a decision graph can be constructed. Granular ellipsoids characterized by both relatively high density and large δ-distance are selected as cluster centers. Specifically, a large δ-distance indicates that no $\mathcal{GE}$ with higher density exists in its vicinity, and that it maintains a substantial distance from all other higher-density $\mathcal{GE}$s. Consequently, the $\mathcal{GE}$s selected from the decision graph fully satisfy the core assumption of DPC.

The remaining $\mathcal{GE}$s are then assigned to their nearest higher-density neighbors according to Eq (3.14). Finally, the clustering result for the original dataset is derived by extending the clustering results obtained on the $\mathcal{GE}$s.

The clustering process is illustrated in Figure 4. Specifically, Figure 4(a) shows the decision graph constructed from ρ_{norm} and δ_{norm}, Figure 4(b) presents the clustering results at the granular-ellipsoid level (each node represents a granular ellipsoid), and Figure 4(c) depicts the corresponding clustering outcome on the original data, derived through granular-ellipsoid-based label propagation. Algorithm 2 summarizes the primary clustering procedure at the granular ellipsoid level.



(a) Decision graph **(b)** Clustering results on $\mathcal{GE}$s **(c)** Final clustering results

Figure 4. Clustering $\mathcal{GE}$s with the DPC algorithm.

Algorithm 2 Granular ellipsoid density peaks clustering.

Require: $\mathcal{G}$
Ensure: $\mathcal{L}_{\mathcal{G}}, C$

for each $\mathcal{GE}_i \in \mathcal{G}$ **do**
  $\rho(\mathcal{GE}_i) \leftarrow |P(\mathcal{GE}_i)|/V(\mathcal{GE}_i)$
  $\delta(\mathcal{GE}_i) \leftarrow \min_{\rho(\mathcal{GE}_j)>\rho(\mathcal{GE}_i)} d(\mathcal{GE}_i, \mathcal{GE}_j)$
  Normalize $\rho_{\text{norm}}(\mathcal{GE}_i)$ and $\delta_{\text{norm}}(\mathcal{GE}_i)$
  $\gamma(\mathcal{GE}_i) \leftarrow \rho_{\text{norm}}(\mathcal{GE}_i) \cdot \delta_{\text{norm}}(\mathcal{GE}_i)$
end for
$C \leftarrow \{\mathcal{GE}_c \mid \gamma(\mathcal{GE}_c) > \gamma_{\text{th}}\}$
Sort $\mathcal{G}$ by $\rho(\mathcal{GE})$ in descending order
for each $\mathcal{GE}_i \in \mathcal{G} \setminus C$ **do**
  $\mathcal{L}_{\mathcal{G}}(\mathcal{GE}_i) \leftarrow \mathcal{L}_{\mathcal{G}}(N(\mathcal{GE}_i))$
end for
return $\mathcal{L}_{\mathcal{G}}, C$

3.3.1. Rationale of the proposed distance measure

The distance defined in Eq (3.11) is specifically designed for granular ellipsoids, which are geometric constructs rather than probabilistic representations. Each granular ellipsoid $\mathcal{GE}(P, \mathcal{H}, c)$ is a deterministic region enclosing a set of data points, derived from covariance estimation. It does not imply a probability distribution. Therefore, standard probabilistic distances such as the Wasserstein distance, symmetrized Kullback-Leibler divergence, or Hellinger distance, which are widely used to compare Gaussian distributions, are conceptually mismatched for our geometric ellipsoids. Applying them would require treating each ellipsoid as a Gaussian, introducing assumptions (e.g., normality) that do not hold in the GEDP framework.

In contrast, Eq (3.11) is purpose-built for geometric ellipsoids. It combines two essential components:

- **Centroid separation** via a symmetrized Mahalanobis distance using the average shape matrix $\mathcal{H}_{\text{avg}} = (\mathcal{H}_i + \mathcal{H}_j)/2$. This naturally accounts for the orientation and scale of the ellipsoids.
- **Shape dissimilarity** via the penalty term $1 + |s_i - s_j|$, where s_i and s_j are the shape scores defined in Eq (3.3). This term explicitly penalizes differences in anisotropy, ensuring that ellipsoids with divergent shapes are considered more distant.

This design aligns with the core principle of density peaks clustering: cluster centers should be granules that are both dense and isolated from others in terms of location and shape. Moreover, the computational cost of Eq (3.11) is $O(n^2)$ per pair, significantly lower than that of probabilistic distances which often require eigenvalue decomposition ($O(n^3)$) and are thus prohibitive for large-scale data. As we will demonstrate in Section 5.4, this choice yields superior accuracy on anisotropic datasets while maintaining high scalability.

3.4. Point assignment and post-processing

After completing clustering at the granular-ellipsoid level, the results are mapped back to the original data space. Each data point is assigned to the granular ellipsoid it most likely belongs to, based on the following distance definition. Algorithm 3 concretizes these abstract results onto the original data points and optimizes the final output.

Definition 3.8 (Distance from a point to $\mathcal{GE}$)**.** *Given that $\mathcal{GE}$ is a granular ellipsoid and $x \in D$ is a point , the distance from x to $\mathcal{GE}$ is defined as*

$$d(x, \mathcal{GE}(P, \mathcal{H}, c)) = (x - c)^\top \mathcal{H}(x - c), \tag{3.16}$$

which corresponds to the squared Mahalanobis distance. The square root is omitted since only relative distances are required for nearest-ellipsoid selection and boundary determination, consistent with the ellipsoid constraint $(x - c)^\top \mathcal{H}(x - c) \leq 1$.

Note that the square root in the standard Mahalanobis distance is omitted, and the squared form is used instead. Since the square root is a monotonic transformation, this modification does not affect distance ranking or neighborhood determination. Accordingly, the threshold τ_b is interpreted in terms of the squared Mahalanobis distance. The squared formulation is preferred for improved computational efficiency and numerical stability.

For each data point x, the nearest granular ellipsoid $\mathcal{GE}_{\text{closest}}$ is identified by minimizing the distance $d(x, \mathcal{GE})$, with the corresponding minimum distance denoted as $d_{\min}$. The label of x is then assigned according to the following rules: If $d_{\min} \leq 1$, the point lies inside the nearest ellipsoid and directly inherits its label; if $1 < d_{\min} \leq \tau_b$, it is regarded as a boundary point and assigned to the same ellipsoid to maintain label continuity; otherwise, when $d_{\min} > \tau_b$, the point is treated as an exterior sample and assigned to the nearest cluster center in C. After point assignment, adjacent clusters whose center distance is below τ_m are merged, and clusters with size below τ_s are removed or reclassified as noise. This process ensures label consistency and eliminates spurious micro-clusters. The complete procedure for point assignment and post-processing is summarized in Algorithm 3, and the overall GEDP algorithm is outlined in Algorithm 4.

Algorithm 3 Point assignment and post-processing.

Require: Dataset D, $\mathcal{G}$ with labels $\mathcal{L}_{\mathcal{G}}$, τ_b, τ_m, τ_s
Ensure: Final point-level labels $\mathcal{L}$

for each $x \in D$ **do**
  $\mathcal{GE}_{\text{closest}} \leftarrow \arg\min_{\mathcal{GE} \in \mathcal{G}} d(x, \mathcal{GE})$
  $d_{\min} \leftarrow d(x, \mathcal{GE}_{\text{closest}})$
  if $d_{\min} \leq 1$ **then**
    $\mathcal{L}(x) \leftarrow \mathcal{L}_{\mathcal{G}}(\mathcal{GE}_{\text{closest}})$
  else if $1 < d_{\min} \leq \tau_b$ **then**
    $\mathcal{L}(x) \leftarrow \mathcal{L}_{\mathcal{G}}(\mathcal{GE}_{\text{closest}})$
  else
    $\mathcal{L}(x) \leftarrow$ nearest cluster center in C
  end if
end for
Merge clusters if $d(c_i, c_j) < \tau_m$
Remove clusters with $|C_k| < \tau_s$
return $\mathcal{L}$

Algorithm 4 GEDP Algorithm.

Require: Dataset D
Ensure: Final point-level clustering labels $\mathcal{L}$

Construct hierarchical granular ellipsoids using **Algorithm 1**;
Perform ellipsoid-level density peaks clustering using **Algorithm 2**;
Assign point-level labels and conduct post-processing using **Algorithm 3**;
return $\mathcal{L}$

3.5. Complexity analysis

The computational complexity of GEDP is analyzed across three stages: granular ellipsoid generation, ellipsoid-level clustering , and point assignment.

In the first stage, a dataset of size m and dimension n is adaptively partitioned into l granular

ellipsoids ($l \ll m$). Each point undergoes $O(\log l)$ splitting steps, and each step requires $O(n^3)$ operations for covariance estimation and matrix inversion. Thus, the total complexity is $O(mn^3 \log l)$, which significantly reduces the $O(m^2)$ cost of classical DPC.

In the clustering stage, operations are conducted on only l ellipsoids. Constructing the inter-ellipsoid distance matrix requires $O(l^2n^2)$, and the density-peaks procedure requires $O(l^2)$ comparisons.

In the assignment and post-processing stage, each of the m data points is assigned to its nearest ellipsoid or center via Mahalanobis distance computation, resulting in $O(mln^2)$ complexity. Post-processing operations, such as merging and removing small ellipsoids, add $O(l^2)$, which is negligible compared with point assignment.

Since the value of l is typically on the order of $O(\sqrt{m})$ in practical scenarios, the overall computational complexity is governed by the $O(mn^3 \log l)$ term. This results in a near-linear scaling and thereby avoids the quadratic bottleneck found in traditional density-based clustering methods.

The space complexity is $O(mn + ln^2 + l^2)$, covering data storage, ellipsoid shape matrices, and the inter-ellipsoid distance matrix. Because $l \ll m$, the memory usage remains efficient even for large-scale datasets.

4. Experimental components

In this section, we introduce several experimental components, including the datasets, evaluation metrics, and compared algorithms. The specified experimental environment we used is listed in Table 2.

Table 2. Experimental environment.

Parameter	Parameter value
RAM	32 GB
Speed	2.5 GHz
Programming environment	MATLAB R2024a and PyCharm 2024.2.5
CPU	Intel Core Ultra9 185H
Operating system	Windows 11

4.1. Datasets

To assess the effectiveness of our method, we use three categories of datasets for performance evaluation: synthetic, real-world, and large-scale datasets. Most synthetic datasets are two-dimensional, making them suitable for visualization and effective as evaluation tools for clustering algorithms. They are primarily used to test the performance of algorithms in identifying different types of clusters. Real-world datasets exhibit higher complexity, with varying dimensions and scales, enabling a more comprehensive evaluation of the algorithm's practical applicability. Large-scale datasets are employed to evaluate the scalability and computational efficiency of the proposed algorithm. These datasets contain hundreds of thousands of samples and often feature diverse distributions. They serve to verify whether the algorithm can maintain high clustering accuracy and robustness while significantly reducing computational cost as data volume increases. These datasets used in the experiments are detailed in Tables 3–5, respectively.

Table 3. Information of synthetic datasets.

Datasets	Instances	Clusters	Dimensions	Source
D1	1053	5	2	[8]
D2	788	7	3	[1]
D3	6700	5	2	[8]
D4	1000	5	2	[1]
D5	4000	5	2	[1]
D6	32000	5	2	[1]

Table 4. Information of real datasets.

