Research on Clustering Algorithm Based on K-DPC

Weiguo Yi*, Wenyue Zhang, Haonan Hu, Shaohua Chen

***Abstract*—Density Peak Clustering (DPC) is a density-based clustering algorithm that exhibits superior performance through its innovative cut-off distance concept. It effectively adapts to clusters of arbitrary shapes and sizes, free from geometric constraints. The fundamental principle involves identifying cluster centers by assessing the local density and relative distance of data points, followed by subsequent point allocation. However, DPC has inherent limitations: in datasets with substantial inter-cluster density variations, local density calculations lack precision, resulting in incorrect selection of cluster centers. Additionally, the point assignment strategy may trigger a "domino effect," influencing the allocation of adjacent points. To address these challenges, this paper introduces K-DPC, a hybrid clustering algorithm that integrates K-nearest neighbor (K-NN) and an enhanced point assignment strategy. K-DPC utilizes the average distance of K-NN and its N-times standard deviation for noise point identification and filtering. By redefining local density using K-mutual nearest neighbors and K-NN, it improves the accuracy of initial cluster center identification. Furthermore, the concept of high-density points is introduced, leveraging the direct density reachability of K-NN points for point assignment. For non-dense data, secondary assignment is performed via cumulative weight calculation. Experimental results on diverse datasets indicate that K-DPC surpasses state-of-the-art clustering methods in terms of accuracy. Notably, it demonstrates robust noise resistance and reduces sensitivity to the domino effect. The proposed method highlights its potential as an effective clustering solution for complex datasets.**

***Index Terms*—Cluster analysis, k-nearest neighbor, density peak clustering, k-mean distance, standard deviation.**

I. Introduction

CLUSTERING, as an unsupervised learning method [1], partitions datasets into groups (clusters) where intra-cluster similarity is maximized and inter-cluster dissimilarity is emphasized. As fundamental tools, clustering techniques facilitate data preprocessing and exploratory analysis, enabling efficient pattern discovery.These algorithms have been widely applied across diverse domains, including image processing [2], [3], document classification [4], intelligent transportation systems [5], and smart grid optimization [6], [7]. Clustering methodologies are typically classified into four main categories: partition-based, hierarchical, grid-based, and density-based approaches [8]. Partition-based methods primarily comprise centroid-based techniques such as K-means [9] and K-medoids [10], while hierarchical clustering is commonly implemented through algorithms like CURE [11]. Density-based approaches are dominated by three seminal methods: DBSCAN for spatial data clustering [12], Density Peak Clustering (DPC) for multi-center detection [13], and Mean-Shift for mode-seeking [14]. For grid-based clustering, the STING algorithm [15] remains a benchmark solution for spatial data partitioning. The DPC algorithm is widely acknowledged for its simplicity, efficiency, avoidance of iterative objective function optimization, and capability to identify multi-shaped clusters. It exhibits robust performance in detecting cluster structures within complex datasets. However, a critical limitation resides in its exclusive dependence on local sample distribution to define local density. This approach may inaccurately characterize the actual density of sample points in datasets with substantial inter-cluster density variations, leading to misselection of cluster centers. Additionally, its assignment strategy is vulnerable to the "domino effect," whereby a single misassigned point can propagate errors cascadingly. A misassigned point not only distorts a cluster's density estimation but also disrupts cluster center establishment. Given that the DPC algorithm's allocation mechanism relies heavily on local density and distance metrics, such misassignments can trigger a cascading effect on the attribution of neighboring samples, potentially causing widespread misclassification. To address these limitations, researchers have proposed various improvement strategies in recent years. For example, the DPC-KNN-PCA algorithm proposed by Du et al. [16] integrates the k-nearest neighbor concept but exhibits high time complexity. Xie et al. [17] developed the FKNN-DPC algorithm, which estimates local density using the exponential kernel of distances to k-nearest neighbors, thereby enhancing clustering performance on multi-dimensional datasets. However, this approach demonstrates suboptimal performance on complex datasets and fails to resolve issues arising from heterogeneous data density. The adaptive DPC algorithm proposed by Liu et al. [18], which leverages k-nearest neighbors and a merging strategy, mitigates the impact of truncation distance on cluster center selection but underperforms on datasets with substantial density variations. Liu et al. [19] also introduced a similarity-based method featuring a two-stage point assignment strategy, which improves cluster center identification but remains sensitive to noise and incurs high computational costs. Bai et al. [20] integrated the DPC algorithm with K-means to reduce computational complexity, yet this approach inherits K-means' limitations, such as susceptibility to local optima. Jiang et al. [21] enhanced cut-off distance determination by incorporating the Gini coefficient and using k-nearest neigh-

Manuscript received April 23, 2025; revised August 7, 2025.

