

# Peak-Padding: Clustering by Padding Density Peaks With the Minimum Padding Cost

Junyi Guan, *Member, IEEE*, Bingbing Jiang<sup>✉</sup>, *Member, IEEE*, Weiguo Sheng<sup>✉</sup>, *Member, IEEE*, Yangyang Zhao<sup>✉</sup>, Sheng Li<sup>✉</sup>, *Senior Member, IEEE*, and Xiongxiang He<sup>✉</sup>

**Abstract**—Clustering complex-shaped clusters is still challenging for most existing clustering algorithms. Herein, the peak-padding clustering algorithm (PeakPad)—clustering by padding density peaks with the minimum padding cost—is proposed. PeakPad executes clustering on the density surface and views complex-shaped clusters as combinations of highly associated single-peak clusters. The minimum padding cost that fully considers the surrounding context of a density peak is proposed to reflect a density peak’s center potential, enabling PeakPad to have robust center detection performance. Unlike mean-shift (MSC), which detects centers based on their attributes in a complex-shaped density surface embedded in the high-dimensional space of density and features, PeakPad detects centers in a standard-shaped surface embedded in the 2-D density-change (DC) density space (composed of density and DC feature). Such standardization allows PeakPad to have fast and robust cluster center detection performance on complex-shaped clusters based on the minimum padding cost. Besides, PeakPad can provide a reasonable evaluation of the association between single-peak clusters by using the minimum padding cost. As a result, PeakPad can fast capture complex-shaped clusters, achieve robust center detection performance, and be suitable for large datasets. Benchmark test results on both synthetic and real datasets demonstrate the effectiveness of PeakPad.

**Index Terms**—Center detection, clustering, complex-shaped clusters, density peak.

## I. INTRODUCTION

CLUSTERING is one of the most fundamental topics in data mining [1]. Its attempt to group similar objects has solidified its irreplaceable role in pattern recognition [2], computer vision [3], image processing [4], biomedical data analysis [22], and so on.

As no consensus on the definition of a cluster has been reached, clustering algorithms based on specific assumptions regarding the nature of a “cluster” were proposed [5]. As a popular nonparametric clustering technique, mean-shift (MSC) [6] treats data points as an empirical probability density function and views local density maximum areas (density peaks) of

the function as centers. It can divide nonspherical clusters by iteratively shifting (assigning) each point to its proximal dense region. However, MSC may overdivide a multipeak cluster (also known as a multicenter cluster) [7], since it views each density peak as a center.

In 2014, *Science* published the clustering method called density peak clustering (DPC) [8]. DPC addresses the overdivision problem of MSC by finding density peaks with high “center potential” as cluster centers. Based on DPC’s assumption, a point’s center potential ( $\gamma = \rho \cdot \delta$ ) depends on two factors:  $\rho$  (local density) and  $\delta$  (the minimum distance to a higher density point). DPC uses a decision graph (i.e., a  $\rho$ - $\delta$  plot) to select centers. Once the centers are chosen, each remaining point is assigned to the cluster of its nearest higher density point.

Although DPC’s center detection is effective and easy to implement, its Euclidean-distance-based  $\delta$  struggles with complex-shaped clusters [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. Cheng et al. [19] used granular balls (GBs), defined their  $\rho$  and  $\delta$ , and clustered them via DPC, significantly reducing runtime. Guan et al. [18] introduced “main” and “satellite” peaks based on density and designed a  $\delta$ -based attenuator to suppress satellite peaks, yielding a more robust decision graph. Many works have also explored alternative distance or dissimilarity measures. For example, Liu et al. [9] used a shared-nearest-neighbor-based  $\delta$  to capture centers in sparse clusters; Guan et al. [10] proposed a density-deviation-cost  $\delta$  to quickly identify main peaks in complex-shaped clusters; Sharma et al. [16] designed a mixture-distance-based  $\delta$  for mixed data, using a divergence-based nonlinear S-distance [20] to better detect complex boundaries [21]; Pizzagalli et al. [11] kept DPC’s  $\delta$  but adopted a shortest-path strategy for more robust shape reconstruction. Despite their improvements, these methods still follow DPC’s unrobust  $\gamma$ -based center detection idea.

DPC’s clustering process can be interpreted as a peak-padding strategy, referred to as “the DPC-padding strategy”: *each point is padded (assigned) to a higher density area based on its padding cost  $\gamma$ , and peaks with relatively large  $\gamma$  values are selected as centers*, as illustrated in Fig. 1 (upper). For example, peak  $p_0$  has a much higher padding cost  $\gamma$  (shown as the red rectangle) than point  $x_0$ , making it a more suitable center candidate. However,  $\gamma$  covers and extends well beyond the valley region—i.e., the area representing the minimum padding cost—and therefore fails to precisely reflect the valley structure, which is key to identifying cluster boundaries [6]. To address this, we propose the min-padding strategy: *points are padded using their minimum padding cost  $\theta$ , and peaks with relatively large  $\theta$  values are selected as centers*, as shown

Received 4 October 2024; revised 3 March 2025, 30 May 2025, and 11 July 2025; accepted 2 September 2025. This work was supported by the National Natural Science Foundation of China under Grant 62306282, Grant 62233016, and Grant 62576128. (Corresponding author: Weiguo Sheng.)

Junyi Guan, Bingbing Jiang, and Weiguo Sheng are with the School of Information Science and Technology, Hangzhou Normal University, Hangzhou 311121, China (e-mail: jonnyguan73@163.com; jiangbb@hznu.edu.cn; w.sheng@ieee.org).

Yangyang Zhao, Sheng Li, and Xiongxiang He are with the College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: 211122030059@zjut.edu.cn; shengli@zjut.edu.cn; hxx@zjut.edu.cn).

Digital Object Identifier 10.1109/TNNLS.2025.3606527

2162-237X © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

Authorized licensed use limited to: Hangzhou Normal University. Downloaded on November 10, 2025 at 09:23:25 UTC from IEEE Xplore. Restrictions apply.

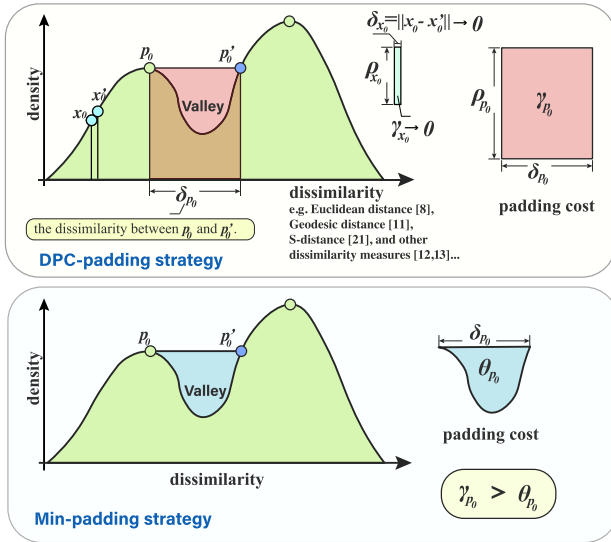


Fig. 1. DPC-padding strategy and the proposed min-padding strategy on a continuous density curve. Notably, nonpeak point  $x_0$  on a continuous curve has no padding cost, since its nearest higher density point  $x'_0$  is infinitely close to it, i.e.,  $\delta_{x_0} = \|x_0 - x'_0\| \rightarrow 0$ . Then,  $\gamma_{x_0} = \rho_{x_0} \cdot \delta_{x_0} \rightarrow 0$ .

in Fig. 1 (bottom). This approach captures the valley structure more accurately.

Unlike the global  $\gamma$  metric, which often stretches beyond the actual density valleys (see Fig. 1, top),  $\theta$  closely follows local density minima, turning hidden valley structures into clear, measurable features. This allows for direct detection of saddle points—something the  $\gamma$  metric cannot do. As shown in Fig. 1 (bottom), our method removes the overextension issue of DPC’s  $\gamma$  at cluster boundaries, producing cluster borders that align well with the true density valleys and providing a solid basis for analyzing complex data distributions.

Fig. 2 compares the performance of the DPC-padding strategy and the min-padding strategy on 1-D continuous density curves. As shown in Fig. 2(a), the DPC-padding strategy produces identical decision graphs for tasks **i** and **ii**. However, for ideal clustering—cutting at lower density valleys—centers should be  $p_1$  and  $p_3$  in task **i**, and  $p_2$  and  $p_3$  in task **ii**. This shows that the  $\gamma$  attribute does not reliably guide correct center selection. In contrast, the proposed min-padding strategy, by explicitly considering the valley structure [Fig. 2(b)], easily solves both tasks. This aligns better with human perception, as our nervous system tends to favor economical solutions [23], illustrated in Fig. 2(c).

The effectiveness of a padding strategy heavily depends on the choice of dissimilarity measure, and “Euclidean distance” is not suitable in this context. Peaks near wide valleys may have larger padding costs than those near deeper valleys when measured by Euclidean distance, despite deep valleys better representing cluster boundaries [10]. In Fig. 3, peak  $p_2$  has a lower minimum padding cost than  $p_1$ , causing the min-padding strategy to yield a suboptimal result by padding peak  $p_2$ . Moreover, accurately calculating the minimum padding cost in Euclidean space is difficult, requiring a specific discrete summation that is time-consuming and prone to unavoidable computational errors.

To address the limitations mentioned above, we propose a new dissimilarity measure for the min-padding strategy called the “density change distance (DC distance).” Building

on this, the Peak-Padding Clustering algorithm (PeakPad)<sup>1</sup> is introduced. The main contributions of PeakPad are as follows.

- 1) The DC-distance-based minimum padding cost effectively reflects the center potential of peaks and can be computed efficiently, enabling PeakPad to achieve robust and high-performance center detection.
- 2) PeakPad treats each density peak as representing a single-peak cluster. By appropriately padding noncenter peaks into final multipeak clusters, it can naturally capture complex-shaped multipeak clusters.
- 3) PeakPad uses kNN distances as input, making it well-suited for clustering large-scale datasets.

The rest of the article is organized as follows. Section II introduces related works; Section III focuses on the proposed PeakPad algorithm; Section IV presents benchmark tests and discussions; and Section V concludes the article.

## II. RELATED WORKS

MSC [6] identifies cluster centers as dense regions in the feature space. It models data points as an empirical probability density function using a specific kernel function and locates the local maxima of this function, which correspond to dense regions. Each point iteratively performs an “MSC” procedure—a gradient ascent on the local density surface—until convergence. The shifted points then form groups, while the stationary points (local maxima) are identified as cluster centers, thus completing the clustering process. Although MSC effectively captures clusters of arbitrary shapes without requiring the number of clusters to be specified in advance, its tendency to treat every local density maximum as a center can lead to overdivision in multipeak clusters [7].

Although DPC [8] builds on the core idea of MSC, it differs by not treating all density peaks as centers. Instead, DPC selects centers based on its center assumption—that cluster centers are density peaks with both high density and relatively large distance from points of higher density. For clustering, DPC assigns each noncenter point to the same cluster as its nearest neighbor with higher density, a process known as *DPC’s allocation strategy*. Typically, DPC is viewed as a hierarchical clustering method [25] employing a specific linkage metric called the minimum center-boosting distance [25], which aligns with the proposed *Peak-Padding strategy*.

Nevertheless, DPC’s allocation strategy often fails to accurately reconstruct complex shapes, as it may wrongly associate unrelated points [11]. Furthermore, as mentioned in the Introduction, DPC’s center detection based on the  $\gamma$  metric (referred to as  $\gamma$ -based center detection) is unreliable for multipeak clusters. Although many methods [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] have been proposed to improve center detection, they still depend on DPC’s original  $\gamma$ -based approach and lack robustness for multipeak clusters. In contrast, PeakPad offers robust center detection for complex-shaped clusters by adopting a new center assumption: cluster centers are density peaks with relatively large minimum padding costs.

DBSCAN [26], a classic density-based clustering method, considers a cluster as a set of density-connected points. The “density-connected” is defined using  $\epsilon$  and *minPts* parameters. Although DBSCAN can reconstruct complex-shaped clusters and eliminate noise, its parameter tuning is often tedious. Later works, e.g., [27], developed the adaptive parameter tuning, still, they would possibly merge high-overlapping clusters

<sup>1</sup>The code is available at <https://github.com/Guanjunyi/PeakPadding>.

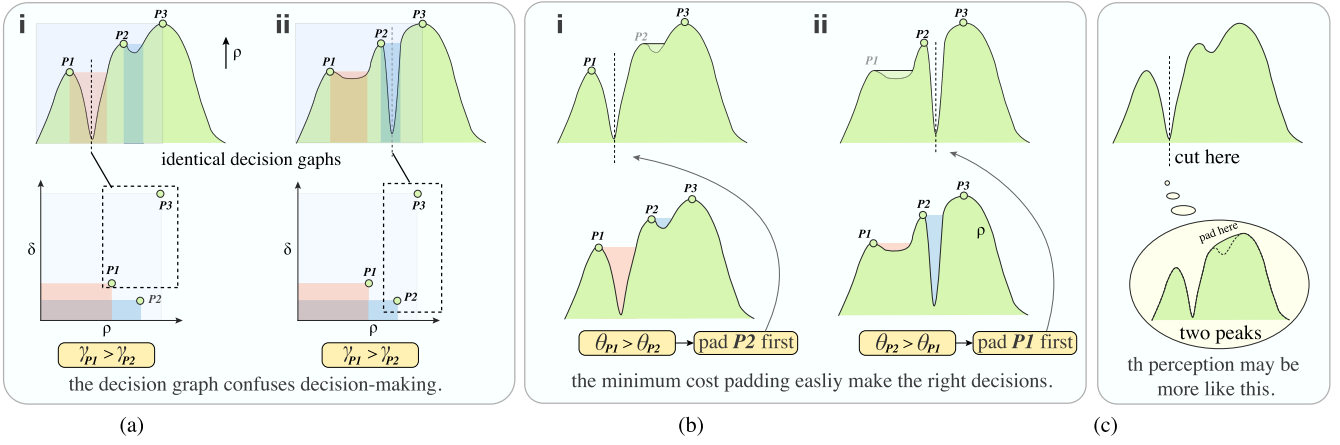


Fig. 2. (a) DPC-padding strategy's center detection, (b) min-padding strategy's center detection, and (c) human perception of peak-padding.

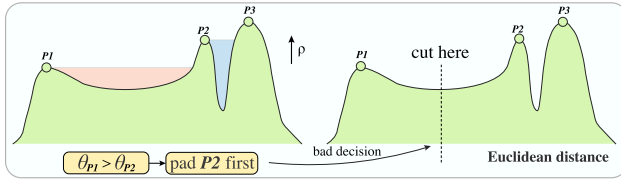


Fig. 3. Limitation of the min-padding strategy on Euclidean space.

[28]. In PeakPad, the centers of high-overlapping clusters can be detected to avoid merging. Inspired by DBSCAN, points within a cluster are ensured to be density-connected and the concept of “DC distance” is introduced to quantify the density-connectivity between points to achieve the robust reconstruction of complex-shaped clusters.

Guan et al. [13] argued that being a “density peak” is a necessary but not sufficient condition for identifying cluster centers. To address this, they introduced the “peak graph,” a sparse graph structure where nodes represent density peaks. This approach not only aids in selecting appropriate centers but also significantly reduces computational complexity without sacrificing accuracy. Cheng et al. [29] use natural density peaks (NDPs) combined with neighbor-based graph distance (GD) to perform clustering via  $K$ -means [47], allowing detection of arbitrarily shaped clusters. Each NDP represents a single-peak cluster, containing all associated data points and having the highest density. In PeakPad, we first detect single-peak clusters represented by local density peaks in the dataset. Unlike the NDP- $K$ -means method [29], we utilize the peak graph structure and design a min-padding strategy on it to obtain the final multipeak clusters.

### III. PEAK-PADDING CLUSTERING (PEAKPAD)

This section provides a detailed introduction to the proposed PeakPad algorithm. Fig. 4 illustrates its clustering process, which consists of two main steps: 1) building the peak graph and 2) performing peak-padding clustering.

*Step 1:* Given the dataset  $X$  and parameter  $k$ , PeakPad first estimates the density of data points [using (1)], then detects density peaks, single-peak clusters, border points, and saddle points (according to Definitions 1 to 4). Using the adjacency relationships among single-peak clusters, PeakPad constructs the peak graph.

*Step~2:* Based on the peak graph, PeakPad iteratively pads peaks to adjacent higher density peaks by minimizing the padding cost  $\theta$  [as defined in (3)] while simultaneously

generating a dendrogram. Next, it selects the top  $n_c$  peaks with the largest  $\theta$  values as cluster centers via a clear decision graph, following our center Assumption 1. The dendrogram is then cut into  $n_c$  peak groups, and finally, clustering is completed by grouping all points within the same peak group into clusters.