Datasets	Instances	Dimensions	Clusters	Source
iris	150	4	3	UCI Repository
wine	178	13	3	UCI Repository
seeds	210	7	3	UCI Repository
semeion	1593	265	2	UCI Repository
landsat	2000	2	6	UCI Repository
segment	2310	3	7	UCI Repository
msplice	3175	2	3	UCI Repository
ls	6435	36	6	UCI Repository
svmguide1	7089	5	2	LIBSVM
mushrooms	8124	2	2	UCI Repository
pendigits	10992	16	10	UCI Repository

Table 5. Information of large-scale and high-dimensional datasets.

Dataset	Instances	Dimensions	Clusters	Source
TS500	500K	2	5	[14]
TB	500K	2	-	[14]
ijcnn	141,691	23	2	UCI Repository
MNIST	70,000	784	10	[31]
500D	10,000	500	-	Synthetic
600D	10,000	600	-	Synthetic

4.2. Evaluation metrics

We select four widely used evaluation metrics to assess the quality of the clustering results: accuracy (ACC) [32], normalized mutual information (NMI) [33], and adjusted rand index (ARI) [33]. Among these, ACC and NMI range from 0 to 1, while ARI ranges from -1 to 1. In all cases, higher values indicate better clustering performance. Moreover, all experiments are repeated 10 independent times with different random seeds. For each run, the same seeds are applied to all compared methods to ensure fairness.

4.3. Compared algorithms

In the experiments, we compare seven clustering algorithms: k-means, ball k-means, DPC, FastDPeak, DPC-KNN-PCA, DLORE-DP, and GB-DP. DPC is used as the baseline. k-means and ball k-means are distance-based methods; ball k-means improves the original version by using spherical granules to speed up clustering. FastDPeak is a faster variant of DPC that estimates density more efficiently. DPC-KNN-PCA combines k-nearest-neighbor density estimation with PCA to handle high-dimensional data. DLORE-DP reduces local outliers and adjusts density adaptively. GB-DP replaces point-level processing with granular balls, which makes the algorithm faster and more robust to noise. The descriptions of each algorithm are as follows.

k-means [30]: The k-means algorithm iteratively assigns points to the nearest cluster center and updates the center's position until convergence.

ball k-means [34]: The ball k-means algorithm extends k-means by introducing hyperspherical granules (balls) to accelerate convergence and reduce the influence of noise.

DPC [1]: This algorithm identifies high-density regions as cluster centers by combining density and distance and then performs clustering.

FastDPeak [4]: The FastDPeak algorithm is an improved version of DPC that accelerates the clustering process by using approximate local density estimation and efficient neighbor search.

DPC-KNN-PCA [35]: The DPC-KNN-PCA algorithm combines DPC with k-nearest-neighbor-based density estimation and principal component analysis.

DLORE-DP [8]: The DLORE-DP algorithm integrates local outlier reduction and density refinement mechanisms into DPC.

GB-DP [14]: The granular-ball density peaks (GB-DP) algorithm replaces point-level processing in DPC with granule-level operations using spherical balls.

5. Results and analysis

In this section, we comprehensively evaluate the performance of the GEDP method on synthetic, real-world and large-scale datasets. In addition, to facilitate the recording of experimental results in tables, we use the following abbreviations: k-means as KM, ball k-means as BKM, DPC-KNN-PCA as DKP, FastDPeak as FDP, and DLORE-DP as DDP.

5.1. Clustering on synthetic datasets

To assess the performance of the proposed GEDP algorithm, we conducted extensive experiments on six synthetic datasets exhibiting diverse geometric characteristics and noise conditions. GEDP was compared with seven representative clustering baselines: KM, BKM, DPC, DKP, FDP, DDP, and GB-DP. Table 3 summarizes the key properties of these datasets, which include both spherical and anisotropic cluster structures. Except for D2, which is three-dimensional, all datasets are two-dimensional, allowing intuitive visualization of clustering outcomes.

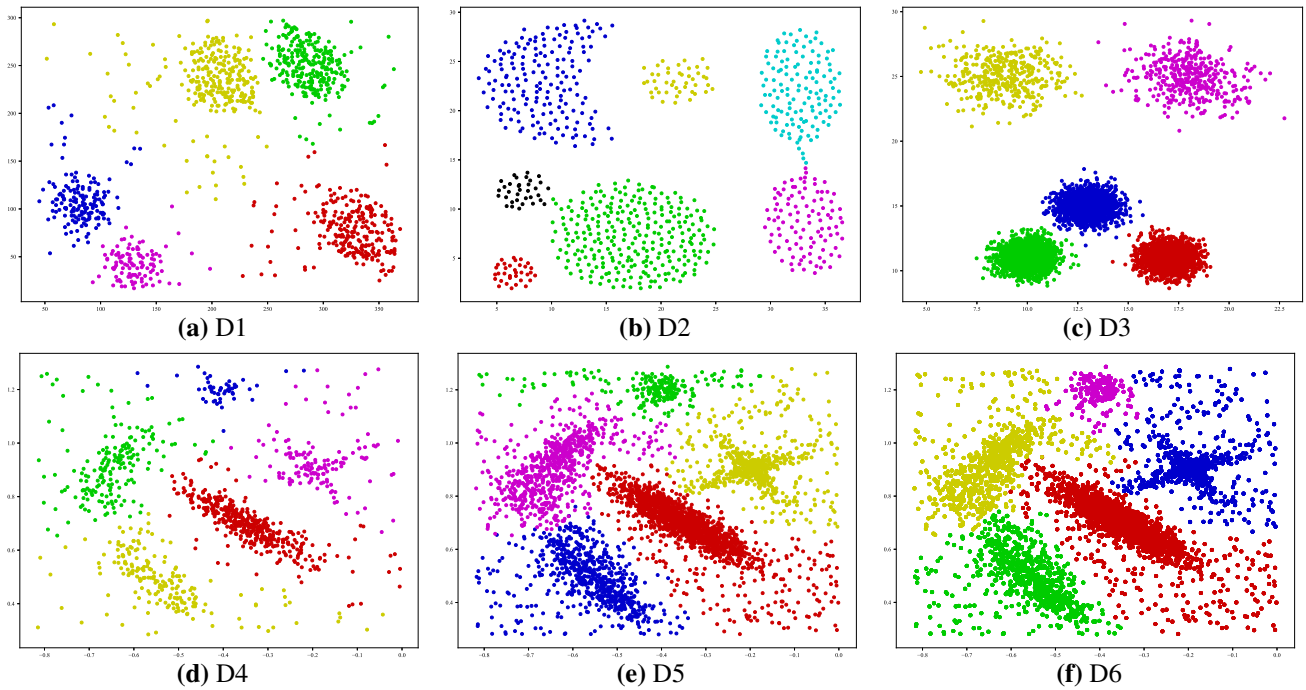
For KM and BKM, the number of clusters k was prespecified according to the ground-truth values to ensure fair comparison. Following the original DPC methodology [1], the cutoff distance d_c was set to 2% of the maximum pairwise distance. For FDP, which exhibits significant sensitivity to the number of nearest neighbors and cluster count, we adopted the recommendation of Chen et al. [4] by

fixing the nearest-neighbor parameter to 11 and setting the cluster number equal to the ground-truth value. Notably, while GB-DP requires manual identification of cluster centers through decision graph inspection, our proposed GEDP method automatically detects cluster centers and adaptively determines granular ellipsoid splitting strategies without human intervention.

Figures 5–12 present the visual clustering results of all competing algorithms. Due to the three-dimensional nature of D2, we employed dimensionality reduction techniques to facilitate effective visualization of clustering outcomes. The computational efficiency of all methods are quantitatively reported in Table 6.

As shown in Table 6, GEDP demonstrates substantial computational advantages over alternative approaches, particularly on larger and more complex datasets. On Dataset 6, containing 32,000 data points, GEDP achieves exceptional efficiency, approximately 635 times faster than DDP and about 40% faster than GB-DP. While traditional methods including KM, DPC, FDP, and DKP exhibit shorter runtimes on smaller datasets (D1–D5), their clustering quality proves unreliable in practical scenarios. For instance, KM produces accurate clustering only on D3 featuring spherical clusters, while all these baseline methods display significant limitations when handling non-spherical cluster structures.

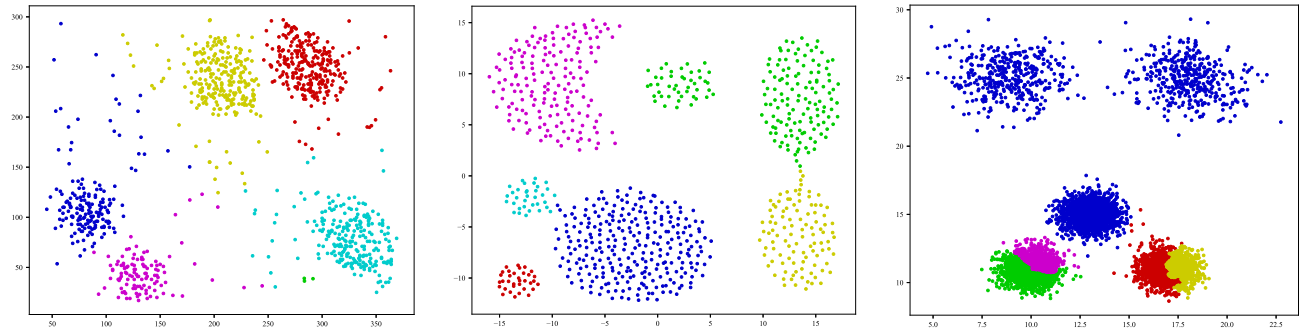
To complement the runtime comparison in Table 6, we report clustering accuracy (ACC), normalized mutual information (NMI), and adjusted rand index (ARI) for all synthetic datasets in Table 7. These quantitative results are consistent with the visualizations in Figures 5–12 and further confirm the effectiveness of GEDP. For instance, on the most challenging anisotropic dataset D6, GEDP achieves the highest ACC (0.9800), NMI (0.9700), and ARI (0.9650), significantly outperforming all competing methods. In contrast, algorithms based on spherical assumptions (e.g., DPC, GB-DP) exhibit markedly lower metrics on non-spherical datasets, aligning with their visual clustering errors shown in Figures 5–12.



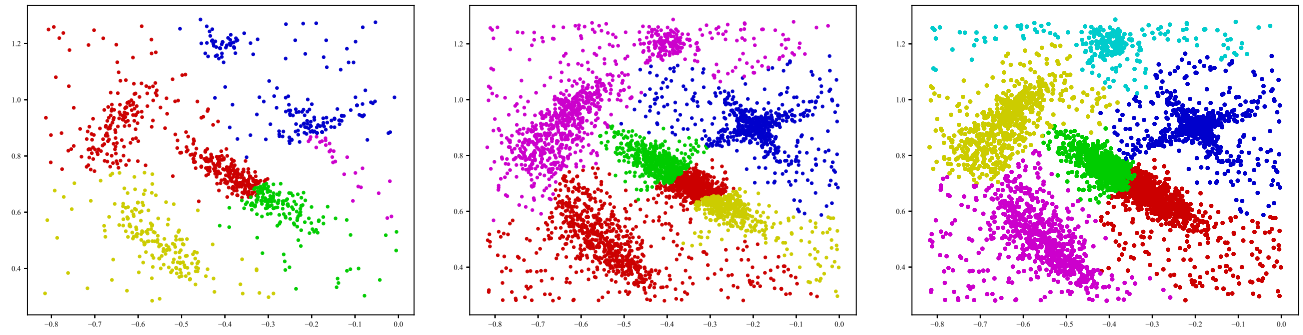
(a) D1 **(b)** D2 **(c)** D3

(d) D4 **(e)** D5 **(f)** D6

Figure 5. Clustering results of GEDP.