This work was supported by the Educational Department of Liaoning Province (No. JYTMS20230012) and the Liaoning Provincial Department of Transportation Science and Technology Program (No.2024-353-5).

W. G. Yi is an associate professor at the School of Railway Intelligent Engineering, Dalian Jiaotong University, Dalian, LiaoNing, 116021, China (corresponding author; e-mail: jiekexun98@163.com).

W. Y. Zhang is a postgraduate student at the School of Railway Intelligent Engineering, Dalian Jiaotong University, Dalian, LiaoNing, 116021, China (e-mail: 2263803570@qq.com).

H. N. Hu is a postgraduate student at the School of Materials Science and Engineering, Dalian Jiaotong University, Dalian, LiaoNing, 116021, China (e-mail: hu95haonan@163.com).

S. H. Chen is a professor at the School of Railway Intelligent Engineering, Dalian Jiaotong University, Dalian, LiaoNing, 116021, China (e-mail: chengshineng@163.com).

bors for center point identification, improving performance on datasets with large density differences. Nevertheless, its accuracy in high-dimensional spaces remains constrained, and it occasionally fails to identify cluster centers accurately. Yuan et al. [22] developed a k-nearest neighbor density peak clustering algorithm with an adaptive merging strategy, refining the k-nearest neighbor approach to improve performance on complex datasets. Chen et al. [23] proposed the DPC-NNO algorithm, which combines inverse nearest neighbors and k-nearest neighbors to define local density, effectively addressing the initial point assignment problem in datasets with heterogeneous densities. Zhou et al. [24] utilized mutual k-nearest neighbors and local kernel density to enhance local density estimation, addressing the initial point assignment issue; however, it remains sensitive to outliers, potentially degrading clustering accuracy.Overall, existing algorithms such as DPC-KNN-PCA and FKNN-DPC confront challenges including sensitivity to density disparities, truncation distance dependency, noise/outlier vulnerability, high computational complexity, and susceptibility to local optima. To address these limitations, this study proposes enhancements comprising robust noise/outlier management, optimized local density estimation, computational complexity reduction, local optima avoidance, and integration of high-density point classification. These improvements collectively enhance both the clustering performance and algorithmic robustness.

To overcome the limitations of the DPC algorithm and its current improved versions, this paper puts forward an enhanced K-DPC algorithm. This algorithm aims to tackle the difficulties in determining cluster centers and non - center points, and to alleviate the domino effect resulting from point assignment. The key innovations of the K - DPC algorithm are summarized as follows:

1) Noise Point Preprocessing: This study proposes a density-threshold filtering strategy to effectively identify and remove low-density noise points during the preprocessing stage.

2) Cluster Center Optimization: This study proposes a novel local density reconstruction method that integrates k-mutual nearest neighbors (kMNN) and k-nearest neighbors (kNN). By applying dual-neighborhood constraints, this approach alleviates computational biases stemming from inter-class density disparities. Consequently, it improves the discriminability of cluster centers in low-density regions and minimizes misjudgments arising from ambiguous local density estimations.

3) Non-central Point Assignment: The proposed algorithm incorporates high-density points and direct density reachability based on k-nearest neighbors to facilitate the initial classification. For low-density points, a Gaussian kernel weighting model is used to calculate density-sensitive aggregation weights. This approach effectively resolves the ambiguity in point assignment within low - density regions and significantly improves overall clustering performance.

Section II introduces the DPC algorithm and analyzes its limitations. Section III details the proposed K-DPC algorithm. Section IV evaluates the performance of the improved K-DPC algorithm against traditional methods using diverse datasets. Finally, Section V concludes the paper.