#### A. Peak Graph Building

1) *Detection of Peaks:* Given a dataset  $X = \{x_1, x_2, \dots, x_n \mid x_i \in \mathbb{R}^d\}$ , a kNN-based density function  $\rho : \mathbb{R}^d \rightarrow \mathbb{R}^+$  is applied to estimate point  $x_i$ 's density  $\rho_{x_i}$ , as follows:

$$\rho_{x_i} = \sum_{x_j \in N_k(x_i)} e^{-d_{ij}^2} \quad (1)$$

where  $N_k(x_i)$  indicates the set of  $x_i$ 's  $k$  nearest neighbors.

On this basis, a density peak is viewed as the density maximum point within its  $k$  nearest neighbors [24], as follows.

*Definition 1:* If  $\rho_{x_i} > \max_{x_j \in N_k(x_i)}(\rho_{x_j})$ , then  $x_i$  is a density peak  $p$ .

According to Definition 1, the dataset  $X$  is divided into peaks  $P$  and nonpeak points  $\bar{P}$ , i.e.,  $X = P \cup \bar{P}$ , where  $P = \{p_1, p_2, \dots, p_{n_p}\}$  indicates all  $n_p$  peaks of dataset  $X$ .

Based on the clustering idea of MSC [6], each nonpeak point  $x_i \in \bar{P}$  is associated with its nearest neighbor  $x_j$  within a higher density area, i.e.,  $x_j = \arg \min_x d_{xx_i}$ , s.t.  $x \in N_k(x_i)$ ,  $\rho_x > \rho_{x_i}$ . As a result, each nonpeak point will be associated with a peak. Then, a single-peak cluster is defined as follows.

*Definition 2:* A single-peak cluster, denoted as  $S(p)$ , is composed of one peak  $p$  and all nonpeak points associated with the peak.

Therefore, peak  $p$  is considered as the representative point (center) of  $S(p)$ .

2) *Graph Building:* Peaks  $P$  is built into a peak graph  $G(P, E)$ , where  $E = \{e_{p_i p_j} \mid \exists B_{p_i p_j}\}$ . If border point(s)  $B_{p_i p_j}$  exist(s) between the single-peak clusters  $S(p_i)$  and  $S(p_j)$ , then an edge  $e_{p_i p_j}$  connects  $p_i$  and  $p_j$  in the peak graph  $G$ .

Border points are considered to only exist on both sides of the borderline between intersecting single-peak clusters. That is to say, the borderline divides the mutual-proximal border points of different single-peak clusters. Therefore, to effectively detect the mutual-proximity between border points, a small-value  $k_b$  is introduced

$$k_b = \min(k, \lfloor 2 \ln(n) \rfloor) \quad (2)$$

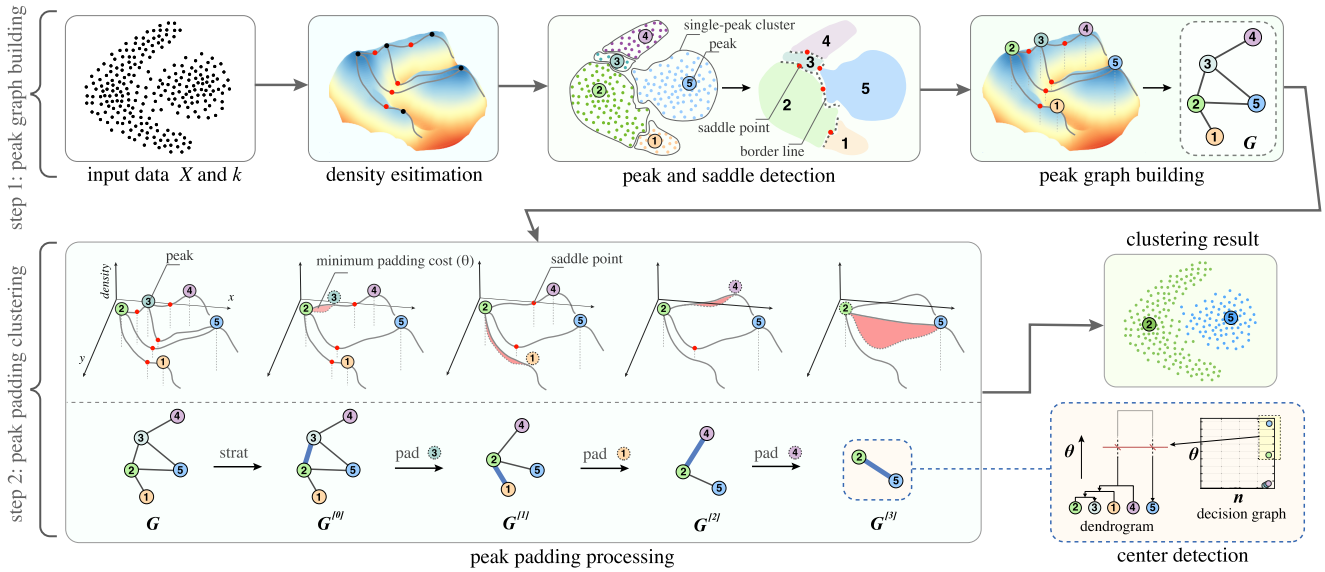


Fig. 4. Clustering idea of the proposed PeakPad algorithm.

where parameter  $k$  and “2” are used to prevent  $k_b$  from becoming too small to detect border points, and  $\lfloor \cdot \rfloor$  is a floor function.

Then, border points  $B_{p_i p_j}$  are defined as follows:

**Definition 3:** If mutual-proximity points  $x_a$  and  $x_b$  are in different single-peak clusters, i.e.,  $x_a \in S(p_i) \cap N_{k_b}(x_b)$ ,  $x_b \in S(p_j) \cap N_{k_b}(x_a)$ , then, the low density point between  $x_a$  and  $x_b$  is a border point  $b$  between peaks  $p_i$  and  $p_j$ , i.e.,  $b = \arg \min_{x \in [x_a, x_b]} (\rho_x)$ ,  $b \in B_{p_i p_j}$ . This ensures that the border point has a lower density than either of the associated density peaks.

Besides, a saddle point is defined as follows.

**Definition 4:** If border point  $b$  has the highest density in its border set  $B$ , then point  $b$  is a saddle point, denoted as  $b^*$ , i.e.,  $b^* = \arg \max_{b \in B} (\rho_b)$ .

According to Definition 3, border points can be readily identified. If two single-peak clusters share border points, their corresponding peaks are connected in the peak graph. As shown in Fig. 4 (Step 1: peak graph building), peak areas that share border points are linked in the peak graph.

In what follows, a detailed description of the min-padding strategy on peak graph  $G$  will be given.

### B. Min-Padding Strategy

The min-padding strategy is a bottom-up hierarchical clustering technique applied to peak graphs [13].

- 1) Initially, we have a peak graph  $G$  containing  $n_p$  peaks. Consider  $G^{[t]} = G(P^{[t]}, E^{[t]})$  as the peak graph after  $t$  padding operations, where  $t \in \{0, 1, \dots, n_p - 1\}$ , and  $P^{[t]} = \{p_1^{[t]}, p_2^{[t]}, \dots, p_{n_p}^{[t]}\}$  is the set of peaks at time  $t$ .
- 2) In each padding operation, we select the peak  $p_i^{[t]}$  with the global minimum padding cost and pad it into its adjacent higher density peak  $p_j^{[t]}$ . The original peak  $p_i^{[t]}$  is then removed from  $P^{[t]}$ , updating the peak set to  $P^{[t+1]} = P^{[t]} \setminus \{p_i^{[t]}\}$ .
- 3) Simultaneously,  $p_j^{[t+1]}$  inherits all existing connections in  $E^{[t]}$  from both peaks  $p_i^{[t]}$  and  $p_j^{[t]}$  to other peaks, updating the edge set to  $E^{[t+1]}$ . The peak graph is then updated as  $G^{[t+1]} = G(P^{[t+1]}, E^{[t+1]})$ .

- 4) Update  $t \leftarrow t + 1$  and repeat the process until  $|P^{[t]}| = 1$  (i.e., the peak graph reduces to a single peak), completing the padding task.

During the padding process, peak  $p_i$  can obtain the minimum padding cost, as follows:

$$\begin{aligned} \theta_{p_i} &= \min_{V_{p_i^{[t]} p_j^{[t]}}} \text{Cost}(V_{p_i^{[t]} p_j^{[t]}}) \\ \text{s.t. } &\exists e_{p_i^{[t]} p_j^{[t]}} \in E^{[t]}, \rho_{p_j^{[t]}} > \rho_{p_i^{[t]}} \\ &t \in \{0, 1, \dots, n_p - 1\} \end{aligned} \quad (3)$$

where  $V_{p_i^{[t]} p_j^{[t]}}$  is a valley between adjacent peaks  $p_i$  and  $p_j$  within peak graph  $G^{[t]}$  (i.e.,  $\exists e_{p_i^{[t]} p_j^{[t]}} \in E^{[t]}$ ), and  $\text{Cost}(\cdot)$  is a padding cost function.

Equation (3) implies that a peak is only allowed to be padded to an adjacent higher density peak within its connected component. Therefore, if peak  $p_i$  is the highest density peak of a connected component in peak graph  $G$ , then, peak  $p_i^{[t]}$  ( $\forall t$ ) has no adjacent higher density peak to satisfy its padding operation, i.e.,  $\theta_{p_i} = \emptyset$ . Consequently, peak  $p_i$  must serve as a cluster center, referred to as a “center peak.”

For each center peak  $p_i$  with  $\theta_{p_i} = \emptyset$ , its padding cost is assigned as follows:

$$\theta_{p_i} = \beta \cdot \max_{\theta_p \neq \emptyset} (\theta_p) \quad (4)$$

where the scaling factor  $\beta > 1$  (default: 2) ensures that center peaks stand out clearly in the  $\theta$ -decision graph. Under the min-padding strategy, center peaks are iteratively padded into the global maximum-density peak.

### C. Center Assumption

Based on the min-padding strategy, our center assumption naturally emerges.

**Assumption 1:** Cluster centers are density peaks with a relatively large minimum padding cost  $\theta$  value.

In PeakPad, a cluster center can be regarded as the representative point of a cluster, as it corresponds to the highest density point within that cluster.

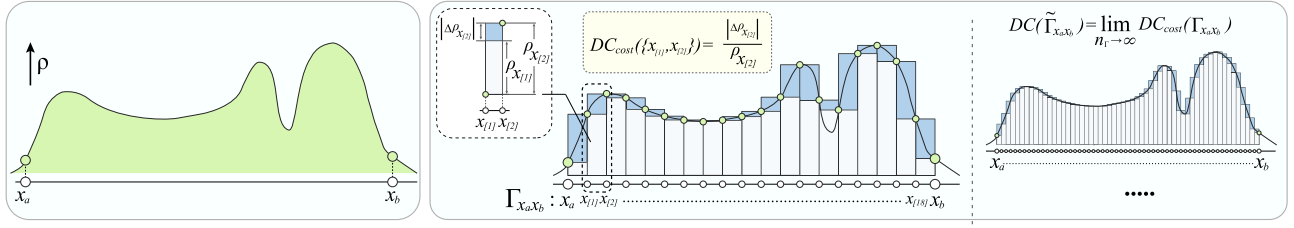


Fig. 5. Idea of DC distance based on DC cost.

Based on Assumption 1, a decision graph—similar to that used in DPC [8]—can be constructed to assist in the manual selection of cluster centers without requiring prior knowledge. Once the cluster centers are identified, PeakPad performs clustering by iteratively padding the remaining noncenter peaks to centers based on the minimum padding cost. Alternatively, the number of clusters can be preset, and the corresponding number of peaks with the highest padding costs can be selected as centers. Clustering is then completed by assigning each peak's cluster label to its corresponding single-peak cluster.

A detailed explanation of the “DC distance”—a dissimilarity measure used for estimating  $\theta$ —will be provided in Section III-D.

#### D. Density Change Distance

1) *DC distance of a Path*: Consider a compact path  $\Gamma_{x_i x_j} = \{x_{[0]}, x_{[1]}, \dots, x_{[n_\Gamma]}\}$  between points  $x_i$  and  $x_j$ , satisfying that adjacent points along the path are neighbors, where  $x_{[0]} = x_i$  and  $x_{[n_\Gamma]} = x_j$ . The *density change (DC) cost* of path  $\Gamma_{x_i x_j}$  is defined as follows.

**Definition 5:** For a path  $\Gamma_{x_i x_j}$  between points  $x_i$  and  $x_j$ , its DC cost is the sum of relative DCs of points along path  $\Gamma_{x_i x_j}$ , as follows.

$$DC_{\text{cost}}(\Gamma_{x_i x_j}) = \sum_{m=1}^{n_\Gamma} \frac{|\Delta\rho_{x_{[m]}}|}{\rho_{x_{[m]}}} \quad (5)$$

where  $\Delta\rho_{x_{[m]}} = \rho_{x_{[m]}} - \rho_{x_{[m-1]}}$  represents the *density change* from points  $x_{[m]}$  to  $x_{[m-1]}$ , and  $|\Delta\rho_{x_{[m]}}| = |\rho_{x_{[m]}} - \rho_{x_{[m-1]}}|$  denotes the *absolute DC*.

Let  $\lambda = \max\{|\Delta\rho_{x_{[1]}}|, |\Delta\rho_{x_{[2]}}|, \dots, |\Delta\rho_{x_{[n_\Gamma]}}|\}$ . Since dataset  $X$  can be assumed to be sampled from an underlying continuous and smooth probability density surface, then a continuous path  $\tilde{\Gamma}_{x_i x_j}$  between  $x_i$  and  $x_j$  on the surface can be obtained by letting  $\lambda \rightarrow 0$  (i.e.,  $n_\Gamma \rightarrow \infty$ ). Then, the DC distance is defined as follows.

**Definition 6:** For a continuous path  $\tilde{\Gamma}_{x_i x_j}$ , its DC cost is defined as its DC distance

$$DC(\tilde{\Gamma}_{x_i x_j}) = \lim_{n_\Gamma \rightarrow \infty} DC_{\text{cost}}(\Gamma_{x_i x_j}) = \lim_{n_\Gamma \rightarrow \infty} \sum_{m=1}^{n_\Gamma} \frac{|\Delta\rho_{x_{[m]}}|}{\rho_{x_{[m]}}} \quad (6)$$

Fig. 5 illustrates the idea of DC distance based on the DC cost. As shown, when  $n_\Gamma \rightarrow \infty$ , path  $\Gamma_{x_a x_b} \rightarrow \tilde{\Gamma}_{x_a x_b}$ . Meanwhile,  $DC_{\text{cost}}(\Gamma_{x_i x_j}) \rightarrow DC(\tilde{\Gamma}_{x_i x_j})$ .

Due to the existence of  $|\cdot|$  in (6),  $DC(\tilde{\Gamma}_{x_i x_j})$  cannot be expressed in definite integral form [30]. To solve this,  $\tilde{\Gamma}$  is considered as a combination of density-change monotone paths (denoted as  $\tilde{\Gamma}^\circ$ ) as follows:

$$\tilde{\Gamma}_{x_i x_j} = \bigcup_{z=0}^{n_z} \tilde{\Gamma}_{x_{(z)} x_{(z+1)}}^\circ \quad (7)$$

where  $x_{(0)} = x_i$ ,  $x_{(n_z+1)} = x_j$ , and  $x_{(1)}, x_{(2)}, \dots, x_{(n_z)} \in \tilde{\Gamma}_{x_i x_j}$  are all local density extremum points along path  $\tilde{\Gamma}_{x_i x_j}$ , i.e.,  $x_{(1)}, x_{(2)}, \dots, x_{(n_z)}$  on the path that are either local density minima or local density maxima.

Along each density-change monotone segment  $\tilde{\Gamma}_{x_{(z)} x_{(z+1)}}^\circ = \{x_{[0]}, x_{[1]}, \dots, x_{[n_{\Gamma(z)}]}\}$ , where  $x_{[0]} = x_{(z)}$  and  $x_{[n_{\Gamma(z)}]} = x_{(z+1)}$ , the DC  $\Delta\rho_{x_{[m]}}$  will maintain the same sign. Thus, the DC distance of  $\tilde{\Gamma}_{x_{(z)} x_{(z+1)}}^\circ$  can be calculated as follows:

$$\begin{aligned} DC(\tilde{\Gamma}_{x_{(z)} x_{(z+1)}}^\circ) &= \lim_{n_{\Gamma(z)} \rightarrow \infty} \sum_{m=1}^{n_{\Gamma(z)}} \frac{|\Delta\rho_{x_{[m]}}|}{\rho_{x_{[m]}}} \\ &= \left| \lim_{n_{\Gamma(z)} \rightarrow \infty} \sum_{m=1}^{n_{\Gamma(z)}} \frac{\Delta\rho_{x_{[m]}}}{\rho_{x_{[m]}}} \right| \\ &= \left| \int_{\rho_{x_{(z)}}}^{\rho_{x_{(z+1)}}} \frac{1}{\rho_x} d\rho_x \right| \\ &= \left| \ln \frac{\rho_{x_{(z+1)}}}{\rho_{x_{(z)}}} \right|. \end{aligned} \quad (8)$$

Therefore, (6) can be rewritten as follows:

$$DC(\tilde{\Gamma}_{x_i x_j}) = \sum_{z=0}^{n_z} DC(\tilde{\Gamma}_{x_{(z)} x_{(z+1)}}^\circ) = \sum_{z=0}^{n_z} \left| \ln \frac{\rho_{x_{(z+1)}}}{\rho_{x_{(z)}}} \right|. \quad (9)$$

Particularly, if  $\tilde{\Gamma}_{x_i x_j}$  is a density-change monotone path, i.e.,  $\tilde{\Gamma}_{x_i x_j} = \tilde{\Gamma}_{x_i x_j}^\circ$ , then the DC distance between  $x_i$  and  $x_j$  is a fixed value, as follows:

$$DC(\tilde{\Gamma}_{x_i x_j}^\circ) = \left| \ln \frac{\rho_{x_j}}{\rho_{x_i}} \right| = |\ln \rho_{x_i} - \ln \rho_{x_j}| \quad (10)$$

Equation (10) can work for a single-peak cluster because there must be a  $\tilde{\Gamma}^\circ$  path between any two points within a single-peak cluster, see Fig. 6(a).