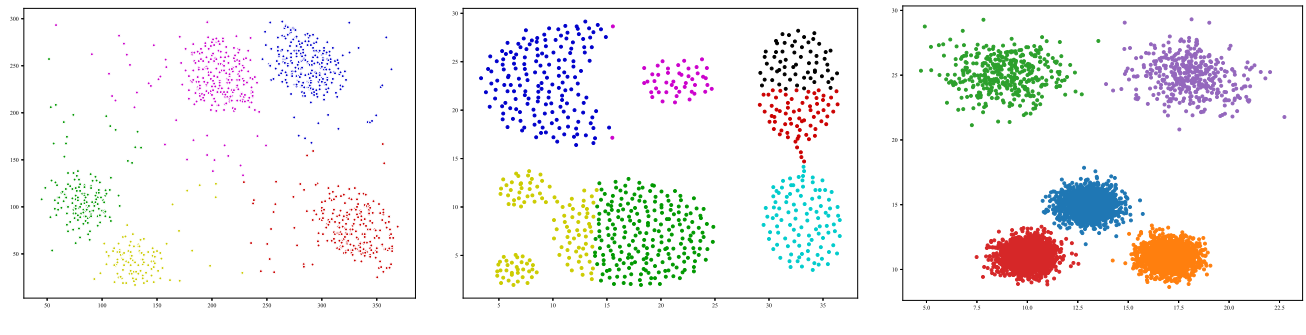


(a) D1 **(b)** D2 **(c)** D3

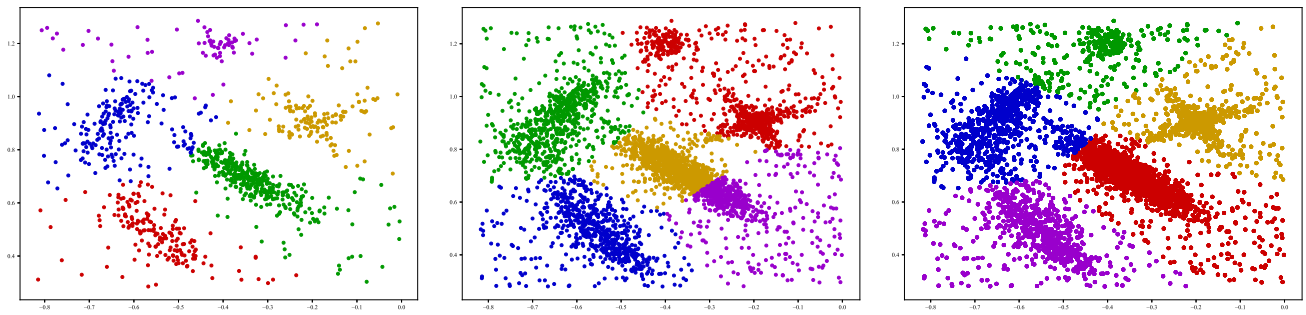


(d) D4 **(e)** D5 **(f)** D6

Figure 6. Clustering results of GB-DP.

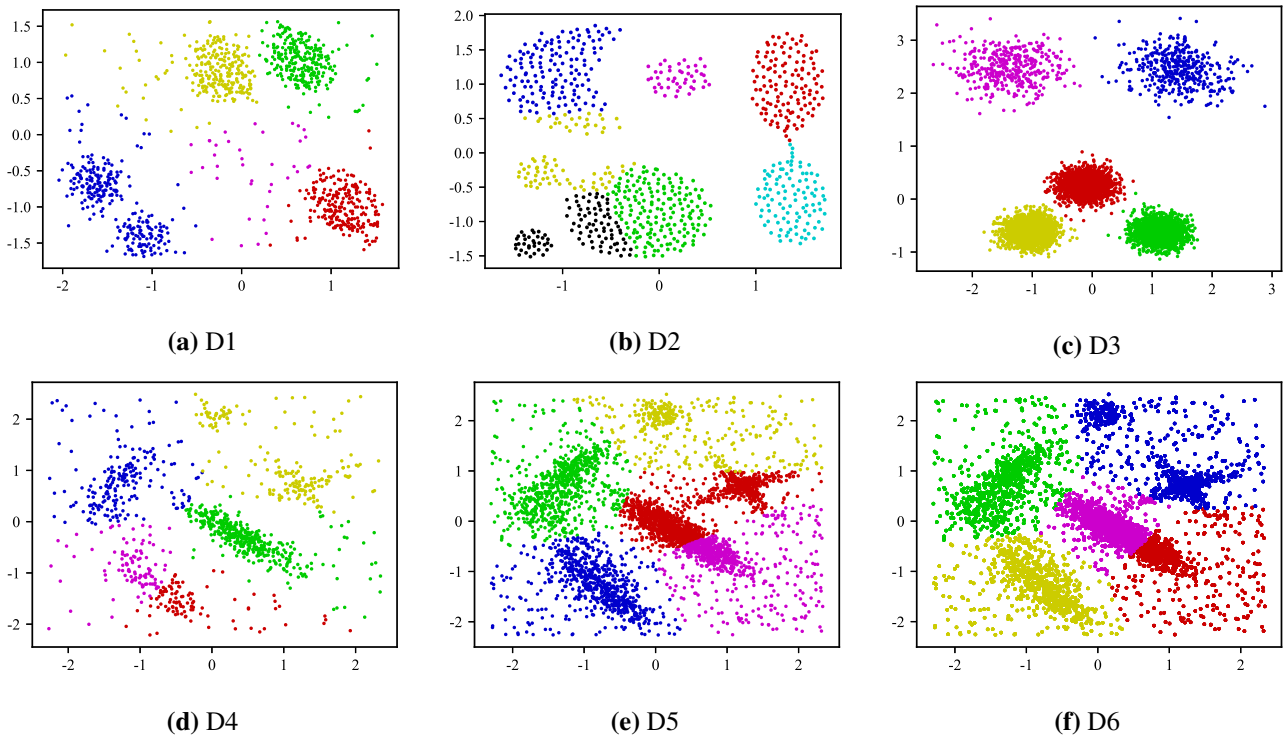


(a) D1 **(b)** D2 **(c)** D3



(d) D4 **(e)** D5 **(f)** D6

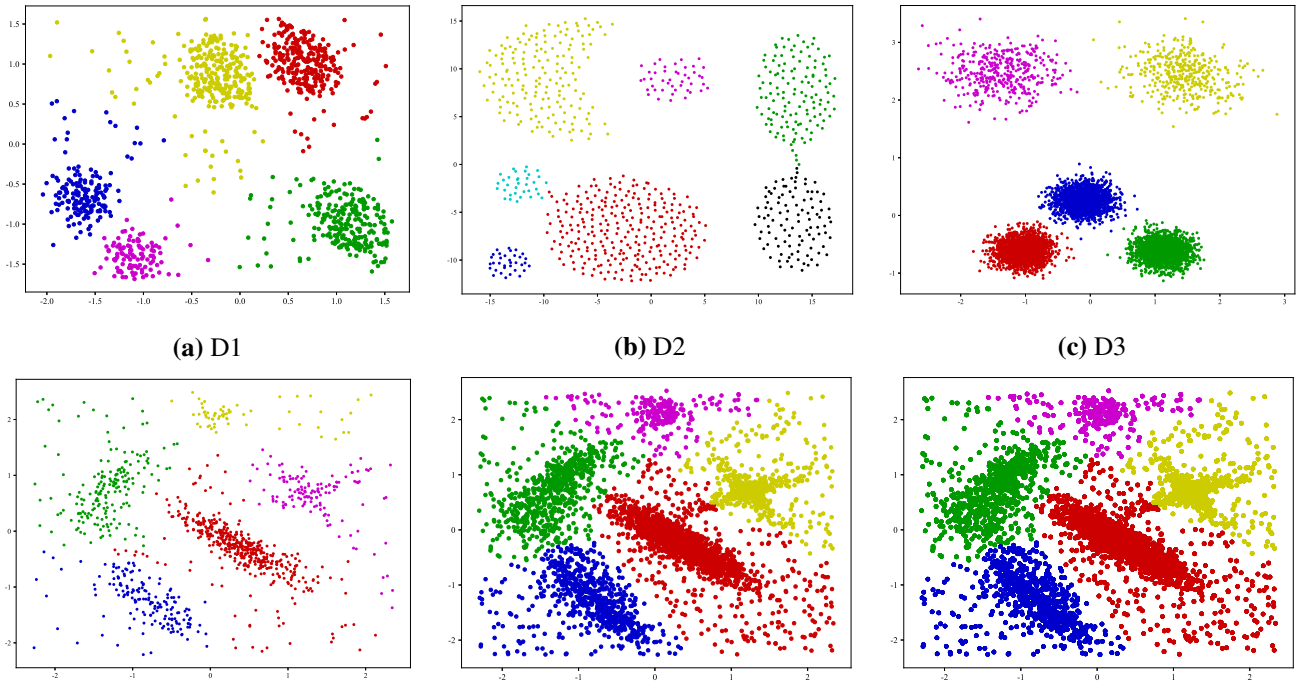
Figure 7. Clustering results of k-means.



(a) D1 **(b)** D2 **(c)** D3

(d) D4 **(e)** D5 **(f)** D6

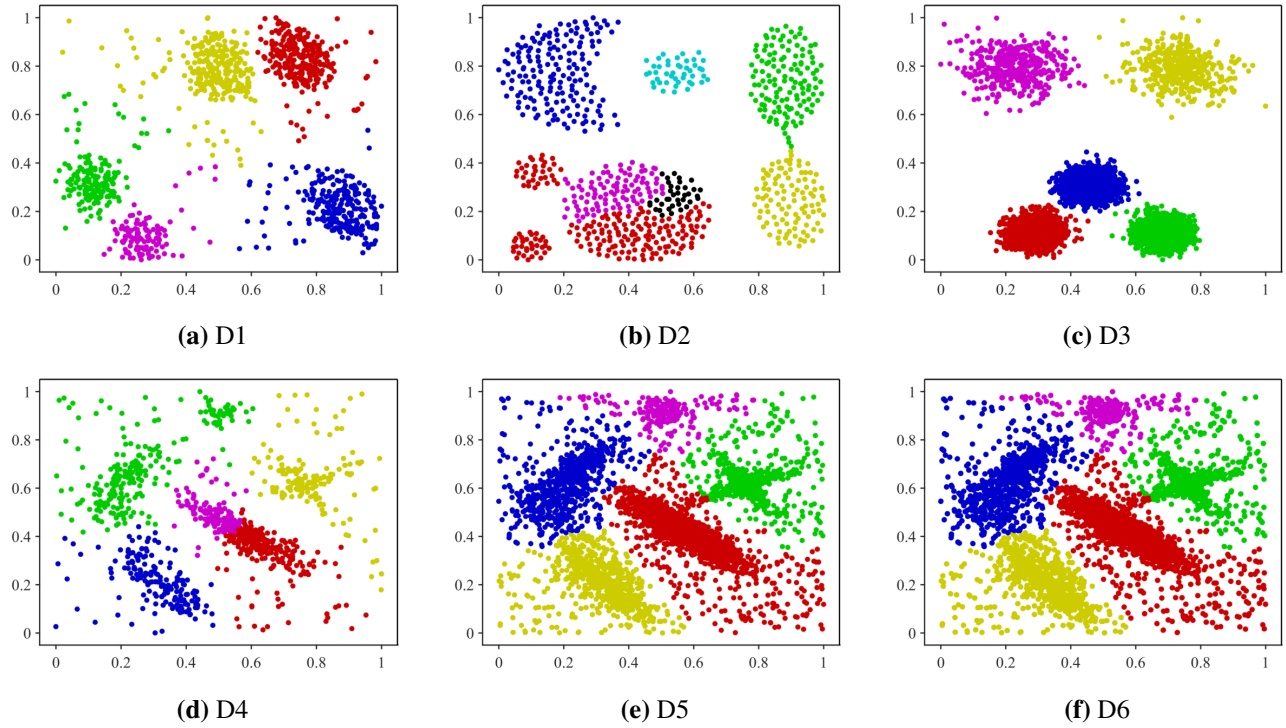
Figure 8. Clustering results of ball k-means.