II. Introduction to the Principle of Related Algorithms

A. DPC algorithm

The Density Peaks Clustering (DPC) algorithm is a classic density-based clustering method predominantly applied to spatial data analysis. It is grounded in two core concepts: local density and relative distance. Local density denotes the count of neighboring points surrounding a given point, typically computed using a Gaussian kernel function. Relative distance is defined as the distance from a point to its nearest neighbor with a higher local density; for the point with the highest local density, its relative distance is calculated as the distance to the nearest point with the second-highest local density.

The traditional DPC has two classical methods for the calculation of the local density, the truncated kernel method and the Gaussian kernel method, assuming that the data set $X = \{x1, x2, x3...xn\}$, for $\forall \mathbf{x_i}, \mathbf{x_j} \in \mathbf{X}$, two different local density calculation formulas are as follows.

Truncation kernel:

$$\rho_i = \chi\left[d\left(x_i, x_j - d_c\right)\right], \chi\left(x\right) = \begin{cases} 1 & x < 0 \\ 0 & x \geq 0 \end{cases} \quad (1)$$

Gaussian kernel:

$$\rho_i = \sum_{x_j \in X} exp\left[-\left(\frac{d\left(x_i - x_j\right)}{d_c}\right)^2\right] \quad (2)$$

Where χ is the indicator function, which is 1 if the condition is satisfied and 0 otherwise. $\mathbf{d_{ij}}$ is the Euclidean distance between points $\mathbf{x_i}$ and $\mathbf{x_j}$, $\mathbf{d_c}$is the truncation distance, and $\rho_\mathbf{i}$ is the local density of point i.

The relative distance is for each point i, traverse all its neighbors to find the nearest neighbor j whose local density is higher than $\rho_\mathbf{i}$, and then calculate the distance $\mathbf{d_{ij}}$ between i and j

Relative distance:

$$\delta_i = min_{j:\rho_j > \rho_i} d_{ij} \quad (3)$$

If the local density ρ_i of $\mathbf{x_i}$ is the largest in the dataset X, the relative distance δ_i of $\mathbf{x_i}$ is as follows.

$$\delta_i = max\delta_i, i \neq j \quad (4)$$

After obtaining the local density and relative distance, ρ_i and δ_i are used to construct the decision map, where the abscissa is the local density and the ordinate is the relative distance. Cluster centers can be selected based on the decision diagram, and the point with the highest decision value is usually selected as a candidate for the cluster center. Because these points usually have high local density and large relative distance, indicating that they are in the center of the high-density region and no other points around them have higher density.

Decision value:

$$\gamma_i = \rho_i \cdot \delta_i \quad (5)$$

The main steps of the DPC algorithm are as follows. First, the local density of each point is computed using Equation (1) or (2). Then, the pairwise distances between all points in the dataset are calculated to form a distance matrix D, and the relative distance of each point is determined using

Equation (3) or (4). Next, the decision value is computed using Equation (5), and all points are ranked in descending order based on this value. The top K points are selected as cluster centers, and finally, the remaining points are assigned to the cluster of their nearest neighbor with a higher local density.

III. The K-DPC algorithm

Based on an analysis of the classical Density Peak Clustering (DPC) algorithm, this paper proposes an enhanced version, the K-DPC algorithm. The clustering procedure involves key steps: first, the dataset is refined by identifying and removing noise points. Second, the concepts of k-nearest neighbors (kNN) and k-mutual nearest neighbors (kMNN) are integrated to optimize local density calculation, which alleviates bias caused by inter-class density disparities, improves the accuracy of cluster center selection, and reduces the risk of misidentifying cluster centers. Additionally, a point assignment strategy based on kNN density reachability and a neighbor-weighting model is introduced, effectively mitigating the domino effect and enhancing the reliability and accuracy of non-central point assignment. Overall, the algorithm significantly improves the clustering performance of complex datasets through the optimization of key processes.

A. Removal of noise points

In the classical Density Peaks Clustering (DPC) algorithm, noise data points are typically not explicitly identified or processed, which disrupts local density calculations and cluster center identification, thereby degrading clustering accuracy. To address this, this paper proposes a density-threshold-based noise filtering strategy. By quantifying the local density of each data point, points with densities below a predefined threshold are identified as noise and eliminated during preprocessing. This approach yields a cleaner dataset for subsequent clustering, ultimately enhancing clustering quality.