2) *DC Distance Between Adjacent Peaks*: Consider density peaks  $p_i, p_j \in P$  are in two intersecting single-peak clusters. Therefore, a path  $\tilde{\Gamma}_{p_i p_j}$  can be split into two density-change monotone paths, as  $\tilde{\Gamma}_{p_i p_j} = \tilde{\Gamma}_{p_i b}^\circ \cup \tilde{\Gamma}_{b p_j}^\circ$ , where point  $b$  can be any point of  $B_{p_i p_j}$  between single-peak clusters, and  $\forall b \in B_{p_i p_j}, \rho_b < \min(\rho_{p_i}, \rho_{p_j})$ . Then, for each border point  $b$ , a path with DC distance  $DC(\tilde{\Gamma}_{p_i b}^\circ \cup \tilde{\Gamma}_{b p_j}^\circ)$  can be obtained as follows:

$$\begin{aligned} DC(\tilde{\Gamma}_{p_i b}^\circ \cup \tilde{\Gamma}_{b p_j}^\circ) &= DC(\tilde{\Gamma}_{p_i b}^\circ) + DC(\tilde{\Gamma}_{b p_j}^\circ) \\ &= \left| \ln \frac{\rho_{p_i}}{\rho_b} \right| + \left| \ln \frac{\rho_b}{\rho_{p_j}} \right| = \ln \frac{\rho_{p_i}}{\rho_b} + \ln \frac{\rho_{p_j}}{\rho_b} \\ &= \ln \rho_{p_i} + \ln \rho_{p_j} - 2 \ln \rho_b \end{aligned} \quad (11)$$

herein, the DC distance between two adjacent peaks is defined as follows.

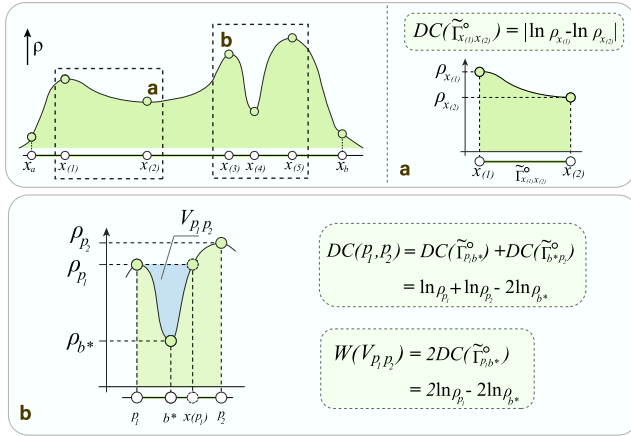


Fig. 6. (a) DC distance within a single-peak cluster, and (b) DC distance between peaks and the DC width of a valley.

**Definition 7:** For density peaks  $p_i$  and  $p_j$  within two intersecting single-peak clusters, their DC distance  $DC(p_i, p_j)$  is defined as the DC distance of the shortest path between them

$$DC(p_i, p_j) = \min_{b \in B_{p_i p_j}} DC(\tilde{\Gamma}_{p_i b} \cup \tilde{\Gamma}_{b p_j}) \\ = \min_{b \in B_{p_i p_j}} (\ln \rho_{p_i} + \ln \rho_{p_j} - 2 \ln \rho_b). \quad (12)$$

With saddle  $b^* = \arg \max_{b \in B_{p_i p_j}} (\rho_b)$ , we have

$$DC(p_i, p_j) = \ln \rho_{p_i} + \ln \rho_{p_j} - 2 \ln \rho_{b^*} \quad (13)$$

Let  $\rho_{p_i} < \rho_{p_j}$ , the DC width  $W(V_{p_i p_j})$  of valley  $V_{p_i p_j}$  is the horizontal DC distance from peak  $p_i$  to peak  $p_j$ 's peak area, denoted as  $P_{\text{area}}(p_j)$  (i.e., the local density area leading by peak  $p_j$ ). In other words,  $W(V_{p_i p_j})$  is the DC distance between peak  $p_i$  and its nearest same-density point  $x(p_i) \in P_{\text{area}}(p_j)$ , where  $\rho_{p_i} = \rho_{x(p_i)}$ . Therefore, according to (13), we have

$$W(V_{p_i p_j}) = \ln \rho_{p_i} + \ln \rho_{x(p_i)} - 2 \ln \rho_{b^*} \\ = 2 (\ln \rho_{p_i} - \ln \rho_{b^*}) \quad (14)$$

Equations (13) and (14) indicate that the DC distance and the valley DC width between adjacent density peaks can be obtained as long as the saddle point  $b^*$  is found in the borderline  $B_{p_i p_j}$ , as shown in Fig. 6(b).

3) **DC Surface Path Expression:** Given a path  $\tilde{\Gamma}_{x_i x_j}$ , its *density surface path* is defined as in (15). With  $\Gamma_{x_i x_j} \in \mathbb{R}^d$ ,  $\Phi(\tilde{\Gamma}_{x_i x_j}) \in \mathbb{R}^{d+1}$

$$\Phi(\tilde{\Gamma}_{x_i x_j}) \triangleq (\tilde{\Gamma}_{x_i x_j}, \rho(\tilde{\Gamma}_{x_i x_j})). \quad (15)$$

Given a function  $\xi: \mathbb{R}^d \rightarrow \mathbb{R}^1$  satisfies  $\forall x_a, x_b \in \tilde{\Gamma}_{x_i x_j}, |\xi_{x_a} - \xi_{x_b}| = DC(x_a, x_b)$ . Path  $\xi(\tilde{\Gamma}_{x_i x_j})$  as the mapping of  $\tilde{\Gamma}_{x_i x_j}$  in DC space is called a *DC path*. Then, the *DC surface path* of  $\tilde{\Gamma}_{x_i x_j}$  is defined as follows:

$$\Omega(\tilde{\Gamma}_{x_i x_j}) \triangleq (\xi(\tilde{\Gamma}_{x_i x_j}), \rho(\tilde{\Gamma}_{x_i x_j})). \quad (16)$$

Notably, mapping  $\Phi(\tilde{\Gamma}_{x_i x_j})$  to  $\Omega(\tilde{\Gamma}_{x_i x_j})$ , i.e., mapping density surface path to DC surface path, can be considered as a dimension reduction operation.

As known, path  $\tilde{\Gamma}_{x_i x_j}$  can be arbitrary shapes in feature space, so density surface path  $\Phi(\tilde{\Gamma}_{x_i x_j})$  can also own arbitrary shapes. But what is fascinating is that DC surface path

$\Omega(\tilde{\Gamma}_{x_i x_j}) \triangleq (\xi(\tilde{\Gamma}_{x_i x_j}), \rho(\tilde{\Gamma}_{x_i x_j}))$  can enjoy a standard shape (that a function combination can express), since a functional relationship exists between  $\xi(\cdot)$  and  $\rho(\cdot)$ , as will be shown later.

Given a density-change monotone path  $\tilde{\Gamma}_{x_i x_j}^\circ$ , according to (10), its DC path  $\xi(\tilde{\Gamma}_{x_i x_j}^\circ)$  is a fixed line segment with length  $DC(x_i, x_j) = |\ln(\rho_{x_i}/\rho_{x_j})| = |\xi_{x_i} - \xi_{x_j}|$  between  $\xi_{x_i}$  and  $\xi_{x_j}$  in DC space. Then, the mapping of path  $\tilde{\Gamma}_{x_i x_j}^\circ$  is a line segment at  $[\xi_{x_i}, \xi_{x_j}]$  in DC space, where  $\forall x \in \tilde{\Gamma}_{x_i x_j}^\circ$ , its mappin  $\xi_x$  in DC space is

$$\begin{aligned} \xi_x &= \xi_{x_i} + DC(x_i, x) \\ &= \xi_{x_i} + |\xi_x - \xi_{x_i}| \\ &= \xi_{x_i} + \left| \ln \frac{\rho_x}{\rho_{x_i}} \right| \\ &= \xi_{x_i} + \chi(\tilde{\Gamma}_{x_i x_j}^\circ) \cdot \ln \frac{\rho_x}{\rho_{x_i}} \end{aligned} \quad (17)$$

where function  $\chi(\cdot) \in \{-1, 0, 1\}$  returns the monotonicity of path  $\tilde{\Gamma}^\circ$  as in the following equation:

$$\chi(\tilde{\Gamma}_{x_i x_j}^\circ) = \begin{cases} 1, & \rho_{x_j} > \rho_{x_i} \\ 0, & \rho_{x_j} = \rho_{x_i} \\ -1, & \rho_{x_j} < \rho_{x_i} \end{cases} \quad (18)$$

where  $\chi(\tilde{\Gamma}^\circ) = 1$  means that  $\tilde{\Gamma}^\circ$  is a density-boosting path. Therefore, for path  $\tilde{\Gamma}_{x_i x_j}^\circ$ , the functional relation between  $\xi(\cdot)$  and  $\rho(\cdot)$  can be described as follows:

$$\begin{aligned} \rho_{\xi_x} &= \rho_{x_i} \cdot e^{\chi(\tilde{\Gamma}_{x_i x_j}^\circ)(\xi_x - \xi_{x_i})} \\ \xi_x &\in [\xi_{x_i}, \xi_{x_j}] \end{aligned} \quad (19)$$

According to (7), any shaped path  $\tilde{\Gamma}_{x_i x_j}$  can be viewed as a combination of monotone  $\tilde{\Gamma}^\circ$  paths, i.e.,  $\tilde{\Gamma}_{x_i x_j} = \bigcup_{z=0}^{n_z} \tilde{\Gamma}_{x_{(z)} x_{(z+1)}}^\circ$ . Then, its DC surface path function  $f(\cdot)$  is defined as follows:

$$f(\xi_x) = \begin{cases} \rho_{x_{(0)}} \cdot e^{\chi(\tilde{\Gamma}_{x_{(0)} x_{(1)}}^\circ)(\xi_x - \xi_{x_{(0)}})} \\ \xi_x \in [\xi_{x_{(0)}}, \xi_{x_{(1)}}] \\ \rho_{x_{(1)}} \cdot e^{\chi(\tilde{\Gamma}_{x_{(1)} x_{(2)}}^\circ)(\xi_x - \xi_{x_{(1)}})} \\ \xi_x \in (\xi_{x_{(1)}}, \xi_{x_{(2)}}] \\ \vdots \\ \rho_{x_{(n_z)}} \cdot e^{\chi(\tilde{\Gamma}_{x_{(n_z)} x_{(n_z+1)}}^\circ)(\xi_x - \xi_{x_{(n_z)}})} \\ \xi_x \in (\xi_{x_{(n_z)}}, \xi_{x_{(n_z+1)}}] \end{cases} \quad (20)$$

where  $\xi_{x_{(1)}}, \xi_{x_{(2)}}, \dots, \xi_{x_{(n_z)}}$  are the mappings of all local density extremum points  $x_{(1)}, x_{(2)}, \dots, x_{(n_z)}$  in DC space, and  $\xi_{x_{(0)}} \leftarrow \xi_{x_i}, \xi_{x_{(n_z+1)}} \leftarrow \xi_{x_j}$ . According to (20), we get the following.

**Theorem 1:** For paths  $\tilde{\Gamma}_{x_a x_b} = \bigcup_{z=0}^{n_z} \tilde{\Gamma}_{x_{(z)} x_{(z+1)}}^\circ$  and  $\tilde{\Gamma}_{x_c x_d} = \bigcup_{z=0}^{n_z} \tilde{\Gamma}_{x'_{(z)} x'_{(z+1)}}^\circ$ , if  $\rho_{x_{(m)}} = \rho_{x'_{(m)}}$ ,  $m = \{0, 1, \dots, n_z + 1\}$ , then, DC surface paths  $\Omega(\tilde{\Gamma}_{x_a x_b})$  and  $\Omega(\tilde{\Gamma}_{x_c x_d})$  have the same shape.

Inspired by Theorem 1, arbitrary-shaped density surface paths are transformed into DC surface paths to achieve a shape-standardization of density surface paths. On this basis, the DC-based minimum padding cost  $\theta$  of peaks can be easily calculated.

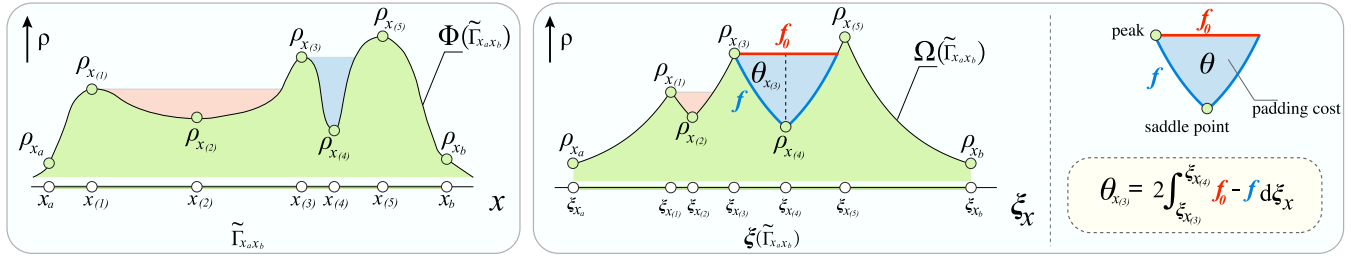


Fig. 7. Calculation process of our DC-based minimum padding cost.

4) *DC-Based Minimum Padding Cost*: Valley  $V_{p_i p_j}$  is a DC surface path  $\Omega(\tilde{\Gamma}_{p_i x(p_i)})$ , where  $x(p_i) \in P_{\text{area}}(p_j)$  and  $\rho_{p_i} = \rho_{x(p_i)}$ . According to (20), we have valley  $V_{p_i p_j}$ 's DC surface path function as follows:

$$f(\xi_x) = \begin{cases} \rho_{p_i} \cdot e^{-(\xi_x - \xi_{p_i})}, & \xi_x \in [\xi_{p_i}, \xi_{b^*}] \\ \rho_{b^*} \cdot e^{(\xi_x - \xi_{b^*})}, & \xi_x \in [\xi_{b^*}, \xi_{x(p_i)}] \end{cases} \quad (21)$$

According to (21), function  $f(\xi_x)$  is about axis  $\xi_x = \xi_{b^*}$  symmetry. Therefore, the area of  $V_{p_i p_j}$  is the enclosed area composed of function  $f(\xi_x)$  and horizontal line  $f_0(\xi_x) = \rho_{p_i}$  on interval  $\xi_x \in [\xi_{p_i}, \xi_{x(p_i)}]$ , as follows:

$$\begin{aligned} \text{Cost}(V_{p_i p_j}) &= \int_{\xi_{p_i}}^{\xi_{x(p_i)}} f_0(\xi_x) - f(\xi_x) d\xi_x \\ &= \int_{\xi_{p_i}}^{\xi_{b^*}} \rho_{p_i} - f(\xi_x) d\xi_x + \int_{\xi_{b^*}}^{\xi_{x(p_i)}} \rho_{p_i} - f(\xi_x) d\xi_x \\ &= 2 \cdot \int_{\xi_{p_i}}^{\xi_{b^*}} \rho_{p_i} - f(\xi_x) d\xi_x \\ &= 2 \cdot \rho_{p_i} \cdot \int_{\xi_{p_i}}^{\xi_{b^*}} (1 - e^{-(\xi_x - \xi_{p_i})}) d\xi_x \end{aligned} \quad (22)$$

Fig. 7 shows the calculation process of a peak's (i.e., point  $x_{(3)}$ ) DC-based minimum padding cost. As shown, a peak's DC-based minimum padding cost can be fast calculated, as long as the saddle point toward the adjacent higher density peak is found. More importantly, our DC-based minimum padding cost gives a deep valley a large cost, which ensures that PeakPad effectively captures the deep valleys between clusters.