(a) D1 **(b)** D2 **(c)** D3

(d) D4 **(e)** D5 **(f)** D6

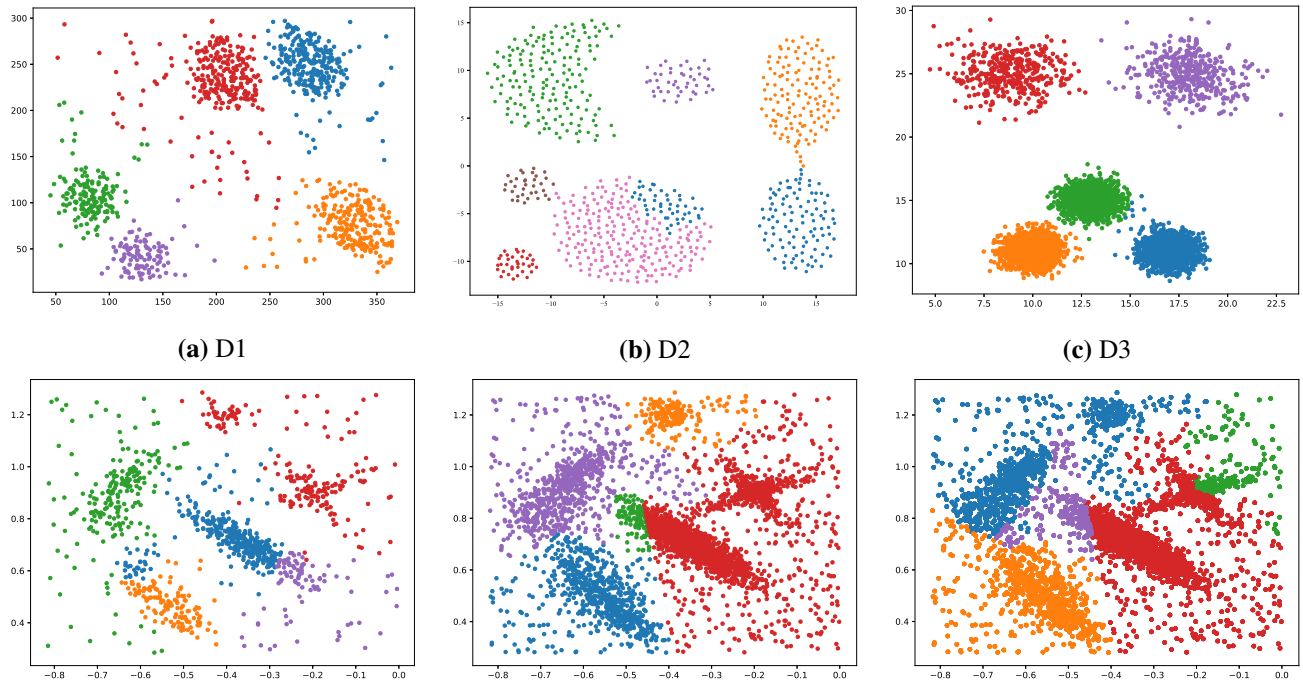
Figure 9. Clustering results of DPC.



(a) D1 **(b)** D2 **(c)** D3

(d) D4 **(e)** D5 **(f)** D6

Figure 10. Clustering results of DPC-KNN-PCA.



(a) D1 **(b)** D2 **(c)** D3

(d) D4 **(e)** D5 **(f)** D6

Figure 11. Clustering results of FastDPeak.

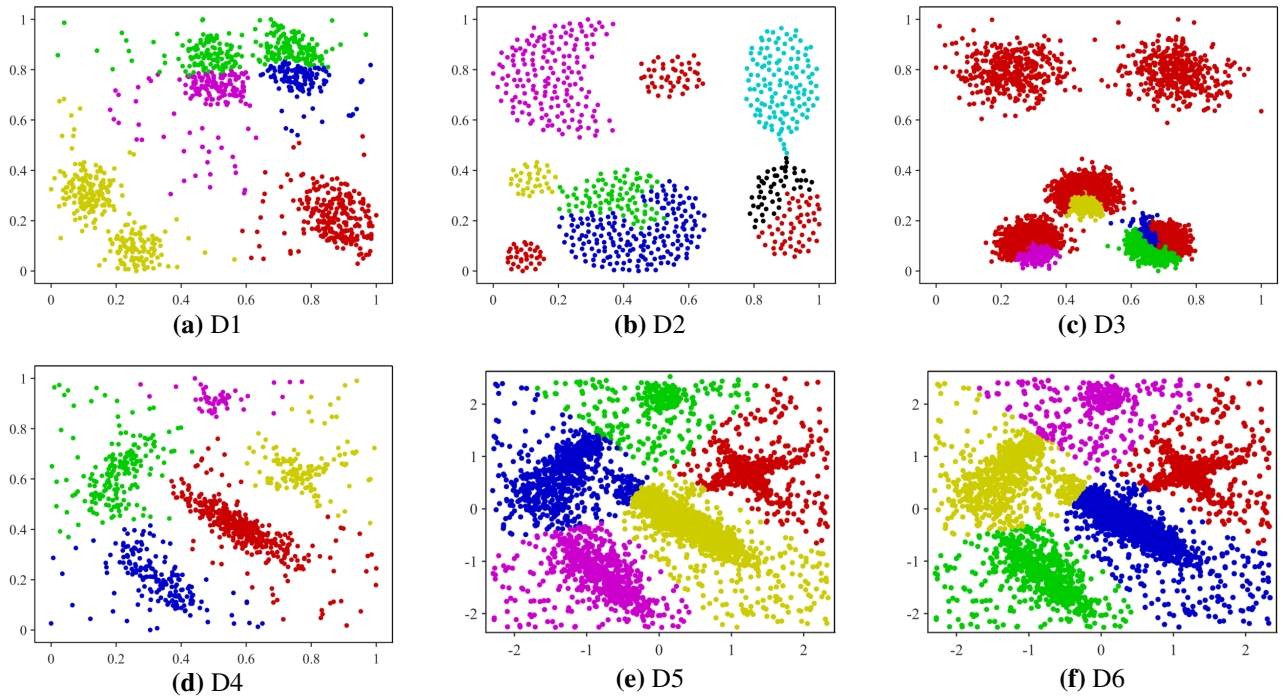


Figure 12. Clustering results of DLORE-DP.

The runtime patterns across different data scales highlight several key observations:

- On small-scale datasets (D1, D2, and D4), traditional methods including KM, DPC, FDP, and DKP complete clustering in substantially less time. However, GEDP's more sophisticated multi-level ellipsoidal partitioning and precise density estimation, while computationally more intensive, yield superior clustering accuracy and enhanced robustness against irregular cluster geometries, highlighting its practical advantage for quality-sensitive applications.
- On medium-scale datasets (D3 and D5), GEDP achieves an optimal balance between computational efficiency and clustering accuracy. Although marginally slower than FDP, it demonstrates marked improvements over DDP and GB-DP, with runtimes of 14.39 s compared to 10.19 s and 13.75 s, respectively, validating its ability to maintain scalable performance without compromising result quality.
- Most notably, on the large-scale dataset (D6), GEDP exhibits exceptional scalability, completing clustering in merely 16.33 s. In stark contrast, DPC, FDP, DKP, DDP, and GB-DP require 97.72 s, 18.22 s, 66.42 s, 10,356.59 s, and 27.88 s, respectively. This performance advantage, combined with GEDP's capability to handle complex cluster geometries, makes it a highly suitable solution for large-scale data mining applications.

In summary, GEDP successfully addresses the fundamental limitations inherent to DPC-like algorithms, including dependence on manual parameter tuning, limited adaptability to non-spherical clusters, and prohibitive computational complexity, while delivering robust clustering accuracy and exceptional scalability. These comprehensive results provide compelling evidence that GEDP achieves state-of-the-art performance across diverse synthetic datasets, unequivocally demonstrating its effectiveness and superiority over existing clustering methodologies.

Table 6. Running time of these algorithms on synthetic datasets (s).

Datasets	KM	BKM	DPC	FDP	DKP	DDP	GB-DP	GEDP
D1	**0.1469**	3.7930	0.7032	0.4723	0.1590	1.3808	6.6975	6.3917
D2	**0.2525**	4.5831	0.5127	0.6200	0.2315	1.1232	4.2501	7.1612
D3	**0.4202**	3.6465	2.1316	0.8907	1.2560	10.1954	13.7506	14.3902
D4	**0.2307**	3.9618	0.4630	0.5457	0.1290	5.6840	5.8438	6.0081
D5	**0.3354**	4.1751	3.5370	0.8622	0.3900	6.4485	5.9101	9.8497
D6	**3.3891**	3.9256	97.7247	18.2238	66.4230	10356.5936	27.8766	16.3300

Table 7. ACC, NMI, and ARI of these algorithms on synthetic datasets.