Let  $\xi_{p_i} = 0$ , we have the interval  $[\xi_{p_i}, \xi_{b^*}]$  become  $[0, DC(p_i, b^*)] = [0, \ln(\rho_{p_i}/\rho_{b^*})]$ . Hence, area  $\text{Cost}(V_{p_i p_j})$  can be calculated as follows:

$$\begin{aligned} \text{Cost}(V_{p_i p_j}) &= 2 \cdot \rho_{p_i} \cdot \int_0^{\ln \frac{\rho_{p_i}}{\rho_{b^*}}} (1 - e^{-\xi_x}) d\xi_x \\ &= 2 \cdot \rho_{p_i} \cdot \left[ (\xi_x + e^{-\xi_x}) \right]_0^{\ln \frac{\rho_{p_i}}{\rho_{b^*}}} \\ &= 2 \cdot \rho_{p_i} \cdot \left( \ln \frac{\rho_{p_i}}{\rho_{b^*}} + \frac{\rho_{b^*}}{\rho_{p_i}} - 1 \right). \end{aligned} \quad (23)$$

According to (3), the DC-based minimum padding cost is calculated as follows:

$$\begin{aligned} \theta_{p_i} &= \min_{V_{p_i^{[t]} p_j^{[t]}}} \left( 2 \cdot \rho_{p_i} \cdot \left( \ln \frac{\rho_{p_i}}{\rho_{b^*}} + \frac{\rho_{b^*}}{\rho_{p_i}} - 1 \right) \right) \\ \text{s.t. } \exists b^* &\in B_{p_i^{[t]} p_j^{[t]}}, \rho_{p_j^{[t]}} > \rho_{p_i^{[t]}} \\ t &\in \{0, 1, \dots, n_p - 1\} \end{aligned} \quad (24)$$

$\exists b^* \in B_{p_i^{[t]} p_j^{[t]}}$  means that there is saddle point  $b^*$  between single-peak clusters  $S(p_i^{[t]})$  and  $S(p_j^{[t]})$  after  $t$  padding operations. This constraint is equivalent to  $\exists e_{p_i^{[t]} p_j^{[t]}} \in E^{[t]}$  as in (3), since  $E = \{e_{p_i p_j} | \exists B_{p_i p_j}\}$ .

5) *Hierarchical Features*: During clustering, every remaining noncenter peak  $p_i^{[t]}$  leads a single-peak cluster  $S(p_i^{[t]})$ . As a bottom-up hierarchical clustering technique, PeakPad views the padding cost between single-peak clusters as their dissimilarity, therefore, its linkage metric is defined as follows:

$$\begin{aligned} D(S(p_i^{[t]}), S(p_j^{[t]})) &= 2 \cdot \rho_{p_i} \cdot \left( \ln \frac{\rho_{p_i}}{\rho_{b^*}} + \frac{\rho_{b^*}}{\rho_{p_i}} - 1 \right) \\ \text{s.t. } \exists b^* &\in B_{p_i^{[t]} p_j^{[t]}}, \rho_{p_j^{[t]}} > \rho_{p_i^{[t]}} \\ t &\in \{0, 1, \dots, n_p - 1\} \end{aligned} \quad (25)$$

Equation (25) tells that the dissimilarity between two single-peak clusters only depends on the density values of their saddle point and the low-density peak. Based on this, the dissimilarity between single-peak clusters can be fast evaluated by fast-finding saddle points.

Fig. 8 illustrates the clustering process of the PeakPad algorithm on a toy dataset with three clusters. The PeakPad algorithm identifies four peak areas, each guided by a density peak ( $A \sim D$ ), and detects boundary areas between peak areas clusters, where the highest density point is the saddle point. If two peak areas share boundary points, they are considered connected in the peak graph.

Next, peaks are merged based on their minimum padding cost. Peak  $B$ , with the lowest cost, is first padded into peak  $C$ 's area (with  $\theta_B = 2 \cdot \rho_B \cdot (\ln(\rho_B/\rho_{b^*}) + (\rho_{b^*}/\rho_B) - 1) \approx 0.0137$ ), and removed from the list of peaks. Peak  $A$ , with a higher cost, is then padded into peak  $C$ 's area (with  $\theta_A = 2 \cdot \rho_A \cdot (\ln(\rho_A/\rho_{b^*}) + (\rho_{b^*}/\rho_A) - 1) \approx 1.3073$ ), and removed. Peaks  $C$  and  $D$  remain independent, as they have no shared boundaries, and are identified as the center peaks.

To highlight center peaks  $C$  and  $D$  in the decision graph, we set their padding cost to twice the maximum existing padding cost (i.e.,  $\theta_C = \theta_D = 2 \cdot \max(\theta_A, \theta_B) = 2.6146$ ). By observing the decision graph, we can easily identify peaks  $A$ ,  $C$ , and  $D$  as the final cluster centers based on their high padding costs  $\theta$  (while peak  $B$  is padded to peak  $C$ ), and thereby obtain the clustering result.

### E. Pseudocode and Complexity

Algorithms 1 and 2 show the pseudocode of the proposed PeakPad algorithm in two steps: 1) peak graph building and 2) peak-padding clustering.

Peak graph building (step 1) needs a computational complexity of  $O(n \log(n) + n\tilde{k} + n)$ , where  $\tilde{k}$  means that a point's

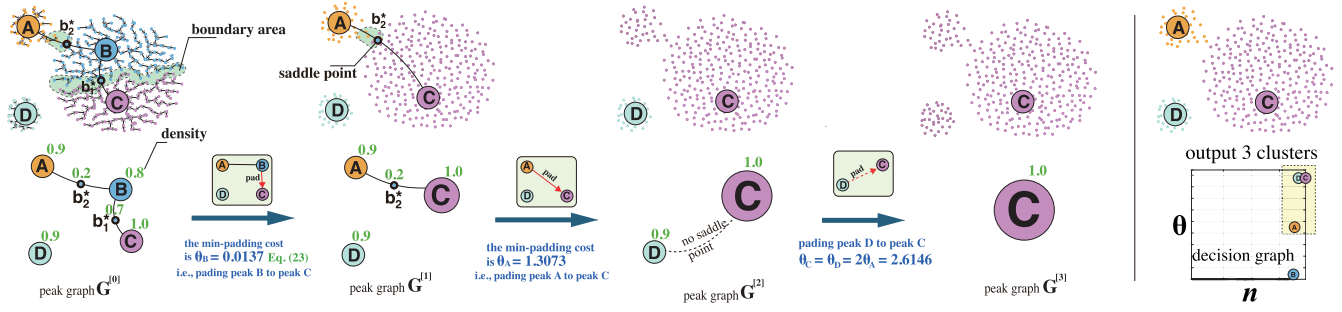


Fig. 8. Clustering process of the PeakPad algorithm on a toy dataset.

**Algorithm 1** PeakPad—Step 1: Peak Graph Building

**Input:** Dataset  $X = \{x_1, x_2, \dots, x_n\}$ , neighborhood parameter  $k$   
**Output:** Peak graph  $G(P, E)$ , single-peak clusters  $S(P) = \{S(p) \mid p \in P\}$

- 1 Compute kNN distances using fast kNN technique [31]
- 2 Estimate the density values  $\rho(X) = \{\rho_{x_i} \mid x_i \in X\}$  using Eq. (1)  
 // Eq. (1): Density estimation
- 3 Initialize peak set:  $P \leftarrow X$
- 4 Initialize data label:
- 5 **for** all points  $x_i \in X$  **do**
- 6    $Label(x_i) \leftarrow$  unique value
- 7   Set flag:  $isPeak(x_i) \leftarrow 1$  // Binary peak status
- 8 **end for**
- 9 Sort  $X$  by density descending:  $X' \leftarrow sort(X, \rho)$
- 10 **for** all points  $x_i \in X'$  (high to low density) **do**
- 11   **for** all neighbors  $x_j \in N_k(x_i)$  (near to far) **do**
- 12     **if**  $\rho_{x_j} > \rho_{x_i}$  **then**
- 13       Update status:  $isPeak(x_i) \leftarrow 0$  (see Definition 1))
- 14       Remove peak:  $P \leftarrow P \setminus \{x_i\}$
- 15       Propagate label:  $Label(x_i) \leftarrow Label(x_j)$
- 16       **break** neighbor loop
- 17     **end if**
- 18   **end for**
- 19 **end for**
- 20 Form single-peak clusters:  $S(P) \leftarrow \{\{x \mid Label(x) = p\} \mid p \in P\}$  (see Definition 2)
- 21 **for** all peak pairs  $(p_i, p_j) \in P \times P$  **do**
- 22   **if** a boundary connection  $B_{p_i p_j}$  exists (see Definition 3) **then**
- 23     Add edge:  $E \leftarrow E \cup \{e_{p_i p_j}\}$
- 24   **end if**
- 25 **end for**
- 26 **return**  $G(P, E)$  and  $S(P)$

nearest higher density point is its  $\tilde{k}$ th neighbor (an average concept). Actually, most points can find a really close higher density point, i.e.,  $\tilde{k} \ll k$ . Then, peak-padding clustering (step 2) needs a complexity of  $O(n_p^2 + n_p)$ .

Therefore, the overall computational complexity of PeakPad is  $O(n \log(n) + n\tilde{k} + n_p^2)$ , with  $\tilde{k}, n_p \ll n$ .

**Algorithm 2** PeakPad—Step 2: Peak-Padding Clustering

**Input:** Dataset  $X$ , peak set  $P$ , peak graph  $G(P, E)$ , single-peak clusters  $S(P) = \{S(p) \mid p \in P\}$   
**Output:** Final clustering result  $C = \{C_1, C_2, \dots, C_{n_c}\}$

- 1 Initialize iteration counter:  $t \leftarrow 0$
- 2 Initialize peak set:  $P^{[t]} \leftarrow P$
- 3 Initialize peak graph:  $G^{[t]} \leftarrow G(P, E)$
- 4 **while**  $|P^{[t]}| > 1$  **do**
- 5   **for** each peak  $p_i \in P^{[t]}$  **do**
- 6     Compute minimum padding cost  $\theta_{p_i}$  via Eq. (24)  
 // Eq. (24): Padding cost definition.
- 7   **end for**
- 8   Record the max padding cost in  $P^{[t]}$ :  $\theta_{max} = \max_{p \in P^{[t]}} \theta_p$
- 9   **for** each peak  $\hat{p} \in P^{[t]}$  with  $\theta_{\hat{p}} = \theta_{max}$  **do**
- 10     Set  $\theta_{\hat{p}} \leftarrow \beta \cdot \theta_{max}$ , with the default value  $\beta = 2$ .  
 // Eq. (4): Padding cost for center peaks
- 11   **end for**
- 12   Select optimal peak to pad:  $p^* \leftarrow \arg \min_{p \in P^{[t]}} \theta_p$
- 13   Update peak set:  $P^{[t+1]} \leftarrow P^{[t]} \setminus \{p^*\}$
- 14   Increment counter:  $t \leftarrow t + 1$
- 15   Rebuild peak graph:  $G^{[t]}$
- 16 **end while**
- 17 Manually select peaks with top  $\theta$  values as cluster centers from decision graph
- 18 Pad (assign) remaining non-center peaks into higher-density peaks with the minimum padding cost
- 19 Associate peaks (representing single-peak clusters) to form final clusters
- 20 **return** Clustering result  $C = \{C_1, C_2, \dots, C_{n_c}\}$

## IV. BENCHMARK TESTS

## A. Test Set Up

**Datasets:** Fifteen commonly used synthetic datasets and thirteen popular real-world datasets are selected to benchmark the proposed clustering algorithm, as shown in Table I. Additionally, 160 synthetic datasets with varying geometric structures and dimensions—including 80 Gaussian sphere datasets and 80 ellipsoidal cluster datasets [54]—are used to evaluate the cluster center detection performance.

**Comparison algorithms and settings:** Four classic clustering algorithms (K-means [47], DBSCAN [26], MSC [6], and self-tuning spectral clustering (SSC) [33]), several recent DPC-based methods (DPC [8], SSSP-DPC [11], SNN-DPC

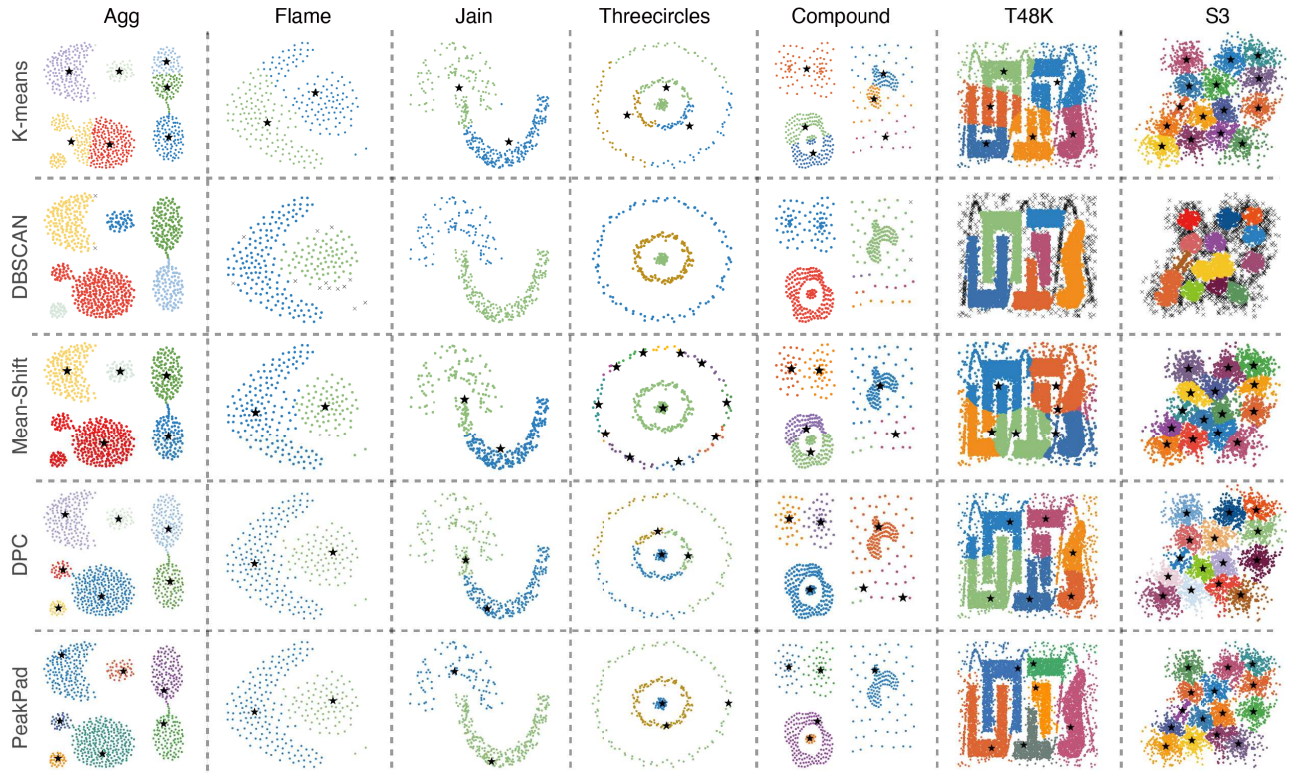


Fig. 9. Results of different algorithms on synthetic datasets.

TABLE I  
DATASETS SUMMARY

| Synthetic Datasets |        |       |                  | Real-world Datasets |       |        |                 |
|--------------------|--------|-------|------------------|---------------------|-------|--------|-----------------|
| Dataset            | Inst.  | Attr. | Clus.            | Dataset             | Inst. | Attr.  | Clus.           |
| Agg                | 788    | 2     | 7 <sup>†</sup>   | Iris                | 150   | 4      | 3 <sup>‡</sup>  |
| Flame              | 240    | 2     | 3 <sup>†</sup>   | Wine                | 178   | 13     | 3 <sup>‡</sup>  |
| Jain               | 373    | 2     | 2 <sup>†</sup>   | Segment             | 2310  | 19     | 7 <sup>‡</sup>  |
| Threecircles       | 299    | 2     | 3 <sup>*</sup>   | Drivepoints         | 606   | 16     | 4 <sup>‡</sup>  |
| Sticks             | 512    | 2     | 4 <sup>*</sup>   | Breastcancer        | 569   | 30     | 2 <sup>‡</sup>  |
| Compound           | 399    | 2     | 6 <sup>†</sup>   | Waveform            | 5000  | 21     | 3 <sup>‡</sup>  |
| R15                | 600    | 2     | 15 <sup>†</sup>  | Movement            | 360   | 90     | 15 <sup>‡</sup> |
| Spiral             | 312    | 2     | 3 <sup>†</sup>   | Parkin              | 195   | 22     | 2 <sup>‡</sup>  |
| T48K               | 8000   | 2     | 6 <sup>†</sup>   | Vote                | 435   | 16     | 2 <sup>‡</sup>  |
| S1                 | 5000   | 2     | 15 <sup>†</sup>  | YTF                 | 10000 | 10     | 41 <sup>§</sup> |
| S3                 | 5000   | 2     | 15 <sup>†</sup>  | USPS                | 11000 | 10     | 10 <sup>§</sup> |
| A2                 | 5250   | 2     | 35 <sup>†</sup>  | MNIST               | 10000 | 500    | 10 <sup>¶</sup> |
| A3                 | 7500   | 2     | 50 <sup>†</sup>  | OlivettiFaces       | 400   | 92×112 | 40 <sup>°</sup> |
| Birchrg1           | 100000 | 2     | 100 <sup>†</sup> |                     |       |        |                 |
| TB10w              | 100000 | 2     | 2 <sup>‡</sup>   |                     |       |        |                 |

<sup>†</sup> [36], <sup>‡</sup> [37], <sup>\*</sup> [33], <sup>‡</sup> [38], <sup>§</sup> [39], <sup>¶</sup> [41], <sup>°</sup> [42]

[9], UP-DPC [17], DBSCAN-DPC [34], GB-DP [19], and TC [50]), and the proposed PeakPad are presented for comparison. The parameters of algorithms are set according to their best performance over a large range of possible configurations. For *K*-means and SSC, the best results among ten runs are selected; while for DPC-based algorithms, appropriate density peaks are selected as cluster centers by observing decision graphs.