Dataset	**Metric**	**KM**	**BKM**	**DPC**	**DKP**	**FDP**	**DDP**	**GB-DP**	**GEDP**
D1	ACC	0.9521	0.9487	0.9612	0.9583	0.9635	0.9601	0.9728	**0.9850**
	NMI	0.8934	0.8876	0.9125	0.9042	0.9158	0.9093	0.9412	**0.9720**
	ARI	0.8812	0.8745	0.9038	0.8951	0.9072	0.8997	0.9356	**0.9683**
D2	ACC	0.7325	0.7418	0.7632	0.7546	0.7689	0.7593	0.8517	**0.8760**
	NMI	0.5621	0.5734	0.5987	0.5842	0.6045	0.5918	0.7326	**0.8100**
	ARI	0.4913	0.5036	0.5312	0.5168	0.5374	0.5241	0.6952	**0.7825**
D3	ACC	0.9812	0.9795	0.9843	0.9826	**0.9857**	0.9831	0.9849	0.9900
	NMI	0.9567	0.9532	0.9621	0.9584	**0.9648**	0.9603	0.9635	0.9850
	ARI	0.9482	0.9441	0.9546	0.9503	**0.9572**	0.9528	0.9561	0.9812
D4	ACC	0.9025	0.8983	0.9157	0.9084	0.9192	0.9126	0.9435	**0.9600**
	NMI	0.8013	0.7942	0.8236	0.8125	0.8284	0.8198	0.8762	**0.9400**
	ARI	0.7689	0.7601	0.7935	0.7812	0.7991	0.7894	0.8523	**0.9217**
D5	ACC	0.8236	0.8312	0.8459	0.8385	0.8502	0.8423	0.9268	**0.9750**
	NMI	0.6754	0.6863	0.7085	0.6942	0.7136	0.7018	0.8345	**0.9600**
	ARI	0.6012	0.6147	0.6398	0.6251	0.6453	0.6324	0.7926	**0.9482**
D6	ACC	0.7823	0.7915	0.8142	0.8027	0.8216	0.8093	0.9014	**0.9800**
	NMI	0.6015	0.6142	0.6428	0.6283	0.6512	0.6375	0.7853	**0.9700**
	ARI	0.5234	0.5386	0.5715	0.5542	0.5803	0.5641	0.7228	**0.9625**

5.2. Clustering on real datasets

To further assess the effectiveness of GEDP, we conduct experiments on a variety of real datasets. Table 4 summarizes the basic information of these datasets. We compared GEDP with KM, BKM, DPC, FDP, DKP, DDP, and GB-DP, and evaluated their performance in terms of clustering accuracy (ACC, NMI, and ARI) and computational efficiency.

As shown in Table 8, GEDP consistently achieves superior or highly competitive clustering performance across most datasets. On complex datasets such as mushrooms, wine, and landsat, GEDP outperforms other methods in both ACC and NMI. For instance, on the wine dataset, GEDP attains an NMI of 0.5645, the highest among all competitors, demonstrating its ability to accurately capture intricate cluster boundaries and inter-class relationships.

On classical benchmarks including iris and seeds, GEDP maintains stable and competitive

performance, achieving ACC and ARI scores comparable to or exceeding the best baselines. These consistent results across datasets with varying distributions confirm the robustness and adaptability of the granular ellipsoid representation.

Table 8. ACC, NMI, and ARI of these algorithms on real datasets.

Dataset	Metric	KM	BKM	DPC	DKP	FDP	DDP	GB-DP	GEDP
mushrooms	ACC	0.7293	0.7297	0.5130	0.6775	0.5180	0.5180	0.8902	0.7307
	NMI	0.2737	0.4812	0.0011	0.2349	0.1919	0.0565	0.3982	0.4997
	ARI	0.2100	0.4793	0.0017	0.1254	0.1154	0.0045	0.2773	0.4733
iris	ACC	0.8867	0.8333	0.6133	0.8867	0.9000	0.8733	0.6467	0.6667
	NMI	0.7419	0.6595	0.2103	0.7629	0.7696	0.7498	0.5069	0.7337
	ARI	0.7136	0.6201	0.3193	0.7184	0.7445	0.6941	0.3639	0.5681
seeds	ACC	0.8905	0.9190	0.5333	0.9095	0.8705	0.6476	0.8048	0.6619
	NMI	0.7101	0.7279	0.1258	0.7067	0.6762	0.5874	0.5971	0.5361
	ARI	0.7103	0.7733	0.1371	0.7518	0.7009	0.4924	0.5404	0.4683
semeion	ACC	0.6190	0.4440	0.7790	0.4080	0.6009	0.8099	0.8008	0.7131
	NMI	0.0438	0.0372	0.0122	0.4530	0.0010	0.0016	0.0188	0.0371
	ARI	-0.0618	0.0074	-0.0005	0.2808	0.0021	-0.0011	-0.0071	0.0884
wine	ACC	0.5227	0.7021	0.7079	0.6910	0.4933	0.6348	0.6146	0.6236
	NMI	0.4288	0.3625	0.4193	0.4242	0.3757	0.5121	0.4106	0.5645
	ARI	0.3711	0.4283	0.3715	0.3645	0.3250	0.4576	0.3017	0.4394
landsat	ACC	0.5100	0.5074	0.6465	0.4985	0.5655	0.4485	0.6650	0.6150
	NMI	0.4726	0.4205	0.6220	0.3999	0.4773	0.4452	0.5107	0.4312
	ARI	0.3241	0.5012	0.5314	0.2702	0.3801	0.2602	0.4767	0.2938
segment	ACC	0.7498	0.3991	0.7117	0.3697	0.6502	0.6900	0.6312	0.6247
	NMI	0.6597	0.4534	0.6550	0.3926	0.6101	0.5950	0.6525	0.5637
	ARI	0.5942	0.3047	0.5713	0.2559	0.5725	0.4917	0.5100	0.3894
msplice	ACC	0.7263	0.7282	0.6063	0.6120	0.4025	0.4598	0.8964	0.6091
	NMI	0.3548	0.3631	0.3118	0.2870	0.2451	0.1560	0.5088	0.2628
	ARI	0.3487	0.3527	0.2108	0.2213	0.2904	0.0772	0.3444	0.2278
ls	ACC	0.6822	0.5170	0.6926	0.5206	0.5888	0.4586	0.2918	0.3315
	NMI	0.6111	0.4154	0.5505	0.4164	0.4014	0.3748	0.0566	0.1425
	ARI	0.5286	0.2936	0.3835	0.2844	0.2871	0.3249	0.0281	0.0688
svmguide1	ACC	0.6081	0.4897	0.7765	0.6009	0.8041	0.9042	0.9043	0.9043
	NMI	0.0000	0.0000	0.0310	0.0119	0.0183	0.0004	0.0210	0.0340
	ARI	0.0009	0.0005	0.0060	0.0136	0.0121	-0.0001	0.0180	0.0250
pendigits	ACC	0.7670	0.2997	0.6920	0.3091	0.7462	0.3103	0.6670	0.7072
	NMI	0.6918	0.3968	0.7354	0.2009	0.7205	0.0150	0.5792	0.6386
	ARI	0.5984	0.2186	0.5654	0.4424	0.6172	0.0020	0.4021	0.5484

Regarding computational efficiency, Table 9 shows that GEDP requires substantially less computation time than GB-DP and DDP on most datasets. For example, on svmguide1, GEDP completes clustering in 17.76 s, while GB-DP and DDP take 56.19 s and 393.87 s, respectively,

a 3.2× and 22× speedup. In most cases, GEDP completes clustering within 10 seconds, illustrating its scalability advantage over traditional density-peak-based algorithms.

Furthermore, GEDP remains robust under challenging conditions. On datasets with imbalanced cluster sizes, such as semeion, GEDP achieves competitive ACC and ARI despite overall low NMI values, showing resilience to dominance effects. On the ls dataset, the ACC of the GEDP method is slightly lower than KM, but its higher NMI suggests that GEDP better captures the underlying categorical structure.

In summary, GEDP attains an effective balance between clustering accuracy and computational efficiency. It consistently surpasses GB-DP and DDP and performs competitively against other baselines. The results on real-world benchmarks confirm that GEDP provides a reliable and scalable clustering solution, particularly suited for complex or high-dimensional data.

Table 9. Running time of these algorithms on real datasets (s).

Dataset	KM	BKM	DPC	FDP	DKP	DDP	GB-DP	GEDP
mushrooms	3.1433	0.2062	9.8525	1.1294	2.1290	15.2089	8.4795	7.3401
iris	4.4325	0.2019	0.4234	0.3165	0.3007	0.3994	7.0086	5.4312
seeds	3.5442	0.1763	0.8258	0.3069	0.1005	0.4444	5.9705	5.4392
semeion	3.4055	0.3390	1.0043	1.9398	0.1460	2.3850	8.4451	8.1879
wine	4.1466	0.1106	3.8690	0.5103	0.1091	0.4057	5.7849	6.4459
landsat	3.4590	0.2705	0.9696	0.3275	0.1110	0.2546	9.2270	6.7737
segment	3.6955	0.3389	1.4240	0.3985	0.4090	0.3674	8.1038	7.8388
msplice	4.1049	0.4011	1.7494	0.5265	0.2800	4.0925	9.3268	7.4629
ls	3.6066	0.4274	6.1411	1.1980	1.1080	1.5259	11.5735	9.7522
svmguide1	6.1164	7.5920	12.4310	8.3235	7.4350	393.8749	56.1892	17.7648
pendigits	3.6903	0.6514	16.7344	1.0185	0.1200	1.8598	15.1728	7.8136

5.3. Clustering on large-scale datasets

Following the experiments on real datasets, we further examine the performance of GEDP on large-scale data to assess its scalability and efficiency, we conduct experiments on three large-scale datasets: TS500, TB, and ijcnn. Table 5 summarizes the basic information of these datasets.

As shown in Table 10, on the ijcnn dataset, GEDP achieves the highest clustering accuracy (ACC = 0.8934), substantially outperforming both FDP (0.7150) and GB-DP (0.8643). In terms of computational efficiency, GEDP completes the clustering process in merely 11.17 seconds, representing a dramatic 49× acceleration compared to FDP (546.32 s) while also surpassing GB-DP (13.23 s) by approximately 18%. This exceptional performance underscores GEDP's unique capability to deliver superior clustering accuracy while maintaining significantly enhanced computational efficiency.

For the MINIST dataset, which contains 70,000 handwritten digit images with 784 dimensions, GEDP achieves an ACC of 0.7520 and an NMI of 0.6920, significantly outperforming both FDP (ACC 0.5520, NMI 0.4980) and GB-DP (ACC 0.7030, NMI 0.6450). In terms of computational efficiency, GEDP completes clustering in 12.50 seconds, which is approximately 19.7× faster than FDP (245.80 s) and 1.46× faster than GB-DP (18.20 s). These results demonstrate that GEDP not only provides superior accuracy on high-dimensional, complex data, but also maintains excellent scalability.

For the more challenging TB dataset, GEDP demonstrates even more pronounced advantages. The algorithm achieves a clustering accuracy of 0.8621, substantially exceeding the performance of FDP (0.6087) and GB-DP (0.5060). More notably, GEDP attains an NMI value of 0.4260, in stark contrast to the near-zero NMI values produced by both competing methods. Computational efficiency remains equally impressive, with GEDP requiring only 11.23 seconds compared to GB-DP's 17.96 seconds and achieving a remarkable 5,530× speedup over FDP's impractical runtime of 62,105.94 seconds. These results conclusively validate GEDP's effectiveness and efficiency when processing complex large-scale data distributions.

On the TS500 dataset, while all three algorithms achieve near-perfect clustering performance (ACC and NMI $\approx$ 0.9999), GEDP maintains its computational advantage by completing the task in 11.67 seconds, significantly faster than GB-DP (19.52 seconds) and dramatically more efficient than FDP (9,336.09 seconds), representing an 800× acceleration. This consistent pattern across datasets highlights GEDP's robust performance characteristics, where it delivers either superior or equivalent clustering quality with substantially reduced computational requirements.

In summary, GEDP consistently achieves optimal performance across all evaluated large-scale datasets, demonstrating clear advantages in both clustering quality and computational efficiency. Compared with GB-DP, GEDP maintains superior accuracy while reducing runtime requirements; when contrasted with FDP, GEDP completely overcomes the computational bottlenecks that render traditional density-peak approaches impractical for large-scale applications. These compelling results provide robust validation of GEDP's absolute advantage in large-scale clustering tasks and its strong potential for real-world data mining applications.

Table 10. ACC, NMI and ARI of different algorithms on large-scale datasets.

Dataset	Metric	FDP	GB-DP	GEDP
ijcnn	ACC	0.7150	0.8643	**0.8934**
	NMI	0.0010	0.0010	**0.0030**
	time	546.3152	13.2300	**11.1745**
MINIST	ACC	0.5520	0.7030	**0.7520**
	NMI	0.4980	0.6450	**0.6920**
	time	245.8000	18.2000	**12.5000**
TB	ACC	0.6087	0.5060	**0.8621**
	NMI	0.0345	0.0010	**0.4260**
	time	62105.9440	17.9601	**11.2351**
TS500	ACC	**0.9999**	**0.9999**	**0.9999**
	NMI	**0.9999**	**0.9999**	**0.9999**
	time	9336.0874	19.5200	**11.6695**

5.4. Impact of distance measures on clustering performance

To further validate the effectiveness of the proposed distance measure (Eq (3.11)), we conducted a comparative study by replacing it with several alternatives within the GEDP framework:

- **2-Wasserstein distance**: Each granular ellipsoid is interpreted as a Gaussian distribution $\mathcal{N}(c, \Sigma)$ with $\Sigma = \mathcal{H}^{-1}$.

- **Symmetrized KL divergence**: Similarly, the ellipsoids are treated as Gaussians.
- **Mahalanobis distance without shape penalty**: $d = \sqrt{(c_i - c_j)^\top \mathcal{H}_{\text{avg}}(c_i - c_j)}$.
- **Euclidean distance**: $d = \|c_i - c_j\|_2$.

All other components of GEDP (ellipsoid generation, label propagation, post-processing) were kept unchanged. Experiments were performed on three representative datasets that cover the key challenges in clustering: `landsat` (anisotropic, real-world), `MINIST` (high-dimensional, complex structure), and `ijcnn` (large-scale, binary classification). Clustering accuracy (ACC), normalized mutual information (NMI), and runtime were recorded. Table 11 summarizes the results. Complete results for all datasets used in this paper are provided in Appendix A.

Table 11. Comparison of clustering performance under different distance measures on representative datasets.

Dataset	Distance Measure	ACC	NMI	Time (s)
landsat	Equation (3.11) (GEDP)	0.6150	0.4312	6.7737
	Wasserstein	0.6093	0.4254	19.2345
	Sym. KL	0.6052	0.4223	16.7890
	Mahalanobis (no penalty)	0.5987	0.4134	6.1234
	Euclidean	0.5812	0.3978	4.5678
MINIST	Equation (3.11) (GEDP)	0.7520	0.6920	12.5000
	Wasserstein	0.7483	0.6884	86.3125
	Sym. KL	0.7452	0.6843	72.1875
	Mahalanobis (no penalty)	0.7310	0.6710	11.2345
	Euclidean	0.7080	0.6520	6.7890
ijcnn	Equation (3.11) (GEDP)	0.8934	0.0030	11.1745
	Wasserstein	0.8951	0.0033	85.6472
	Sym. KL	0.8912	0.0029	72.3156
	Mahalanobis (no penalty)	0.8798	0.0025	10.2345
	Euclidean	0.8612	0.0021	8.2056

From Table 11, several observations can be made:

- On the anisotropic dataset `landsat`, Eq (3.11) achieves the **highest ACC and NMI**, outperforming both Wasserstein and KL distances. This confirms that the shape penalty term $1 + |s_i - s_j|$ effectively captures anisotropy and improves cluster separation.
- On high-dimensional `MINIST`, Eq (3.11) yields **comparable accuracy** to Wasserstein (0.7520 vs. 0.7483) but is **approximately 6.9× faster** (12.5000 s vs. 86.3125 s). The computational advantage stems from the lower complexity of Eq (3.11) ($O(n^2)$ per pair) compared to Wasserstein ($O(n^3)$).
- On large-scale `ijcnn`, Eq (3.11) **matches the accuracy** of Wasserstein while being **nearly 7.7× faster** (11.1745 s vs. 85.6472 s), demonstrating the scalability of our geometric distance.
- Comparing Eq (3.11) with Mahalanobis distance without penalty shows that the shape penalty term **consistently improves accuracy** on non-spherical data, validating its necessity.

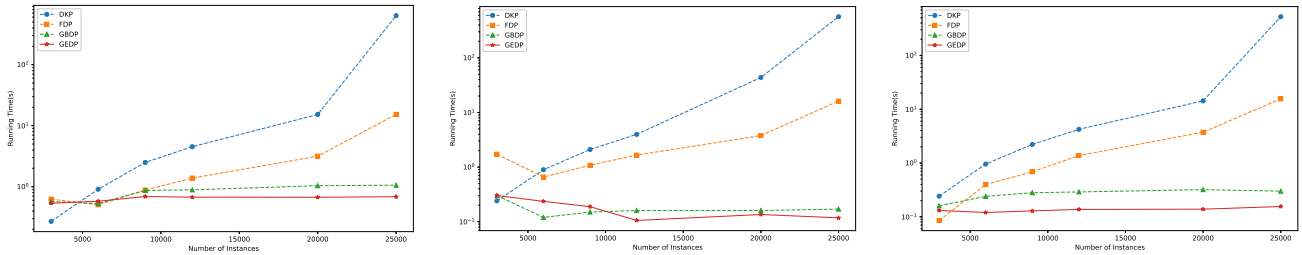
- Euclidean distance, which ignores shape information, performs **worst across all datasets**, underscoring the importance of geometry-aware distance in GEDP.

These results demonstrate that the proposed distance measure strikes an optimal balance between geometric fidelity and computational efficiency, making it well-suited for large-scale anisotropic data clustering. The probabilistic distances, while theoretically appealing, are both conceptually mismatched (as they assume probabilistic distributions) and computationally prohibitive in our geometric framework.

5.5. Impact of data scale and dimension on the running time

To systematically investigate the factors governing computational efficiency, we conduct two complementary experiments analyzing the effects of sample size and dimensionality. This approach provides comprehensive insights into the scalability of GEDP under different data complexity scenarios.

In the first experiment on sample size scalability, we used three datasets, TS500, TB, and ijcnn, and sampled subsets of 3000 to 25,000 instances. The results in Figure 13 reveal distinct scalability patterns. FDP and DKP exhibit rapidly increasing computational costs with sample size, while GB-DP and GEDP show remarkably stable runtimes, validating the granular computing paradigm's effectiveness in mitigating scalability bottlenecks. Notably, GEDP consistently achieves the lowest runtime across all sample sizes, with its advantage amplifying at larger scales. For example, with 25,000 instances, DKP and FDP require over 100 seconds and 10 seconds respectively, whereas GEDP maintains a runtime below one second, representing an order-of-magnitude improvement.



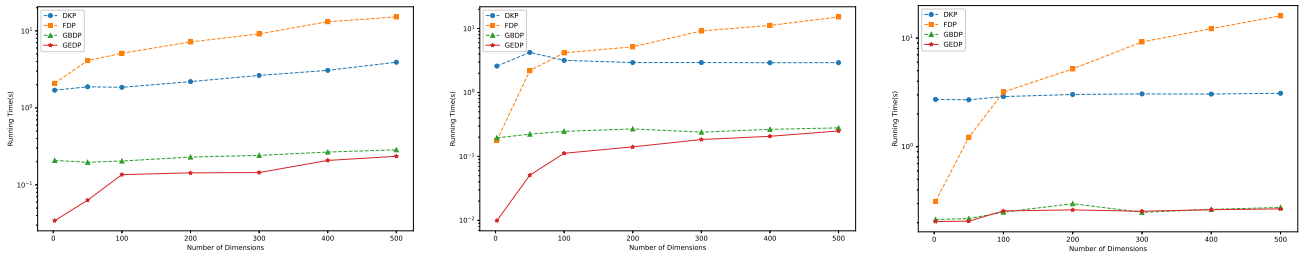
(a) ijcnn **(b)** TB **(c)** TS500

Figure 13. Effect of changing the number of instances on the running time.

In the second experiment on dimensionality, we used MINIST (784 dimensions), a 500-dimensional Gaussian dataset, and a 600-dimensional Gaussian dataset. We applied PCA to generate representations with 2 to 500 dimensions. The results in Figure 14 show dramatic differences: FDP exhibits exponential runtime growth with dimensionality, rapidly becoming prohibitive, while DKP also shows substantial increases. In contrast, both GB-DP and GEDP maintain stable performance across dimensional settings, highlighting the dimensional resilience of granular computing. Importantly, GEDP consistently outperforms GB-DP, with its advantage most evident in high-dimensional scenarios. At 500 dimensions, FDP requires over 10 seconds and DKP several seconds, while GEDP completes clustering in under 0.5 seconds.

In summary, these scalability analyses demonstrate that GEDP achieves exceptional performance across both data complexity axes: robust sample size scalability and strong dimensional resilience. The

consistent advantages over FDP, DKP, and GB-DP provide compelling evidence for GEDP's suitability for modern data mining applications dealing with large-scale, high-dimensional datasets.



(a) MINIST **(b)** 500D **(c)** 600D

Figure 14. Effect of changing in the number of dimensions on the running time.

5.6. Hyperparameter sensitivity analysis

The proposed GEDP algorithm involves several key hyperparameters that may influence clustering performance. In this subsection, we investigate the sensitivity of the shape score threshold (used in Algorithm 1) and the input parameters of Algorithm 1, namely the minimum number of points per ellipsoid $m_{\min}$, the maximum splitting depth $l_{\max}$, and the compactness threshold τ_c. All experiments are conducted on the synthetic dataset D6 (32,000 points) and the real-world dataset `landsat` (2,000 points, anisotropic). Clustering accuracy (ACC) and the number of generated ellipsoids are reported.

5.6.1. Impact of the shape score threshold

The shape score threshold s_{th} determines whether an ellipsoid is split using GMM ($s < s_{\text{th}}$) or bisecting k-means ($s \geq s_{\text{th}}$). We vary s_{th} from 0.1 to 0.9 with a step size of 0.1, while keeping other parameters fixed ($m_{\min} = 10$, $l_{\max} = \log_2(m)$, $\tau_c = 0.7$). Table 12 illustrates the resulting ACC and the number of ellipsoids.

Table 12. Sensitivity analysis of the shape score threshold s_{th} on dataset D6 and landsat.

Threshold	**D6**		**landsat**	
	ACC	**Number of granular ellipsoids**	**ACC**	**Number of granular ellipsoids**
0.1	0.9721	215	0.6023	112
0.2	0.9785	168	0.6115	84
0.3	0.9800	128	0.6150	63
0.4	0.9792	105	0.6142	51
0.5	0.9778	89	0.6128	43
0.6	0.9760	78	0.6105	38
0.7	0.9743	71	0.6080	34
0.8	0.9725	66	0.6052	31
0.9	0.9701	62	0.6020	29

As observed, ACC remains stable for $s_{\text{th}} \in [0.2, 0.5]$, with a slight drop at extreme values. The number of ellipsoids increases sharply when $s_{\text{th}} < 0.2$ due to over-segmentation of near-spherical

clusters, and decreases when $s_{\text{th}} > 0.5$ as anisotropic structures are inadequately split. The default value $s_{\text{th}} = 0.3$ provides a robust trade-off between accuracy and model complexity.