**Data preprocessing:** The min-max normalization [35] is applied to preprocess datasets, aiming to avoid the difference of dimensional metrics.

**Machine configuration:** MATLAB (R2019b) on Windows Server 2019 Datacenter with Intel<sup>2</sup> Xeon<sup>2</sup> Gold 5220R CPU at 2.20 GHz (2 processors) and 256 GB RAM.

**Evaluation metric:** The popular adjusted rand index (ARI) [43], adjusted mutual information (AMI) [43], normalized mutual information (NMI) [44], Fowlkes–Mallows index (FMI) [45], accuracy (ACC), and *F*-score [46] are used for the clustering performance evaluation. Besides, the *F1*-score [48] is applied to measure the robustness of center detection.

### B. Results on Synthetic Datasets

Fig. 9 presents the clustering results of different algorithms. Here, “★” denotes detected cluster centers, “×” marks identified noise points, and distinct colors correspond to different clusters.

As illustrated in Fig. 9, PeakPad achieves near-perfect clustering across all tested datasets. DBSCAN performs well in reconstructing complex shapes but tends to misclassify many border points as noise in the *Agg*, *Compound*, and *S3* datasets. Although DPC correctly clusters *Agg*, *Flame*, and *S3*, it fails on *Jain*, *Threecircles*, *Compound*, and *T48K* datasets due to inaccurate cluster center identification. MSC produces satisfactory results on *Flame* and *S3*, but suffers from oversegmentation on *Threecircles*, *Compound*, and *T48K*, and incorrect center detection on *Jain*. *K*-means fails to handle all complex-shaped datasets.

Table II summarizes quantitative scores of all algorithms, with the best results highlighted. PeakPad consistently achieves top performance across nearly all synthetic datasets. sparse subspace clustering (SSC) outperforms *K*-means on complex shape recognition tasks. Notably, improved DPC variants—SSSP-DPC, SNN-DPC, TC, and

<sup>2</sup>Registered trademark.

TABLE II  
COMPARISON ON SYNTHETIC DATASETS (METRICS: AMI/ARI | NMI/F-SCORE | FMI/ACC)

| Dataset      | K-means          | SSC              | DBSCAN           | MSC              | DPC              | SSSP             | SNN-DPC          | UP-DPC           | DB-DPC           | GB-DP            | TC               | PeakPad          |
|--------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| Agg          | 0.82 0.76        | 0.96 0.97        | 0.91 0.91        | 0.83 0.82        | <b>0.99 0.99</b> | 0.97 0.97        | 0.93 0.93        | 0.92 0.88        | 0.80 0.81        | <b>0.99 0.99</b> | 0.72 0.73        | <b>0.99 0.99</b> |
|              | 0.85 0.85        | 0.97 <b>0.99</b> | 0.95 0.94        | 0.90 0.89        | <b>0.99 0.99</b> | 0.97 0.98        | 0.93 0.96        | 0.92 0.91        | 0.89 0.85        | <b>0.99 0.99</b> | 0.84 0.79        | <b>0.99 0.99</b> |
|              | 0.81 0.91        | 0.98 <b>0.99</b> | 0.93 0.96        | 0.87 0.91        | <b>0.99 0.99</b> | 0.98 0.98        | 0.94 0.96        | 0.91 0.91        | 0.87 0.83        | <b>0.99 0.99</b> | 0.82 0.78        | <b>0.99 0.99</b> |
| Flame        | 0.45 0.49        | 0.54 0.61        | 0.68 0.87        | 0.80 0.87        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.91 0.95        | 0.85 0.90        | <b>1.00 1.00</b> | 0.80 0.87        | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
|              | 0.47 0.85        | 0.55 0.89        | 0.76 0.96        | 0.81 0.97        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.91 0.99        | 0.85 0.97        | <b>1.00 1.00</b> | 0.81 0.97        | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
|              | 0.75 0.85        | 0.81 0.89        | 0.94 0.89        | 0.94 0.97        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.98 0.99        | 0.96 0.98        | <b>1.00 1.00</b> | 0.94 0.97        | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
| Jain         | 0.49 0.58        | 0.63 0.73        | <b>1.00 1.00</b> | 0.49 0.55        | 0.54 0.62        | <b>1.00 1.00</b> | 0.55 0.59        | <b>1.00 1.00</b> | 0.18 -0.03       | <b>1.00 1.00</b> | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
|              | 0.53 0.89        | 0.67 0.93        | <b>1.00 1.00</b> | 0.53 0.88        | 0.58 0.90        | 0.39 0.80        | <b>1.00 1.00</b> | 0.55 0.89        | <b>1.00 1.00</b> | 0.18 0.60        | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
|              | 0.82 0.88        | 0.89 0.93        | <b>1.00 1.00</b> | 0.81 0.87        | 0.84 0.90        | 0.70 0.79        | <b>1.00 1.00</b> | 0.82 0.88        | <b>1.00 1.00</b> | 0.58 0.74        | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
| Threecircles | 0.16 0.06        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.45 0.41        | 0.18 0.03        | 0.51 0.40        | <b>1.00 1.00</b> | 0.54 0.36        | <b>1.00 1.00</b> | 0.05 -0.01       | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
|              | 0.16 0.43        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.53 0.56        | 0.23 0.53        | 0.61 0.73        | <b>1.00 1.00</b> | 0.55 0.65        | <b>1.00 1.00</b> | 0.06 0.42        | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
|              | 0.40 0.48        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.65 0.80        | 0.51 0.58        | 0.70 0.67        | <b>1.00 1.00</b> | 0.62 0.69        | <b>1.00 1.00</b> | 0.38 0.46        | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
| Sticks       | 0.63 0.49        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.71 0.65        | 0.82 0.77        | 0.87 0.83        | <b>1.00 1.00</b> | 0.82 0.69        | <b>1.00 1.00</b> | 0.57 0.42        | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
|              | 0.67 0.66        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.78 0.77        | 0.85 0.85        | 0.89 0.92        | <b>1.00 1.00</b> | 0.82 0.66        | <b>1.00 1.00</b> | 0.59 0.62        | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
|              | 0.64 0.71        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.73 0.91        | 0.84 0.87        | 0.88 0.92        | <b>1.00 1.00</b> | 0.79 0.72        | <b>1.00 1.00</b> | 0.58 0.64        | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
| Compound     | 0.67 0.52        | 0.71 0.49        | 0.70 0.76        | 0.74 0.58        | 0.79 0.81        | 0.84 0.83        | 0.82 0.85        | 0.66 0.42        | <b>0.84 0.86</b> | 0.61 0.48        | 0.67 0.74        | <b>0.84 0.85</b> |
|              | 0.68 0.66        | 0.73 0.66        | 0.79 0.77        | 0.76 0.71        | 0.85 0.84        | 0.84 0.85        | 0.89 0.88        | 0.67 0.65        | <b>0.91 0.89</b> | 0.65 0.63        | 0.81 0.77        | <b>0.91 0.89</b> |
|              | 0.63 0.76        | 0.61 0.78        | 0.84 0.79        | 0.68 0.86        | 0.87 <b>0.88</b> | 0.87 0.87        | <b>0.90 0.87</b> | 0.55 0.64        | <b>0.90 0.88</b> | 0.59 0.76        | 0.83 0.74        | <b>0.90 0.87</b> |
| R15          | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.98</b> | <b>0.99 0.98</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | 0.81 0.59        | <b>0.99 0.99</b> | 0.98 0.97        | <b>0.99 0.99</b> | <b>0.99 0.99</b> |
|              | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | 0.82 0.49        | <b>0.99 0.99</b> | 0.98 0.99        | <b>0.99 0.99</b> | <b>0.99 0.99</b> |
|              | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.98 0.99</b> | <b>0.98 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | 0.64 0.56        | <b>0.99 0.99</b> | 0.98 0.99        | <b>0.99 0.99</b> | <b>0.99 0.99</b> |
| Spiral       | -0.01 -0.01      | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.25 0.12        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.01 0.02        | <b>1.00 1.00</b> | 0.01 0.01        | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
|              | 0.00 0.35        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.40 0.36        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.02 0.42        | <b>1.00 1.00</b> | 0.02 0.41        | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
|              | 0.33 0.35        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.30 0.77        | <b>1.00 1.00</b> | <b>1.00 1.00</b> | <b>1.00 1.00</b> | 0.34 0.42        | <b>1.00 1.00</b> | 0.36 0.39        | <b>1.00 1.00</b> | <b>1.00 1.00</b> |
| T48K         | 0.57 0.48        | 0.67 0.56        | <b>0.93 0.95</b> | 0.48 0.35        | 0.63 0.54        | 0.77 0.76        | 0.59 0.50        | 0.67 0.61        | 0.83 0.86        | 0.51 0.30        | 0.84 0.87        | 0.83 0.86        |
|              | 0.58 0.63        | 0.68 0.72        | <b>0.94 0.97</b> | 0.48 0.53        | 0.66 0.67        | 0.81 0.79        | 0.61 0.65        | 0.69 0.66        | 0.87 0.88        | 0.51 0.55        | 0.87 0.88        | 0.87 0.89        |
|              | 0.57 0.71        | 0.63 0.78        | <b>0.96 0.97</b> | 0.47 0.58        | 0.62 0.74        | 0.80 0.82        | 0.57 0.64        | 0.68 0.72        | 0.89 0.91        | 0.44 0.56        | 0.89 0.90        | 0.89 0.90        |
| S1           | 0.96 0.91        | <b>0.99 0.99</b> | 0.94 0.96        | 0.98 0.98        | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | 0.89 0.70        | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> |
|              | 0.97 0.94        | <b>0.99 0.99</b> | 0.96 0.98        | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | 0.88 0.59        | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> |
|              | 0.92 0.93        | <b>0.99 0.99</b> | 0.97 0.99        | 0.98 0.99        | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | 0.73 0.64        | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> | <b>0.99 0.99</b> |
| S3           | 0.87 0.81        | 0.90 0.86        | 0.73 0.51        | 0.88 0.84        | 0.94 0.93        | 0.88 0.83        | 0.88 0.83        | 0.87 0.80        | 0.91 0.91        | 0.79 0.88        | 0.88 0.83        | <b>0.96 0.95</b> |
|              | 0.87 0.88        | 0.90 0.93        | 0.76 0.79        | 0.88 0.92        | 0.94 0.96        | 0.88 0.84        | 0.88 0.91        | 0.87 0.85        | 0.92 0.95        | 0.81 0.97        | 0.88 0.91        | <b>0.96 0.97</b> |
|              | 0.82 0.89        | 0.87 0.93        | 0.56 0.70        | 0.85 0.92        | 0.93 0.96        | 0.84 0.91        | 0.84 0.91        | 0.81 0.87        | 0.91 0.96        | 0.94 0.97        | 0.84 0.91        | <b>0.96 0.97</b> |
| A2           | 0.97 0.94        | <b>0.99 0.98</b> | 0.82 0.65        | 0.98 0.97        | <b>0.99 0.98</b> | 0.98 0.97        | 0.98 0.98        | 0.89 0.67        | <b>0.99 0.98</b> | 0.95 0.91        | 0.98 0.96        | <b>0.99 0.98</b> |
|              | 0.96 0.96        | <b>0.99 0.99</b> | 0.86 0.87        | 0.98 0.98        | <b>0.99 0.99</b> | 0.98 0.99        | 0.98 0.99        | 0.89 0.66        | <b>0.99 0.99</b> | 0.95 0.96        | 0.98 0.98        | <b>0.99 0.99</b> |
|              | 0.94 0.96        | <b>0.98 0.99</b> | 0.67 0.86        | 0.97 0.98        | <b>0.98 0.99</b> | 0.97 0.99        | <b>0.98 0.99</b> | 0.67 0.68        | <b>0.98 0.99</b> | 0.92 0.96        | 0.97 0.98        | <b>0.98 0.99</b> |
| A3           | 0.97 0.91        | <b>0.99 0.99</b> | 0.82 0.71        | 0.98 0.97        | <b>0.99 0.98</b> | 0.98 0.97        | <b>0.99 0.97</b> | 0.93 0.78        | 0.98 0.98        | 0.94 0.85        | 0.98 0.97        | <b>0.99 0.98</b> |
|              | 0.97 0.93        | <b>0.99 0.99</b> | 0.88 0.88        | 0.98 0.98        | <b>0.99 0.99</b> | 0.98 0.99        | <b>0.99 0.99</b> | 0.94 0.75        | <b>0.99 0.99</b> | 0.94 0.90        | 0.98 0.99        | <b>0.99 0.99</b> |
|              | 0.91 0.94        | <b>0.99 0.99</b> | 0.72 0.90        | 0.97 0.98        | 0.98 0.99        | 0.97 0.99        | <b>0.97 0.99</b> | 0.79 0.79        | 0.98 0.99        | 0.85 0.89        | 0.97 0.99        | 0.98 0.99        |
| Birchrg1     | 0.94 0.86        | --               | 0.80 0.33        | 0.79 0.42        | --               | --               | --               | 0.82 0.51        | 0.96 0.94        | 0.86 0.67        | --               | <b>0.98 0.99</b> |
|              | 0.95 0.90        | --               | 0.86 0.88        | 0.81 0.28        | --               | --               | --               | 0.84 0.47        | 0.97 0.97        | 0.89 0.77        | --               | <b>0.98 0.99</b> |
|              | 0.86 0.91        | --               | 0.40 0.79        | 0.49 0.42        | --               | --               | --               | 0.53 0.53        | 0.94 0.97        | 0.68 0.77        | --               | <b>0.99 0.99</b> |
| TB10w        | 0.41 0.51        | --               | 0.90 0.95        | 0.73 0.82        | --               | --               | --               | 0.38 0.31        | <b>0.97 0.99</b> | 0.54 0.53        | --               | <b>0.98 0.99</b> |
|              | 0.41 0.86        | --               | 0.90 <b>0.99</b> | 0.73 0.95        | --               | --               | --               | 0.38 0.77        | <b>0.97 0.99</b> | 0.54 0.86        | --               | <b>0.98 0.99</b> |
|              | 0.76 0.86        | --               | 0.97 <b>0.99</b> | 0.91 0.95        | --               | --               | --               | 0.69 0.78        | <b>0.99 0.99</b> | 0.77 0.86        | --               | <b>0.99 0.99</b> |

Note: symbol "--" means algorithm memory exhaustion, "SSSP" means SSSP-DPC, and "DB-DPC" means DBSCAN-DPC

DBSCAN-DPC—demonstrate enhanced accuracy compared to the original DPC. Furthermore, GB-DP, DBSCAN-DPC, and UP-DPC exhibit superior scalability, efficiently processing large datasets such as *Birchrg1* and *TB10w* with 100 000 points each. In contrast, DPC and several of its improved versions (SSSP-DPC, SNN-DPC, and TC) fail to execute under the current computational environment due to excessive memory demands caused by their algorithmic complexity.

Overall, the test results validate that PeakPad delivers robust and accurate recognition of complex cluster shapes, demonstrating superior adaptability and computational efficiency compared to competing methods.

### C. Results on Real-World Datasets

To further evaluate the clustering performance of PeakPad on large-scale and high-dimensional datasets, we conducted benchmark tests on ten widely used real-world datasets. These include nine UCI datasets (*Iris*, *Wine*, *Segment*, *Drivepoints*, *Breastcancer*, *Waveform*, *Movementlibras*, *Parkin*, and *Vote*), three large-scale machine learning datasets (*YTF*, *MNIST* pre-processed by [7]), and the well-known *OlivettiFaces* dataset [42] pre-processed by [9]. Table I summarizes the key characteristics of these datasets, encompassing a variety of data dimensions and cluster complexities.

The quantitative results presented in Table III demonstrate that PeakPad consistently delivers superior or competitive performance across all datasets, with the best results

highlighted. Furthermore, Table IV lists the parameter settings for all comparison algorithms (except for the parameter-free TC algorithm [50]), ensuring a fair evaluation.

Fig. 10 visualizes the clustering results on selected complex, high-dimensional datasets using t-SNE [49]. It can be observed that both DBSCAN-DPC and PeakPad achieve clustering structures closely aligned with the true labels, particularly on challenging datasets like *MNIST*. This highlights PeakPad's ability to effectively capture intrinsic data structures even in complex feature spaces.

Overall, these extensive tests validate PeakPad as a robust and effective clustering method, capable of handling diverse and high-dimensional real-world data, thus demonstrating its potential for practical applications.

### D. Robustness of Center Detection

*F1*-score [48] [as in (26)] is used to quantify the center detection robustness of PeakPad, where true positives (TPs) refers to the correctly identified cluster centers that match the ground-truth centers (i.e., the highest density points in each true cluster), false positives (FPs) refers to the incorrect or extra cluster centers detected, and false negatives (FNs) refers to the ground-truth centers that were missed by the algorithm.

To verify the superiority of PeakPad, for each test, each DPC-based algorithm ( $\gamma$ -based center detection) directly selects its true number of centers with the top largest  $\gamma$  (i.e.,  $\rho \cdot \delta$ ) values to obtain the *F1*-score, while PeakPad ( $\theta$ -based center detection) selects the true number of centers with the

TABLE III  
COMPARISON ON REAL-WORLD DATASETS (METRICS: AMI/ARI/NMI/F-SCORE [FMI/ACC])

| Dataset         | K-means   | SSC              | DBSCAN    | MSC              | DPC              | SSSP             | SNN-DPC          | UP-DPC    | DB-DPC           | GB-DP            | TC               | PeakPad          |
|-----------------|-----------|------------------|-----------|------------------|------------------|------------------|------------------|-----------|------------------|------------------|------------------|------------------|
| Iris            | 0.73 0.72 | 0.85 0.87        | 0.49 0.50 | 0.70 0.64        | 0.57 0.45        | 0.88 0.90        | <b>0.91 0.92</b> | 0.71 0.56 | 0.73 0.66        | 0.75 0.70        | 0.57 0.56        | 0.88 0.90        |
|                 | 0.74 0.89 | 0.86 0.95        | 0.57 0.72 | 0.72 0.83        | 0.65 0.71        | 0.89 <b>0.97</b> | <b>0.91 0.97</b> | 0.71 0.60 | 0.75 0.85        | 0.77 0.87        | 0.73 0.78        | 0.88 <b>0.97</b> |
|                 | 0.81 0.89 | 0.91 0.95        | 0.67 0.77 | 0.76 0.85        | 0.69 0.67        | 0.94 <b>0.97</b> | <b>0.95 0.97</b> | 0.76 0.69 | 0.78 0.85        | 0.80 0.88        | 0.77 0.67        | 0.93 <b>0.97</b> |
| Wine            | 0.83 0.85 | <b>0.89 0.91</b> | 0.51 0.42 | 0.67 0.78        | 0.71 0.67        | 0.75 0.74        | 0.87 0.90        | 0.39 0.42 | 0.71 0.76        | 0.81 0.82        | 0.24 0.17        | 0.74 0.73        |
|                 | 0.83 0.95 | <b>0.89 0.97</b> | 0.54 0.70 | 0.77 0.92        | 0.71 0.88        | 0.75 0.91        | 0.88 <b>0.97</b> | 0.40 0.71 | 0.76 0.91        | 0.82 0.93        | 0.34 0.47        | 0.74 0.90        |
|                 | 0.90 0.95 | <b>0.94 0.97</b> | 0.65 0.70 | 0.85 0.96        | 0.78 0.88        | 0.83 0.91        | 0.93 <b>0.97</b> | 0.62 0.71 | 0.84 0.93        | 0.88 0.93        | 0.36 0.72        | 0.82 0.90        |
| Segment         | 0.61 0.50 | 0.65 0.57        | 0.44 0.22 | 0.59 0.48        | 0.70 0.59        | 0.65 0.46        | 0.67 0.58        | 0.42 0.31 | 0.66 0.57        | 0.53 0.37        | 0.46 0.10        | <b>0.71 0.60</b> |
|                 | 0.64 0.66 | 0.66 0.71        | 0.51 0.49 | 0.71 0.67        | <b>0.73 0.74</b> | <b>0.73 0.69</b> | <b>0.70 0.76</b> | 0.43 0.44 | 0.67 0.70        | 0.55 0.59        | 0.35 0.39        | <b>0.73 0.72</b> |
|                 | 0.58 0.63 | 0.63 0.69        | 0.43 0.50 | 0.63 0.58        | <b>0.66 0.75</b> | 0.59 0.68        | 0.65 0.72        | 0.42 0.48 | 0.63 0.72        | 0.48 0.60        | 0.43 0.28        | <b>0.66 0.74</b> |
| Drivepoints     | 0.60 0.52 | 0.61 0.52        | 0.66 0.68 | 0.59 0.64        | 0.77 0.73        | 0.78 0.76        | <b>0.79 0.76</b> | 0.78 0.75 | 0.73 0.75        | 0.78 0.77        | 0.76 0.78        | <b>0.78 0.79</b> |
|                 | 0.61 0.72 | 0.61 0.72        | 0.71 0.80 | 0.71 0.78        | 0.77 0.81        | 0.79 0.84        | 0.79 0.82        | 0.78 0.78 | 0.78 0.84        | 0.85 <b>0.86</b> | 0.85 <b>0.86</b> | <b>0.87 0.86</b> |
|                 | 0.65 0.76 | 0.64 0.76        | 0.76 0.83 | 0.73 <b>0.90</b> | 0.81 0.85        | 0.83 0.85        | 0.82 0.85        | 0.82 0.83 | 0.83 0.84        | 0.84 0.85        | 0.85 0.84        | <b>0.86 0.85</b> |
| Breastcancer    | 0.61 0.73 | 0.69 0.80        | 0.34 0.45 | 0.21 0.39        | 0.41 0.47        | 0.02 -0.02       | <b>0.75 0.85</b> | 0.41 0.41 | 0.64 0.75        | 0.55 0.67        | 0.29 0.41        | 0.63 0.74        |
|                 | 0.62 0.93 | 0.70 0.95        | 0.35 0.84 | 0.33 0.77        | 0.46 0.84        | 0.04 0.67        | <b>0.76 0.96</b> | 0.41 0.82 | 0.64 0.93        | 0.55 0.91        | 0.40 0.72        | 0.64 0.93        |
|                 | 0.88 0.93 | 0.91 0.95        | 0.73 0.84 | 0.72 0.86        | 0.79 0.85        | 0.70 0.63        | <b>0.93 0.96</b> | 0.71 0.82 | 0.88 0.94        | 0.84 0.91        | 0.67 0.90        | 0.88 0.93        |
| Waveform        | 0.36 0.25 | 0.37 0.25        | 0.01 0.00 | 0.36 0.25        | 0.22 0.19        | 0.21 0.20        | <b>0.40 0.42</b> | 0.38 0.28 | 0.28 0.27        | 0.39 0.31        | 0.30 0.28        | 0.39 0.31        |
|                 | 0.36 0.53 | 0.37 0.52        | 0.01 0.46 | 0.36 0.53        | 0.22 0.56        | 0.20 0.58        | <b>0.40 0.74</b> | 0.38 0.58 | 0.28 0.63        | 0.39 0.61        | 0.29 0.63        | 0.39 0.60        |
|                 | 0.50 0.53 | 0.50 0.52        | 0.46 0.39 | 0.50 0.53        | 0.48 0.57        | 0.53 0.56        | <b>0.62 0.75</b> | 0.52 0.58 | 0.57 0.59        | 0.54 0.62        | 0.58 0.59        | 0.53 0.67        |
| Movementslibras | 0.52 0.31 | 0.56 0.34        | 0.15 0.03 | 0.47 0.28        | 0.48 0.26        | 0.49 0.27        | <b>0.58 0.39</b> | 0.37 0.11 | <b>0.58 0.36</b> | 0.40 0.22        | 0.53 0.36        | <b>0.58 0.36</b> |
|                 | 0.59 0.50 | 0.62 0.53        | 0.27 0.27 | 0.65 0.47        | 0.58 0.46        | 0.59 0.46        | <b>0.66 0.56</b> | 0.45 0.27 | 0.65 0.55        | 0.49 0.43        | 0.63 0.52        | <b>0.66 0.54</b> |
|                 | 0.36 0.47 | 0.39 0.51        | 0.24 0.21 | 0.33 0.64        | 0.35 0.45        | 0.36 0.43        | <b>0.45 0.53</b> | 0.24 0.30 | 0.40 <b>0.58</b> | 0.29 0.39        | 0.43 0.46        | 0.42 0.52        |
| Parkin          | 0.00 0.01 | 0.02 -0.04       | 0.12 0.00 | 0.06 0.05        | 0.04 0.07        | 0.03 0.05        | 0.01 0.05        | 0.00 0.00 | <b>0.25 0.02</b> | 0.21 <b>0.10</b> | 0.06 0.01        | 0.14 0.09        |
|                 | 0.00 0.57 | 0.02 0.67        | 0.11 0.66 | 0.05 0.58        | 0.04 0.65        | 0.03 0.63        | 0.01 0.63        | 0.00 0.55 | <b>0.18 0.19</b> | 0.23 0.69        | 0.10 0.09        | 0.12 0.70        |
|                 | 0.54 0.68 | 0.65 0.68        | 0.64 0.73 | 0.55 0.71        | 0.57 0.68        | 0.56 0.68        | 0.59 0.68        | 0.54 0.53 | 0.19 <b>0.87</b> | 0.61 0.75        | 0.14 0.78        | <b>0.75 0.72</b> |
| Vote            | 0.44 0.54 | <b>0.50 0.59</b> | 0.33 0.30 | 0.34 0.53        | <b>0.50 0.56</b> | <b>0.50 0.56</b> | 0.48 0.54        | 0.43 0.55 | 0.43 0.57        | 0.33 0.38        | 0.40 0.47        | 0.41 0.47        |
|                 | 0.50 0.87 | <b>0.50 0.88</b> | 0.35 0.77 | 0.42 0.87        | <b>0.51 0.88</b> | <b>0.51 0.88</b> | 0.49 0.87        | 0.43 0.87 | 0.48 <b>0.88</b> | 0.35 0.81        | 0.42 0.85        | 0.43 0.85        |
|                 | 0.78 0.87 | <b>0.80 0.88</b> | 0.68 0.82 | 0.76 <b>0.89</b> | 0.79 0.88        | 0.79 0.88        | 0.77 0.87        | 0.78 0.87 | 0.78 0.88        | 0.71 0.81        | 0.75 0.84        | 0.75 0.84        |
| YTF             | 0.75 0.57 | 0.75 0.50        | 0.67 0.34 | <b>0.78 0.68</b> | 0.73 0.50        | 0.75 0.51        | 0.72 0.46        | 0.71 0.44 | 0.62 0.29        | 0.65 0.36        | <b>0.78 0.63</b> | <b>0.78 0.54</b> |
|                 | 0.76 0.63 | 0.79 0.62        | 0.77 0.59 | 0.87 <b>0.71</b> | 0.76 0.59        | 0.78 0.62        | 0.76 0.56        | 0.72 0.49 | 0.71 0.58        | 0.67 0.49        | <b>0.88 0.70</b> | 0.81 0.65        |
|                 | 0.58 0.69 | 0.54 0.67        | 0.37 0.86 | <b>0.70 0.94</b> | 0.52 0.62        | 0.54 0.67        | 0.50 0.63        | 0.46 0.48 | 0.39 0.62        | 0.39 0.61        | 0.68 0.59        | 0.58 0.75        |
| USPS            | 0.58 0.47 | 0.71 0.54        | 0.27 0.03 | 0.60 0.42        | 0.39 0.29        | 0.65 0.55        | 0.62 0.43        | 0.55 0.37 | 0.72 <b>0.62</b> | 0.48 0.34        | 0.55 0.38        | <b>0.75 0.62</b> |
|                 | 0.59 0.65 | 0.74 0.67        | 0.25 0.26 | 0.66 0.63        | 0.44 0.47        | 0.75 0.68        | 0.65 0.62        | 0.56 0.50 | 0.73 0.76        | 0.52 0.52        | 0.68 0.52        | <b>0.77 0.78</b> |
|                 | 0.53 0.65 | 0.61 0.68        | 0.30 0.27 | 0.52 0.61        | 0.40 0.45        | 0.65 0.57        | 0.51 0.63        | 0.45 0.52 | 0.66 <b>0.78</b> | 0.44 0.52        | 0.49 0.37        | <b>0.67 0.78</b> |
| MNIST           | 0.84 0.79 | 0.89 0.83        | 0.56 0.21 | 0.70 0.42        | 0.39 0.27        | 0.67 0.31        | 0.79 0.63        | 0.65 0.43 | 0.92 0.93        | 0.59 0.25        | 0.88 0.85        | <b>0.93 0.94</b> |
|                 | 0.86 0.87 | 0.90 0.88        | 0.55 0.55 | 0.70 0.49        | 0.48 0.44        | 0.63 0.51        | 0.83 0.76        | 0.65 0.45 | 0.92 <b>0.97</b> | 0.58 0.51        | 0.89 0.92        | <b>0.93 0.97</b> |
|                 | 0.82 0.86 | 0.85 0.88        | 0.39 0.50 | 0.55 0.56        | 0.42 0.42        | 0.51 0.41        | 0.69 0.78        | 0.53 0.52 | 0.94 <b>0.97</b> | 0.43 0.50        | 0.87 0.92        | <b>0.95 0.97</b> |
| OlivettiFaces   | 0.73 0.57 | 0.77 0.66        | 0.74 0.59 | 0.64 0.65        | 0.73 0.54        | 0.79 0.63        | 0.81 0.68        | 0.71 0.52 | 0.78 0.64        | 0.61 0.47        | <b>0.82 0.71</b> | <b>0.82 0.70</b> |
|                 | 0.85 0.72 | 0.87 0.76        | 0.88 0.75 | 0.89 0.73        | 0.85 0.71        | 0.89 0.73        | 0.90 0.79        | 0.83 0.57 | 0.88 0.76        | 0.78 0.61        | <b>0.91 0.81</b> | <b>0.90 0.81</b> |
|                 | 0.59 0.72 | 0.67 0.79        | 0.60 0.87 | 0.69 <b>0.97</b> | 0.56 0.70        | 0.65 0.73        | 0.69 0.80        | 0.53 0.61 | 0.65 0.75        | 0.49 0.60        | <b>0.72 0.80</b> | 0.71 0.81        |