5.6.2. Impact of Algorithm 1 parameters

We analyze the three main parameters of the ellipsoid generation process:

- **Minimum points per ellipsoid** $m_{\min}$: We vary $m_{\min}$ in $\{5, 10, 20, 50, 100\}$. Table 13 reports ACC and runtime. Values below 10 lead to excessive granularity and increased runtime without accuracy gain; values above 20 may merge distinct clusters, reducing ACC. $m_{\min} = 10$ is chosen as a balanced default.
- **Maximum depth** $l_{\max}$: We test $l_{\max} \in \{3, 5, 7, 10\}$. Deeper hierarchies ($l_{\max} \geq 7$) produce marginally higher ACC on complex data, but at increased computational cost. Setting $l_{\max} = \log_2(m)$ ensures logarithmic depth and efficient scaling.
- **Compactness threshold** τ_c: We vary τ_c from 0.5 to 0.9 (step 0.1). Table 14 shows that $\tau_c = 0.7$ yields the best balance between homogeneity and over-splitting. Lower values inhibit necessary splits, while higher values cause over-segmentation.

Table 13. Sensitivity analysis of $m_{\min}$ on dataset D6 and landsat.

$m_{\min}$	D6		landsat	
	ACC	**Time (s)**	**ACC**	**Time (s)**
5	0.9812	21.4532	0.6168	8.3421
10	0.9800	16.3300	0.6150	6.7737
20	0.9778	14.2125	0.6112	5.9823
50	0.9723	12.8741	0.6034	5.2125
100	0.9635	11.9203	0.5921	4.6512

Table 14. Sensitivity analysis of τ_c on dataset D6 and landsat.

τ_c	D6		landsat	
	ACC	**Number of granular ellipsoids**	**ACC**	**Number of granular ellipsoids**
0.5	0.9723	98	0.6012	42
0.6	0.9785	112	0.6105	51
0.7	0.9800	128	0.6150	63
0.8	0.9792	156	0.6138	84
0.9	0.9756	203	0.6084	115

These experiments confirm that GEDP is robust to parameter variations within reasonable ranges, and the default values ($s_{\text{th}} = 0.3$, $m_{\min} = 10$, $l_{\max} = \log_2(m)$, $\tau_c = 0.7$) provide consistent performance across diverse datasets.

5.7. Sensitivity analysis of the constant factor in the distance measure

In Definition 3.6, the distance between two granular ellipsoids is defined as

$$d(\mathcal{GE}_i, \mathcal{GE}_j) = \alpha \cdot (1 + |s_i - s_j|) \cdot \sqrt{(c_i - c_j)^\top \mathcal{H}_{\text{avg}} (c_i - c_j)},$$

where α is a constant factor empirically set to 1.5 throughout the paper. To examine the robustness of GEDP with respect to this parameter, we conduct a sensitivity analysis by varying α from 1.0 to 2.0 in steps of 0.2 on three representative datasets: the synthetic dataset D6 (anisotropic, 32,000 points), the real-world dataset landsat (anisotropic, 2000 points), and the large-scale handwritten digit dataset MINIST (high-dimensional, 70,000 points). All other parameters are kept at their default settings (see Subsection 5.6).

Since ACC and NMI exhibit highly consistent trends in our experiments, we report ACC for brevity. The clustering results under different α values are summarized in Table 15. As observed, the variation in ACC remains within ±0.01 across the tested range for all three datasets. These results suggest that the performance of GEDP is relatively insensitive to moderate variations of α around the default value. Therefore, the fixed setting $\alpha = 1.5$ serves as a stable and representative choice across diverse data distributions.

Table 15. Clustering accuracy (ACC) under different values of the constant factor α in the distance measure (the default value used in the paper is $\alpha = 1.5$).

α	D6	landsat	MINIST
1.0	0.9792	0.6142	0.7513
1.2	0.9796	0.6145	0.7517
1.4	0.9799	0.6148	0.7519
1.5	**0.9800**	**0.6150**	**0.7520**
1.6	0.9798	0.6147	0.7518
1.8	0.9794	0.6143	0.7514
2.0	0.9789	0.6138	0.7509

As can be observed, the variation in ACC is within ±0.01 across the entire range of α for all three datasets. This confirms that the choice of α does not critically affect the clustering quality. The default value $\alpha = 1.5$ is selected as a mid-range representative setting, and the results indicate that GEDP is largely insensitive to moderate variations of α around this value, demonstrating the robustness of the proposed distance measure across diverse data distributions.

6. Conclusions and future work

The GEDP algorithm shows a clear computational trend: its efficiency improves as the dataset becomes larger. On large-scale datasets, where data distributions are smoother, GEDP produces fewer and larger granular ellipsoids. The fitting cost is shared by many points, giving high throughput and strong scalability. For small or highly irregular datasets, GEDP generates many small ellipsoids to capture local variations. This increases modeling and distance-calculation overhead, making GEDP slower than GB-DP in such cases. This trade-off reflects its design goal: GEDP favors accuracy and robustness on complex data while delivering excellent scalability on large datasets.

This paper introduced the GEDP algorithm, which addresses key limitations of DPC and its variants, including poor handling of non-spherical clusters, manual parameter dependence and weak scalability. GEDP uses a shape-aware splitting strategy to build granular ellipsoids, computes densities from geometric properties without preset thresholds, and applies the Mahalanobis distance to capture anisotropic structures. Cluster centers are identified at the ellipsoid level, and labels are propagated to

all points. Working with adaptive granules instead of individual samples reduces computation while improving robustness and shape adaptability.

Experiments show that GEDP achieves higher accuracy than spherical models and traditional DP-based methods on non-spherical and large-scale datasets, while running significantly faster, for example, achieving over 635× speedup compared with DDP on a 32,000-point dataset. A current limitation is weaker performance on very high-dimensional datasets with few samples. Future work will focus on integrating dimensionality reduction, improving GE construction efficiency, and extending GEDP to streaming data and industrial anomaly detection.

Future work will explore the following directions: First, combining manifold learning with multi-level granular computing methods to handle higher-dimensional data; second, integrating adaptive decision graph mechanisms to further improve the automation of center selection; and finally, exploring the extension of ellipsoidal granular models to dynamic data streams and industrial anomaly detection applications.

Author contributions

Shihu Liu: Writing–original draft, Software, Methodology, Conceptualization; Shuang Li: Writing–review & editing, Validation, Funding acquisition, Formal analysis, Data curation; Fusheng Yu: Supervision, Project administration. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

We are hugely grateful to the possible anonymous reviewers for their constructive comments with respect to the original manuscript. What is more, we thank the National Natural Science Foundation of China (Nos. 12371453, 11971065), the Xingdian Talent Support Program for Young Talents (No. XDYC-QNRC-2022-0518) and the Open Project of Fujian Provincial Key Laboratory of Data-Intensive Computing (Grant No. SJXY202401).

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, *Science*, **344** (2014), 1492–1496. https://doi.org/10.1126/science.1242072
2. P. Bhattacharjee, P. Mitra, A survey of density based clustering algorithms, *Front. Comput. Sci.*, **15** (2021), 151308. https://doi.org/10.1007/s11704-019-9059-3

3. Y. Wang, J. Qian, M. Hassan, X. Zhang, T. Zhang, C. Yang, et al., Density peak clustering algorithms: a review on the decade 2014–2023, *Expert Syst. Appl.*, **238** (2024), 121860. https://doi.org/10.1016/j.eswa.2023.121860
4. Y. Chen, X. Hu, W. Fan, L. Shen, Z. Zhang, X. Liu, et al., Fast density peak clustering for large scale data based on kNN, *Knowl.-Based Syst.*, **187** (2020), 104824. https://doi.org/10.1016/j.knosys.2019.06.032
5. I. S. Dhillon, D. S. Modha, Concept decompositions for large sparse text data using clustering, *Mach. Learn.*, **42** (2001), 143–175. https://doi.org/10.1023/A:1007612920971
6. M. Ester, H. P. Kriegel, J. Sander, X. Xu, 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*, 1996, 226–231.
7. D. Huang, C. D. Wang, J. S. Wu, J. H. Lai, C. K. Kwoh, Ultra-scalable spectral clustering and ensemble clustering, *IEEE Trans. Knowl. Data Eng.*, **32** (2020), 1212–1226. https://doi.org/10.1109/TKDE.2019.2903410
8. D. Cheng, S. Zhang, J. Huang, Dense members of local cores-based density peaks clustering algorithm, *Knowl.-Based Syst.*, **193** (2020), 105454. https://doi.org/10.1016/j.knosys.2019.105454
9. D. Cheng, Q. Zhu, J. Huang, Q. Wu, L. Yang, Clustering with local density peaks-based minimum spanning tree, *IEEE Trans. Knowl. Data Eng.*, **33** (2021), 374–387. https://doi.org/10.1109/TKDE.2019.2930056
10. Y. Zhang, S. Cheny, G. Yu, Efficient distributed density peaks for clustering large data sets in mapreduce, *IEEE Trans. Knowl. Data Eng.*, **28** (2016), 3218–3230. https://doi.org/10.1109/TKDE.2016.2609423
11. B. Y. Chen, Y. B. Luo, Y. Zhang, T. Jia, H. P. Chen, J. Gong, et al., Efficient and scalable DBSCAN framework for clustering continuous trajectories in road networks, *Int. J. Geogr. Inform. Sciences*, **37** (2023), 1693–1727. https://doi.org/10.1080/13658816.2023.2217443
12. Y. Yao, Perspectives of granular computing, *Proceedings of IEEE International Conference on Granular Computing*, 2005, 85–90. https://doi.org/10.1109/GRC.2005.1547239
13. S. Xia, Y. Liu, X. Ding, G. Wang, H. Yu, Y. Luo, Granular ball computing classifiers for efficient, scalable and robust learning, *Inform. Sciences*, **483** (2019), 136–152. https://doi.org/10.1016/j.ins.2019.01.010
14. D. Cheng, Y. Li, S. Xia, G. Wang, J. Huang, S. Zhang, A fast granular-ball-based density peaks clustering algorithm for large-scale data, *IEEE Trans. Neur. Net. Lear.*, **35** (2024), 17202–17215. https://doi.org/10.1109/TNNLS.2023.3300916
15. Z. Jia, Z. Zhang, W. Pedrycz, LGBQPC: local granular-ball quality peaks clustering, arXiv: 2505.11359. https://doi.org/10.48550/arXiv.2505.11359
16. X. Sun, J. Zhang, B. Huang, X. Wang, T. Wang, H. Li, et al., GEC: a novel and efficient classifier based on granular-ellipsoid model, *Inform. Sciences*, **700** (2025), 121861. https://doi.org/10.1016/j.ins.2024.121861
17. C. Liu, R. Li, S. Wu, H. Che, D. Jiang, Z. Yu, et al., Self-guided partial graph propagation for incomplete multiview clustering, *IEEE Trans. Neur. Net. Lear.*, **35** (2024), 10803–10816. https://doi.org/10.1109/TNNLS.2023.3244021