Note: "SSSP" means SSSP-DPC, and "DB-DPC" means DBSCAN-DPC

TABLE IV  
PARAMETER SETTINGS OF DIFFERENT ALGORITHMS ON THE TESTED DATASETS

| Dataset         | K-means | SSC     | DBSCAN       | MSC         | DPC     | SSSP-DPC | SNN-DPC | UP-DPC      | DBSCAN-DPC          | GB-DP | PeakPad |
|-----------------|---------|---------|--------------|-------------|---------|----------|---------|-------------|---------------------|-------|---------|
| Parameter       | $n_c$   | $n_c/k$ | $Eps/MinPts$ | $bandwidth$ | $n_c/p$ | $n_c/p$  | $n_c/k$ | $s/p/l$     | $k/\epsilon/MinPts$ | $n_c$ | $n_c/k$ |
| Agg             | 7       | 7/15    | 0.0050/8     | 0.2100      | 7/2     | 7/2      | 7/20    | 788/50/20   | 11/0.028/4          | 7     | 7/28    |
| Flame           | 2       | 2/10    | 0.0070/6     | 0.2500      | 2/3     | 2/2      | 2/5     | 240/50/20   | 3/0.3/1             | 2     | 2/15    |
| Jain            | 2       | 2/5     | 0.0800/23    | 0.3300      | 2/2     | 2/2      | 2/11    | 373/10/30   | 15/0.028/1          | 2     | 2/19    |
| Threecircles    | 3       | 3/15    | 0.0080/3     | 0.2150      | 3/1     | 3/2      | 3/6     | 299/10/20   | 15/0.028/1          | 3     | 3/17    |
| Sticks          | 4       | 4/9     | 0.0090/3     | 0.2600      | 4/2     | 4/2      | 4/11    | 512/50/20   | 11/0.028/4          | 4     | 4/15    |
| Compound        | 6       | 6/10    | 0.0090/2     | 0.1600      | 5/0.5   | 5/2      | 5/5     | 399/10/10   | 11/0.025/4          | 5     | 5/6     |
| R15             | 15      | 15/10   | 0.0500/27    | 0.0700      | 15/2    | 15/2     | 15/15   | 600/50/20   | 11/0.01/4           | 15    | 15/24   |
| Spiral          | 3       | 3/5     | 0.0900/2     | 0.1300      | 3/2     | 3/2      | 3/5     | 312/50/20   | 11/0.01/1           | 3     | 3/18    |
| T48K            | 6       | 6/15    | 0.0270/28    | 0.2450      | 6/2     | 6/2      | 6/20    | 1000/150/10 | 11/0.028/4          | 6     | 6/60    |
| S1              | 15      | 15/15   | 0.0020/2     | 0.0500      | 15/2    | 15/2     | 15/20   | 1000/50/20  | 11/0.028/4          | 15    | 15/71   |
| S3              | 15      | 15/20   | 0.0300/60    | 0.0700      | 15/2    | 15/2     | 15/35   | 1000/15/10  | 11/0.01/2           | 15    | 15/50   |
| A2              | 35      | 35/15   | 0.0011/3     | 0.0500      | 35/1    | 35/1     | 35/43   | 1000/50/20  | 9/0.02/4            | 35    | 35/30   |
| A3              | 50      | 50/10   | 0.0090/2     | 0.0500      | 50/1    | 50/1     | 50/50   | 1000/50/10  | 9/0.028/4           | 50    | 50/50   |
| Birchrg1        | 100     | 100/10  | 0.009/31     | 0.1000      | 100/0.5 | 100/0.5  | 100/10  | 5000/500/20 | 11/0.01/5           | 100   | 100/50  |
| TB10w           | 2       | 2/11    | 0.01/17      | 0.2000      | 2/0.5   | 2/1      | 2/7     | 5000/500/20 | 10/0.028/9          | 2     | 2/50    |
| Iris            | 3       | 3/15    | 0.1300/2     | 0.2400      | 3/2     | 3/3      | 3/15    | 150/50/20   | 11/0.01/5           | 3     | 3/15    |
| Wine            | 3       | 3/30    | 0.5000/2     | 0.6000      | 3/2     | 3/2      | 3/18    | 178/50/20   | 5/0.1/4             | 3     | 3/20    |
| Segment         | 7       | 7/30    | 0.3000/2     | 0.6000      | 7/2     | 7/1      | 7/7     | 1000/50/20  | 29/0.025/4          | 8     | 8/46    |
| Drivepoints     | 4       | 4/45    | 0.4000/11    | 0.4000      | 4/3     | 4/3      | 4/50    | 606/50/20   | 35/0.27/4           | 3     | 3/60    |
| Breastcancer    | 2       | 2/5     | 0.4000/21    | 0.5000      | 2/1     | 2/1      | 2/12    | 569/50/20   | 25/0.018/4          | 2     | 2/11    |
| Waveform        | 3       | 3/15    | 0.4000/9     | 0.6000      | 3/2     | 3/2      | 3/7     | 1000/50/20  | 7/0.02/4            | 4     | 4/71    |
| Movementslibras | 15      | 15/10   | 0.9000/8     | 1.2000      | 15/2    | 15/2     | 15/11   | 65/50/20    | 5/0.001/4           | 15    | 15/9    |
| Parkin          | 2       | 2/10    | 0.1000/5     | 0.5000      | 2/2     | 2/2      | 2/5     | 1000/50/20  | 5/0.015/4           | 2     | 2/21    |
| Vote            | 2       | 2/15    | 0.7000/23    | 1.2000      | 2/2     | 2/2      | 2/11    | 435/50/20   | 11/0.1/4            | 2     | 2/35    |
| YTF             | 41      | 41/30   | 0.1000/7     | 0.3000      | 41/2    | 41/2     | 41/30   | 1000/50/20  | 11/0.3/4            | 41    | 41/55   |
| USPS            | 10      | 10/20   | 0.1000/2     | 0.3500      | 10/0.5  | 10/0.5   | 10/20   | 1000/50/20  | 11/0.01/4           | 11    | 11/45   |
| MNIST           | 10      | 10/20   | 2.1000/15    | 3.0000      | 10/1    | 10/1     | 10/20   | 3746/50/20  | 11/0.01/5           | 10    | 10/20   |
| OlivettiFaces   | 40      | 40/10   | 0.7000/2     | 0.7000      | 40/0.2  | 40/0.3   | 40/6    | 400/50/20   | 4/0.0005/4          | 40    | 40/5    |

top largest  $\theta$  values. The test results are reported in Table V, with the best results highlighted. As shown, PeakPad gets the highest scores on all datasets (except for *Segment*), verifying the effectiveness of PeakPad in center detection

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN}. \quad (26)$$

Fig. 11 shows the clustering results of DPC and PeakPad by selecting forty centers on the *OlivettiFaces* [42] dataset, where cluster centers are marked in white circles. As Fig. 11 shows, in terms of clustering effect, PeakPad is superior; and in terms of center detection, DPC incorrectly identified nine centers (marked by white boxes) with  $F1 = 0.78$ , while PeakPad only misidentified four centers with  $F1 = 0.90$ .

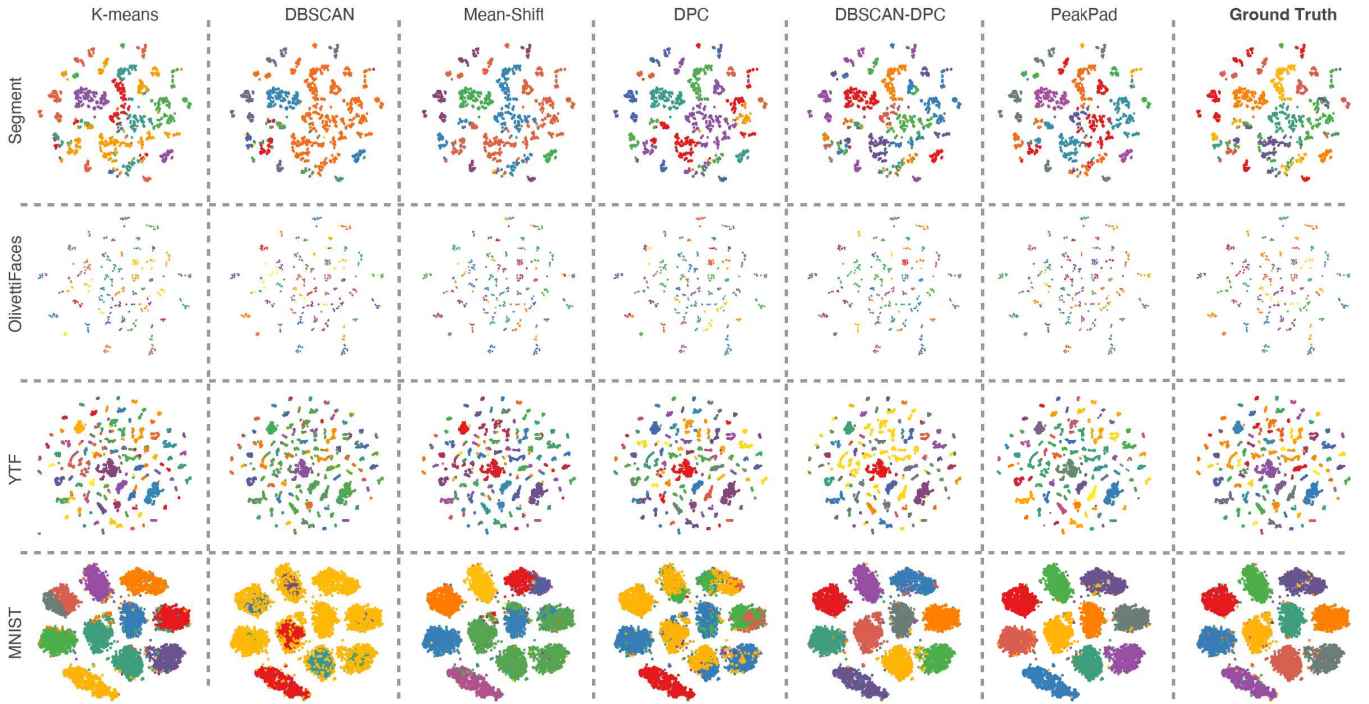


Fig. 10. t-SNE-based visualization comparison of different algorithms on real-world datasets.

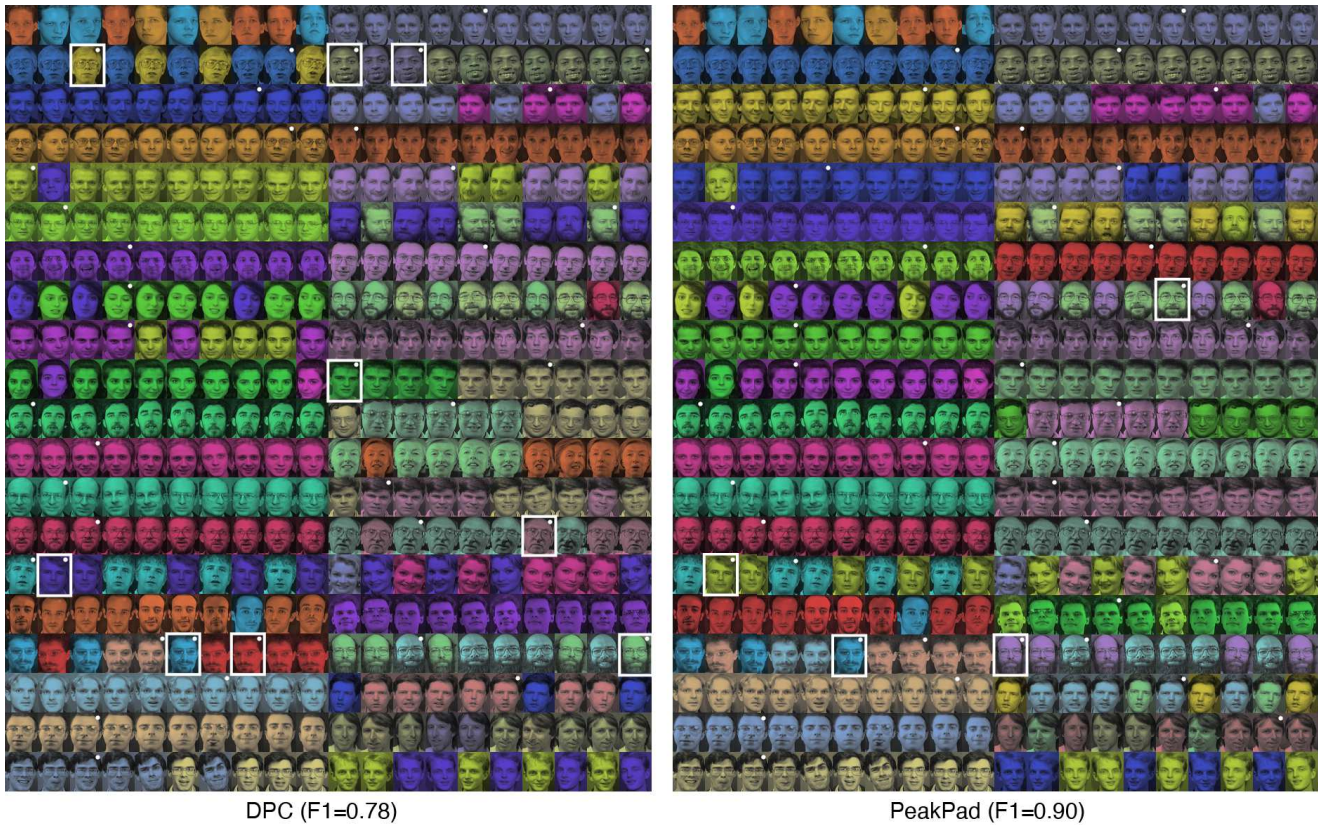


Fig. 11. Clustering results of DPC and PeakPad on the *OlivettiFaces* dataset.

To further validate the effectiveness of PeakPad, we conducted tests on 160 synthetic datasets with varying geometric structures and dimensions, including 80 Gaussian sphere datasets and 80 ellipsoidal cluster datasets [54]. These were designed to rigorously evaluate the robustness of the proposed

$\theta$ -based center detection mechanism. All algorithms were tested with fixed, well-tuned parameters. As shown in Fig. 12, PeakPad consistently outperforms other methods in both center detection and overall clustering accuracy, achieving average  $F1$ -scores of 0.9588 and 0.8788 on Gaussian and ellipsoidal

TABLE V  
COMPARISON OF CENTER DETECTION PERFORMANCE ( $F1$ -SCORE)

| Algorithm | Agg         | Flame       | Jain        | Threecircles | Compound    | R15         | Spiral      | Iris        | Wine        | Segment     | Drivepoints | Breastcancer | Waveform    | YTF         | USPS        | MNIST       | OlivettiFaces |
|-----------|-------------|-------------|-------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|--------------|-------------|-------------|-------------|-------------|---------------|
| DPC       | <b>1.00</b> | <b>1.00</b> | 0.50        | 0.67         | 0.67        | <b>1.00</b> | <b>1.00</b> | 0.67        | <b>1.00</b> | 0.71        | <b>0.75</b> | 0.50         | <b>0.67</b> | 0.54        | 0.60        | 0.50        | 0.78          |
| SNN-DPC   | <b>1.00</b> | <b>1.00</b> | 0.50        | 0.67         | <b>0.83</b> | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>0.86</b> | <b>0.75</b> | <b>1.00</b>  | <b>0.67</b> | 0.51        | 0.40        | 0.50        | 0.78          |
| DPC-CE    | <b>1.00</b> | <b>1.00</b> | 0.50        | 0.67         | <b>0.83</b> | <b>1.00</b> | <b>1.00</b> | 0.67        | <b>1.00</b> | 0.71        | 0.50        | 0.50         | <b>0.67</b> | 0.54        | 0.60        | 0.70        | 0.75          |
| PeakPad   | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>1.00</b>  | <b>0.83</b> | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | <b>1.00</b> | 0.71        | <b>0.75</b> | <b>1.00</b>  | <b>0.67</b> | <b>0.56</b> | <b>0.80</b> | <b>1.00</b> | <b>0.90</b>   |

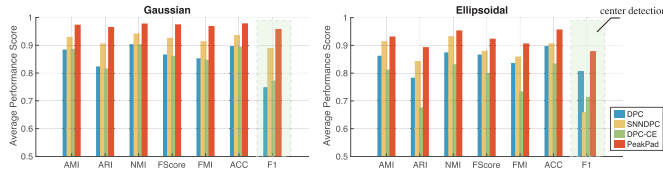


Fig. 12. Comparative performance of DPC-based algorithms and PeakPad on 160 synthetic datasets (80 Gaussian spheres and 80 ellipsoids).

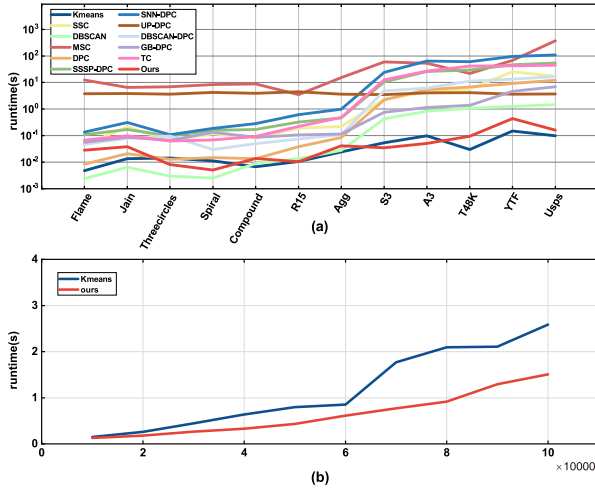


Fig. 13. (a) Runtime comparison of different algorithms and (b) runtime comparison between  $K$ -means and ours on ten different size sampling datasets of the *Birch91* dataset.

datasets, respectively. It also ranks highest across all seven evaluation metrics (e.g., AMI, ARI, and NMI), demonstrating the geometric adaptability of the proposed  $\theta$ -based center detection mechanism.

As verified, PeakPad owns a more robust center detection performance than other DPC-based algorithms, which is also sound evidence of the effectiveness and robustness of our new center assumption.

#### E. Speed of PeakPad

Execution speed is also an important criterion for evaluating an algorithm's performance in large-scale data clustering tasks. Fig. 13(a) presents the runtime comparison of all comparison algorithms on different datasets. As shown, PeakPad and  $K$ -means outperform other algorithms, but is much faster than DPC and SNN-DPC (an improved version of DPC), especially for large-scale datasets, such as *YTF* and *USPS*.

To further verify the fast speed of PeakPad, we conducted speed comparisons between PeakPad and  $K$ -means using the *Birch91* dataset which consisted of 100 000 points, as in Fig. 13(b). As shown, PeakPad is faster than  $K$ -means. When handling 100 000 data points, PeakPad takes only 1.5 s,

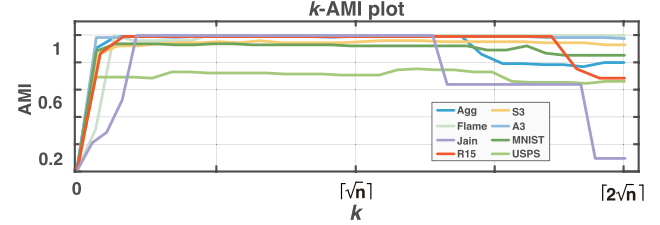


Fig. 14.  $k$ -AMI plot of the PeakPad algorithm on different datasets with  $k \in [0, 2\lceil n^{1/2} \rceil]$ .

which is 1 s faster than  $K$ -means. Moreover, PeakPad also outperforms  $K$ -means in terms of accuracy (see Table II).

As verified, PeakPad with fast speed is promising for large-scale data clustering.