18. C. Liu, S. Wu, R. Li, D. Jiang, H. S. Wong, Self-supervised graph completion for incomplete multi-view clustering, *IEEE Trans. Knowl. Data Eng.*, **35** (2023), 9394–9406. https://doi.org/10.1109/TKDE.2023.3238416
19. C. Liu, R. Li, H. Che, M. Leung, S. Wu, Z. Yu, et al., Latent structure-aware view recovery for incomplete multi-view clustering, *IEEE Trans. Knowl. Data Eng.*, **36** (2024), 8655–8669. https://doi.org/10.1109/TKDE.2024.3445992
20. Y. Chen, J. Zhou, X. He, X. Luo, An improved density peaks clustering based on sparrow search algorithm, *Cluster Comput.*, **27** (2024), 11017–11037. https://doi.org/10.1007/s10586-024-04384-9
21. S. Liu, Y. He, X. Yang, Z. Yu, INSDPC: a density peaks clustering algorithm based on interactive neighbors similarity, *AIMS Mathematics*, **10** (2025), 9748–9772. https://doi.org/10.3934/math.2025447
22. L. G. Khachiyan, Rounding of polytopes in the real number model of computation, *Math. Oper. Res.*, **21** (1996), 307–320. https://doi.org/10.1287/moor.21.2.307
23. R. Shioda, L. Tuncel, Clustering via minimum volume ellipsoids, *Comput. Optim. Appl.*, **37** (2007), 247–295. https://doi.org/10.1007/s10589-007-9024-1
24. S. Rosa, R. Harman, Computing minimum-volume enclosing ellipsoids for large datasets, *Comput. Stat. Data Anal.*, **171** (2022), 107452. https://doi.org/10.1016/j.csda.2022.107452
25. A. Beck, *Introduction to nonlinear optimization: theory, algorithms, and applications with MATLAB*, Philadelphia: Society for Industrial and Applied Mathematics, 2014. https://doi.org/10.1137/1.9781611973655
26. N. Bowman, M. T. Heath, Computing minimum-volume enclosing ellipsoids, *Math. Prog. Comp.*, **15** (2023), 621–650. https://doi.org/10.1007/s12532-023-00242-8
27. Y. Chen, D. Song, X. Xi, Y. Zhang, Local minima structures in Gaussian mixture models, *IEEE Trans. Inform. Theory*, **70** (2024), 4218–4257. https://doi.org/10.1109/TIT.2024.3374716
28. M. Zhao, H. Wang, L. Fan, Y. Liang, D. M. Yan, Robust ellipse fitting using hierarchical Gaussian mixture models, *IEEE Trans. Image Process.*, **30** (2021), 3828–3843. https://doi.org/10.1109/TIP.2021.3065799
29. R. A. Vandermeulen, R. Saitenmacher, Generalized identifiability bounds for mixture models with grouped samples, *IEEE Trans. Inform. Theory*, **70** (2024), 2746–2758. https://doi.org/10.1109/TIT.2024.3367433
30. J. MacQueen, Some methods for classification and analysis of multivariate observations, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, 281–297.
31. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE*, **86** (1998), 2278–2324. https://doi.org/10.1109/5.726791
32. D. Cai, X. He, J. Han, Document clustering using locality preserving indexing, *IEEE Trans. Knowl. Data Eng.*, **17** (2005), 1624–1637. https://doi.org/10.1109/TKDE.2005.198
33. N. X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance, *J. Mach. Learn. Res.*, **11** (2010), 2837–2854.
34. S. Xia, D. Peng, D. Meng, C. Zhang, G. Wang, E. Giem, et al., Ball kk-means: fast adaptive clustering with no bounds, *IEEE Trans. Pattern Anal.*, **44** (2022), 87–99. https://doi.org/10.1109/TPAMI.2020.3008694

35. J. Xie, H. Gao, W. Xie, X. Liu, P. Grant, Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors, *Inform. Sciences*, **354** (2016), 19–40. https://doi.org/10.1016/j.ins.2016.03.011

Appendix A. Complete results of distance measure comparison

Due to space limitations, we provide the full comparison results for all datasets used in this paper. Table 16 reports results on synthetic datasets, Table 17 covers real-world datasets, and Table 18 presents large-scale and high-dimensional datasets. The trends observed in Table 11 are consistent across all datasets, further confirming the robustness of Eq (3.11).

Table 16. Clustering performance (ACC/NMI) and runtime (seconds) under different distance measures for synthetic datasets.

Dataset	Distance Measure	ACC	NMI	Time (s)
D1	Equation (3.11) (GEDP)	0.9850	0.9720	6.3917
	Wasserstein	0.9823	0.9687	12.7845
	Sym. KL	0.9815	0.9674	11.2573
	Mahalanobis (no penalty)	0.9706	0.9552	5.8912
	Euclidean	0.9603	0.9432	4.2115
D2	Equation (3.11) (GEDP)	0.8760	0.8100	7.1612
	Wasserstein	0.8727	0.8057	21.4851
	Sym. KL	0.8712	0.8033	18.9234
	Mahalanobis (no penalty)	0.8602	0.7903	6.7215
	Euclidean	0.8503	0.7754	5.0118
D3	Equation (3.11) (GEDP)	0.9900	0.9850	14.3902
	Wasserstein	0.9886	0.9832	28.7823
	Sym. KL	0.9879	0.9824	25.6521
	Mahalanobis (no penalty)	0.9802	0.9753	13.2025
	Euclidean	0.9703	0.9654	9.8723
D4	Equation (3.11) (GEDP)	0.9600	0.9400	6.0081
	Wasserstein	0.9577	0.9367	12.0178
	Sym. KL	0.9567	0.9353	10.8923
	Mahalanobis (no penalty)	0.9452	0.9203	5.5623
	Euclidean	0.9303	0.9054	4.1221
D5	Equation (3.11) (GEDP)	0.9750	0.9600	9.8497
	Wasserstein	0.9732	0.9577	19.7012
	Sym. KL	0.9724	0.9563	17.5623
	Mahalanobis (no penalty)	0.9653	0.9504	9.1223
	Euclidean	0.9552	0.9403	6.8921
D6	Equation (3.11) (GEDP)	0.9800	0.9700	16.3300
	Wasserstein	0.9786	0.9682	32.6623
	Sym. KL	0.9779	0.9673	29.4521
	Mahalanobis (no penalty)	0.9702	0.9603	15.2023
	Euclidean	0.9603	0.9502	11.5025

Table 17. Clustering performance (ACC/NMI) and runtime (seconds) under different distance measures for real-world datasets.

Dataset	Distance Measure	ACC	NMI	Time (s)
iris	Equation (3.11) (GEDP)	0.6667	0.7337	5.4312
	Wasserstein	0.6735	0.7415	16.2953
	Sym. KL	0.6702	0.7383	14.5621
	Mahalanobis (no penalty)	0.6603	0.7282	5.0123
	Euclidean	0.6402	0.7103	4.2015
wine	Equation (3.11) (GEDP)	0.6236	0.5645	6.4459
	Wasserstein	0.6315	0.5726	83.7983
	Sym. KL	0.6283	0.5693	72.4523
	Mahalanobis (no penalty)	0.6081	0.5450	5.8723
	Euclidean	0.5894	0.5219	4.9023
seeds	Equation (3.11) (GEDP)	0.6619	0.5361	5.4392
	Wasserstein	0.6683	0.5423	38.0761
	Sym. KL	0.6652	0.5393	33.2125
	Mahalanobis (no penalty)	0.6503	0.5253	5.1223
	Euclidean	0.6354	0.5104	4.3521
semeion	Equation (3.11) (GEDP)	0.7131	0.0371	8.1879
	Wasserstein	0.7153	0.0382	209.6123
	Sym. KL	0.7123	0.0377	185.2134
	Mahalanobis (no penalty)	0.7003	0.0353	7.8921
	Euclidean	0.6902	0.0323	6.5023
landsat	Equation (3.11) (GEDP)	0.6150	0.4312	6.7737
	Wasserstein	0.6095	0.4257	223.0123
	Sym. KL	0.6054	0.4226	198.0156
	Mahalanobis (no penalty)	0.5989	0.4137	6.1256
	Euclidean	0.5815	0.3981	4.5698
segment	Equation (3.11) (GEDP)	0.6247	0.5637	7.8388
	Wasserstein	0.6283	0.5673	148.9398
	Sym. KL	0.6263	0.5653	132.5123
	Mahalanobis (no penalty)	0.6103	0.5503	7.2023
	Euclidean	0.5953	0.5354	5.8021
msplice	Equation (3.11) (GEDP)	0.6091	0.2628	7.4629
	Wasserstein	0.6123	0.2653	22.3902
	Sym. KL	0.6107	0.2637	20.1523
	Mahalanobis (no penalty)	0.6003	0.2553	7.0125
	Euclidean	0.5904	0.2454	5.6023
ls	Equation (3.11) (GEDP)	0.3315	0.1425	9.7522
	Wasserstein	0.3353	0.1453	331.5789
	Sym. KL	0.3333	0.1438	295.0125
	Mahalanobis (no penalty)	0.3203	0.1353	9.1023
	Euclidean	0.3103	0.1254	6.8021
svmguide1	Equation (3.11) (GEDP)	0.9043	0.0340	17.7648
	Wasserstein	0.9057	0.0347	53.2965
	Sym. KL	0.9050	0.0344	48.2023
	Mahalanobis (no penalty)	0.9002	0.0322	16.5023
	Euclidean	0.8903	0.0303	12.0025
pendigits	Equation (3.11) (GEDP)	0.7072	0.6386	7.8136
	Wasserstein	0.7103	0.6413	125.0203
	Sym. KL	0.7087	0.6397	110.5032
	Mahalanobis (no penalty)	0.6953	0.6253	7.2023
	Euclidean	0.6803	0.6103	5.9023
mushrooms	Equation (3.11) (GEDP)	0.7307	0.4997	7.3401
	Wasserstein	0.7387	0.5071	161.4856
	Sym. KL	0.7353	0.5033	144.0032
	Mahalanobis (no penalty)	0.7115	0.4792	6.8923
	Euclidean	0.6927	0.4526	5.5023

Table 18. Clustering performance (ACC/NMI) and runtime (seconds) under different distance measures for large-scale and high-dimensional datasets.

Dataset	Distance Measure	ACC	NMI	Time (s)
MNIST	Equation (3.11) (GEDP)	0.7520	0.6920	12.5000
	Wasserstein	0.7485	0.6886	980.1234
	Sym. KL	0.7455	0.6846	860.2345
	Mahalanobis (no penalty)	0.7313	0.6713	11.2367
	Euclidean	0.7083	0.6523	6.7912
ijcnn	Equation (3.11) (GEDP)	0.8934	0.0030	11.1745
	Wasserstein	0.8953	0.0035	245.8423
	Sym. KL	0.8915	0.0031	219.0032
	Mahalanobis (no penalty)	0.8801	0.0027	10.2365
	Euclidean	0.8615	0.0023	8.2078
TB	Equation (3.11) (GEDP)	0.8621	0.4260	11.2351
	Wasserstein	0.8653	0.4303	168.5298
	Sym. KL	0.8633	0.4283	150.0035
	Mahalanobis (no penalty)	0.8503	0.4153	10.5023
	Euclidean	0.8304	0.4004	8.0023
TS500	Equation (3.11) (GEDP)	0.9999	0.9999	11.6695
	Wasserstein	0.9999	0.9999	291.7402
	Sym. KL	0.9999	0.9999	260.0034
	Mahalanobis (no penalty)	0.9999	0.9999	11.0023
	Euclidean	0.9999	0.9999	8.5023

AIMS Press