#### F. Parameter Insensitivity

The main parameter  $k$  (i.e., the number of neighbors) in PeakPad is set  $k = \lceil n^{1/2} \rceil$  as default. Parameter  $k$  is needed in our kNN-based density estimation, density peak detection, and peak graph building. Thus, the parameter insensitivity of  $k$  deserves a discussion.

Fig. 14 presents the  $k$ -AMI plot of several tested datasets, with  $k \in [0, 2\lceil n^{1/2} \rceil]$  and a given correct cluster number as inputs. As Fig. 14 shows, within the range of  $[0, 2\lceil n^{1/2} \rceil]$ , PeakPad obtains a stable optimal performance over a wide range around  $k = \lceil n^{1/2} \rceil$ , which verifies the effectiveness of  $k = \lceil n^{1/2} \rceil$  setting and the insensitivity of PeakPad to parameter  $k$ .

Except for parameter  $k$ , PeakPad also needs  $n_c$  (cluster number) and  $k_b$  (border point detection parameter).  $n_c$  is selected by observing our decision graph, just like in DPC, and  $k_b$  is a fixed setting as in (2).

#### V. CONCLUSION

The proposed Peak-Padding Clustering (PeakPad) algorithm addresses the challenge of clustering complex-shaped clusters by decomposing them into high-association single-peak clusters. This transformation simplifies the clustering task from assigning data points to clusters to associating density peaks. By introducing the concept of DC distance, PeakPad converts arbitrary-shaped density surface paths into standardized DC surface paths, enabling more accurate evaluation of associations among single-peak clusters.

PeakPad identifies cluster centers as density peaks with relatively high minimum padding costs in the peak graph and assigns noncenter peaks (representing single-peak clusters) to their corresponding complex-shaped clusters. This approach yields accurate complex shapes clustering and robust center detection. Furthermore, due to its reliance on kNN-based distances, PeakPad is efficient and scalable to large datasets. Its effectiveness has been validated on both synthetic and real-world datasets, demonstrating strong performance in clustering and center detection.

However, PeakPad's clustering performance depends heavily on the quality of single-peak cluster identification, which in turn relies on accurate density estimation. Currently, a simple kNN-based estimator is used for efficiency, but there is room for improvement, especially in high-dimensional or noisy data scenarios [22], [51], [52]. Future work could incorporate nonlinear dimensionality reduction and feature selection techniques to enhance the accuracy and robustness of density estimation. Adaptive imputation strategies, such as evolving fuzzy systems, may also help enable real-time, uncertainty-aware clustering.

Although PeakPad shows strong performance in center detection, it still requires manual selection of cluster centers. As automatic center detection is critical in many real-world applications, future enhancements will focus on integrating Bayesian nonparametric models [53] and dynamic update mechanisms to enable fully automatic cluster center identification.

## REFERENCES

- [1] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010, doi: [10.1016/j.patrec.2009.09.011](https://doi.org/10.1016/j.patrec.2009.09.011).
- [2] M. Gao and G.-Y. Shi, "Ship-handling behavior pattern recognition using AIS sub-trajectory clustering analysis based on the T-SNE and spectral clustering algorithms," *Ocean Eng.*, vol. 205, Jun. 2020, Art. no. 106919, doi: [10.1016/j.oceaneng.2020.106919](https://doi.org/10.1016/j.oceaneng.2020.106919).
- [3] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 132–149.
- [4] T. Lei, P. Liu, X. Jia, X. Zhang, H. Meng, and A. K. Nandi, "Automatic fuzzy clustering framework for image segmentation," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 9, pp. 2078–2092, Sep. 2020, doi: [10.1109/TFUZZ.2019.2930030](https://doi.org/10.1109/TFUZZ.2019.2930030).
- [5] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015, doi: [10.1126/science.aaa8415](https://doi.org/10.1126/science.aaa8415).
- [6] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Inf. Theory*, vol. IT-21, no. 1, pp. 32–40, Jan. 1975.
- [7] H. Averbuch-Elor, N. Bar, and D. Cohen-Or, "Border-peeling clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 7, pp. 1791–1797, Jul. 2020, doi: [10.1109/TPAMI.2019.2924953](https://doi.org/10.1109/TPAMI.2019.2924953).
- [8] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014, doi: [10.1126/science.1242072](https://doi.org/10.1126/science.1242072).
- [9] R. Liu, H. Wang, and X. Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *Inf. Sci.*, vol. 450, pp. 200–226, Jun. 2018, doi: [10.1016/j.ins.2018.03.031](https://doi.org/10.1016/j.ins.2018.03.031).
- [10] J. Guan, S. Li, X. He, and J. Chen, "Clustering by fast detection of main density peaks within a peak digraph," *Inf. Sci.*, vol. 628, pp. 504–521, May 2023, doi: [10.1016/j.ins.2023.01.144](https://doi.org/10.1016/j.ins.2023.01.144).
- [11] D. U. Pizzagalli, S. F. Gonzalez, and R. Krause, "A trainable clustering algorithm based on shortest paths from density peaks," *Sci. Adv.*, vol. 5, no. 10, Oct. 2019, Art. no. eaax3770, doi: [10.1126/sciadv.aax3770](https://doi.org/10.1126/sciadv.aax3770).
- [12] W. Guo, W. Wang, S. Zhao, Y. Niu, Z. Zhang, and X. Liu, "Density peak clustering with connectivity estimation," *Knowl.-Based Syst.*, vol. 243, May 2022, Art. no. 108501, doi: [10.1016/j.knosys.2022.108501](https://doi.org/10.1016/j.knosys.2022.108501).
- [13] J. Guan, S. Li, X. He, and J. Chen, "Peak-graph-based fast density peak clustering for image segmentation," *IEEE Signal Process. Lett.*, vol. 28, pp. 897–901, 2021, doi: [10.1109/LSP.2021.3072794](https://doi.org/10.1109/LSP.2021.3072794).
- [14] M. Du, S. Ding, and H. Jia, "Study on density peaks clustering based on K-nearest neighbors and principal component analysis," *Knowl.-Based Syst.*, vol. 99, pp. 135–145, May 2016, doi: [10.1016/j.knosys.2016.02.001](https://doi.org/10.1016/j.knosys.2016.02.001).
- [15] Y. Yang, J. Cai, H. Yang, and X. Zhao, "Density clustering with divergence distance and automatic center selection," *Inf. Sci.*, vol. 596, pp. 414–438, Jun. 2022, doi: [10.1016/j.ins.2022.03.027](https://doi.org/10.1016/j.ins.2022.03.027).
- [16] K. K. Sharma, A. Seal, A. Yazidi, and O. Krejcar, "A new adaptive mixture distance-based improved density peaks clustering for gearbox fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–16, 2022, doi: [10.1109/TIM.2022.3216366](https://doi.org/10.1109/TIM.2022.3216366).
- [17] L. Ma et al., "UP-DPC: Ultra-scalable parallel density peak clustering," *Inf. Sci.*, vol. 660, Mar. 2024, Art. no. 120114, doi: [10.1016/j.ins.2024.120114](https://doi.org/10.1016/j.ins.2024.120114).
- [18] J. Guan, S. Li, J. Zhu, X. He, and J. Chen, "Fast main density peak clustering within relevant regions via a robust decision graph," *Pattern Recognit.*, vol. 152, Aug. 2024, Art. no. 110458, doi: [10.1016/j.patcog.2024.110458](https://doi.org/10.1016/j.patcog.2024.110458).
- [19] D. Cheng, Y. Li, S. Xia, G. Wang, J. Huang, and S. Zhang, "A fast granular-ball-based density peaks clustering algorithm for large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 17202–17215, Dec. 2024, doi: [10.1109/TNNLS.2023.3300916](https://doi.org/10.1109/TNNLS.2023.3300916).
- [20] S. Chakraborty and S. Das, "K-means clustering with a new divergence-based distance metric: Convergence and performance analysis," *Pattern Recognit. Lett.*, vol. 100, pp. 67–73, Dec. 2017, doi: [10.1016/j.patrec.2017.09.025](https://doi.org/10.1016/j.patrec.2017.09.025).
- [21] K. K. Sharma and A. Seal, "Spectral embedded generalized mean based K-nearest neighbors clustering with S-distance," *Expert Syst. Appl.*, vol. 169, May 2021, Art. no. 114326, doi: [10.1016/j.eswa.2020.114326](https://doi.org/10.1016/j.eswa.2020.114326).
- [22] I. H. Jarman, T. A. Etchells, D. Bacciu, J. M. Garibaldi, I. O. Ellis, and P. J. G. Lisboa, "Clustering of protein expression data: A benchmark of statistical and neural approaches," *Soft Comput.*, vol. 15, no. 8, pp. 1459–1469, Aug. 2011.
- [23] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Trans. Comput.*, vol. C-20, no. 1, pp. 68–86, Jan. 1971, doi: [10.1109/T-C.1971.223083](https://doi.org/10.1109/T-C.1971.223083).
- [24] J. Guan, S. Li, X. He, J. Zhu, and J. Chen, "Fast hierarchical clustering of local density peaks via an association degree transfer method," *Neurocomputing*, vol. 455, pp. 401–418, Sep. 2021, doi: [10.1016/j.neucom.2021.05.071](https://doi.org/10.1016/j.neucom.2021.05.071).
- [25] J. Guan, S. Li, X. Chen, X. He, and J. Chen, "DEMOS: Clustering by pruning a density-boosting cluster tree of density mounts," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 10, pp. 10814–10830, Oct. 2023, doi: [10.1109/TKDE.2023.3266451](https://doi.org/10.1109/TKDE.2023.3266451).
- [26] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *KDD*, vol. 96, no. 34, pp. 226–231, 1996.
- [27] K. Sawant, "Adaptive methods for determining DBSCAN parameters," *Int. J. Innov. Sci.*, pp. 329–334, 2014.
- [28] A. Bechini, F. Marcelloni, and A. Renda, "TSF-DBSCAN: A novel fuzzy density-based approach for clustering unbounded data streams," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 3, pp. 623–637, Mar. 2022, doi: [10.1109/TFUZZ.2020.3042645](https://doi.org/10.1109/TFUZZ.2020.3042645).
- [29] D. Cheng, J. Huang, S. Zhang, S. Xia, G. Wang, and J. Xie, "K-means clustering with natural density peaks for discovering arbitrary-shaped clusters," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 8, pp. 11077–11090, Aug. 2024, doi: [10.1109/TNNLS.2023.3248064](https://doi.org/10.1109/TNNLS.2023.3248064).
- [30] T. M. Apostol, *Calculus, Volume 1*. Hoboken, NJ, USA: Wiley, 1991.
- [31] N. Bhatia and Vandana, "Survey of nearest neighbor techniques," 2010, *arXiv:1007.0085*.
- [32] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, Sep. 1977.
- [33] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. NIPS*, 2004, pp. 1601–1608.
- [34] J. Hou, H. Lin, H. Yuan, and M. Pelillo, "Flexible density peak clustering for real-world data," *Pattern Recognit.*, vol. 156, Dec. 2024, Art. no. 110772, doi: [10.1016/j.patcog.2024.110772](https://doi.org/10.1016/j.patcog.2024.110772).
- [35] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems," *Pattern Recognit.*, vol. 39, no. 5, pp. 761–775, May 2006.
- [36] P. Fränti and S. Sieranoja, "K-means properties on six clustering benchmark datasets," *Appl. Intell.*, vol. 48, no. 12, pp. 4743–4759, Dec. 2018. [Online]. Available: <http://cs.uef.fi/sipu/datasets/>
- [37] T. Qiu and Y.-J. Li, "Fast LDP-MST: An efficient density-peak-based clustering method for large-size datasets," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4767–4780, May 2023.
- [38] K. Bache and M. Lichma. (2013). *UCI Machine Learning Repository*. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [39] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. CVPR*, Jun. 2011, pp. 529–534.
- [40] T. Hastie, J. Friedman and R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2009.
- [41] Y. LeCun, C. Cortes, and C. Burges. (2010). *MNIST Handwritten Digit Database*. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>

- [42] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Jan. 1994, pp. 138–142.
- [43] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, no. 95, pp. 2837–2854, 2010.
- [44] P. A. Estevez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 189–201, Feb. 2009.
- [45] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *J. Amer. Stat. Assoc.*, vol. 78, no. 383, pp. 553–569, Sep. 1983.
- [46] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation," in *Proc. Australas. Joint Conf. Artif. Intell.*, 2006, pp. 1015–1021.
- [47] J. MacQueen et al., "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, Oakland, CA, USA, vol. 1, 1967, pp. 281–297.
- [48] A. Patil, D. Huard, and C. Fonnesbeck, "PyMC: Bayesian stochastic modelling in Python," *J. Stat. Softw.*, vol. 35, no. 4, pp. 1–81, 2010.
- [49] L. V. D. Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.
- [50] J. Yang and C.-T. Lin, "Autonomous clustering by fast find of mass and distance peaks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 7, pp. 5336–5349, Jul. 2025.
- [51] G. Duka, S. Travin, I. Zinicovscaia, and R. E. Precup, "Approach to evaluate the data of moss biomonitoring studies: Preprocessing and preliminary ranking," *Romanian J. Inf. Sci. Technol.*, vol. 2023, nos. 3–4, pp. 278–288, Sep. 2023.
- [52] Ü. Kılıç, E. S. Eşsiz, and M. K. Keleş, "Binary anarchic society optimization for feature selection," *Romanian J. Inf. Sci. Technol.*, vol. 2023, nos. 3–4, pp. 351–364, 2023.
- [53] D. Bacciu and D. Castellana, "Bayesian mixtures of hidden tree Markov models for structured data clustering," *Neurocomputing*, vol. 342, pp. 49–59, May 2019.
- [54] L. H. D. S. Fernandes, A. C. Lorena, and K. Smith-Miles, "Towards understanding clustering problems and algorithms: An instance space analysis," *Algorithms*, vol. 14, no. 3, p. 95, Mar. 2021.



**Junyi Guan** (Member, IEEE) received the Ph.D. degree from Zhejiang University of Technology (ZJUT), Hangzhou, China, in January 2022.

He held a post-doctoral position at ZJUT from March 2022 to September 2024. Since September 2024, he has been with Hangzhou Normal University (HZNU), Hangzhou, as an Associate Professor. His current research interests include data mining, pattern recognition, unsupervised learning, and machine learning.



**Bingbing Jiang** (Member, IEEE) received the B.Sc. degree from Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China, in 2014, and the Ph.D. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2019.

He is currently an Associate Professor with Hangzhou Normal University (HZNU), Hangzhou, China. He has published more than 50 technical papers at prestigious international journals and conferences in these fields, such as IEEE TRANS-

ACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNS), IEEE TRANSACTIONS ON CYBERNETICS (TCYB), IEEE TRANSACTIONS ON MULTIMEDIA (TMM), IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS (TII), IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS), IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE (TETCI), ACM TKDD, Information Fusion, Pattern Recognition, AAAI, IJCAI, ACM MM, and ACM CIKM. His research interests include semisupervised learning, multimodel/view fusion and learning, feature selection, and clustering.

Dr. Jiang serves as an Editorial Board Member for *Applied Soft Computing*, and an Associate Editor for *Neurocomputing*.



**Weiguo Sheng** (Member, IEEE) received the M.Sc. degree in information technology from the University of Nottingham, Nottingham, U.K., in 2002, and the Ph.D. degree in computer science from Brunel University London, Uxbridge, U.K., in 2005.

He worked as a Researcher at the University of Kent, Canterbury, U.K., and Royal Holloway, University of London, London, U.K. He is currently a Professor at Hangzhou Normal University (HZNU), Hangzhou, China. His research interests include evolutionary computation, data mining/clustering, pattern recognition, and machine learning.



**Yangyang Zhao** is currently pursuing the Ph.D. degree with the College of Information Engineering, Zhejiang University of Technology (ZJUT), Hangzhou, China.

Her current research interests include data mining, pattern recognition, and machine learning.



**Sheng Li** (Senior Member, IEEE) received the bachelor's degree from Zhejiang University of Technology (ZJUT), Hangzhou, China, in 2006, and the M.Sc. degree in communications engineering and the Ph.D. degree in electronic engineering from the University of York, York, U.K., in 2007 and 2010, respectively.

From November 2010 to October 2011, he was a Post-Doctoral Researcher with the Ilmenau University of Technology, Ilmenau, Germany. Since April 2012, he has been with ZJUT, where he is currently an Associate Professor. His research interests include signal processing, machine learning, and pattern recognition.

Dr. Li received the K. M. Stott Prize for Excellence in Scientific Research in 2010. He received the Best Paper Award from the VTC 2011 spring for the track signal processing for wireless communication.



**Xiongxiang He** received the M.S. degree from Qufu Normal University, Qufu, China, in 1994, and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 1997.

He held a post-doctoral position with Harbin Institute of Technology, Harbin, China, from 1998 to 2000. He joined Zhejiang University of Technology (ZJUT), Hangzhou, in 2001, where he has been a Professor with the College of Information Engineering. His research areas include nonlinear control, signal processing, and pattern recognition.