Definition 1:

For a given dataset $X = \{x1, x2, x3...xn\}$, for $\forall \mathbf{x_i}, \mathbf{x_j} \in \mathbf{X}$, the average distance and standard deviation of x_i from its k-nearest neighbors are as follows.

$$AvgDist_i = \frac{\sum_{x_j \in knn_{x_i}} dist(x_i, x_j)}{K} \tag{6}$$

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(AvgDist_i - \frac{\sum_{i=1}^{n}(AvgDist_i)}{n}\right)^2} \tag{7}$$

$knn(x_i)$is the set of the K nearest neighbors of x_i,calculate the average distance and standard deviation between each point and its k-nearest neighbors. Noise points typically do not cluster closely with other points; therefore, the distances between a noise point and its nearest neighbors tend to be more dispersed. Consequently, noise points generally exhibit relatively large average distances and standard deviations. Thus, points with a larger sum of average distance and standard deviation are more likely to be noise points.

Definition 2:

For a given dataset $X = \{x1, x2, x3...xn\}$, for $\forall \mathbf{x_i} \in \mathbf{X}$, the noise points are:

$$AvgDist_i > \frac{\sum_{i=1}^{n}(AvgDist_i)}{n} + N\sigma \tag{8}$$

In this paper, a specific threshold is used to distinguish noise points from non-noise points in the dataset. If the average distance of a data point's k-nearest neighbors exceeds the overall average distance of all points' k-nearest neighbors plus N times the standard deviation, the point is classified as noise. When additional noise points are artificially added, N is set to 2 to improve detection effectiveness. In the absence of artificially added noise, N is set to 8.

B. Selection of cluster centers

The classical Density Peak Clustering (DPC) algorithm identifies cluster centers by multiplying local density and density distance. However, its local density calculation is hypersensitive to parameters such as the cutoff distance. In low-density regions, the neighborhood range is often ambiguously defined, leading to suboptimal cluster center identification. Additionally, pronounced density disparities between clusters can induce calculation bias. To address these issues, this paper introduces a local density reconstruction method integrating K-mutual nearest neighbors (K-MNN) and K-nearest neighbors (K-NN), optimizing local density calculation via a dual-nearest-neighbor constraint mechanism. This approach effectively mitigates bias arising from inter-class density disparities, enhances cluster center identification in low-density areas, minimizes misclassification of center points due to inadequate local density estimation, and elevates the accuracy of initial cluster center selection.

Definition 3:

For a given dataset $X = \{x1, x2, x3...xn\}$, for $\forall \mathbf{x_i}, \mathbf{x_j} \in \mathbf{X}$,the k-mutual nearest neighbors of x_i are as follows.

$$KMNN = \{x_i, x_j \in X \mid x_i \in knn(x_j), x_j \in knn(x_i)\} \tag{9}$$

K-Mutual Nearest Neighbors (KMNN),Data x_i in data set X is in the k-nearest neighbor set $knn(x_j)$ of x_j ,Also x_j is in the set $knn(x_i)$ of k-nearest neighbors of x_i,then the data x_i and x_j are called k-mutual neighbors.

Definition 4:

For a given dataset $X = \{x1, x2, x3...xn\}$, for $\forall \mathbf{x_i}, \mathbf{x_j} \in \mathbf{X}$,the k-inverse nearest neighbors of x_i are as follows.

$$INN = \{x_i, x_j \in X \mid x_i \in knn(x_j)\} \tag{10}$$

K-inverse nearest neighbor is referred to as INN,If x_i in a dataset X is in the set $knn(x_j)$ of x_j's k-nearest neighbors, we say x_j is x_i's k-inverse nearest neighbor.

Definition 5:

Degree of correlation: For a given dataset $X = \{x1, x2, x3...xn\}$, for $\forall \mathbf{x_i}, \mathbf{x_j} \in \mathbf{X}$,The degree of correlation between x_i and x_j is:

$$\mathrm{R}\left(\mathbf{x_i}, \mathbf{x_j}\right) = \begin{cases} a \cdot \left(exp\left(-d_{ij}\right)^2\right) & x_j \in knn(x_i) \\ \frac{1}{(k \cdot d_{ij}+1)} & x_j \notin knn(x_i) \end{cases} \tag{11}$$

Equation (11) classifies samples into two categories: k-nearest neighbor samples and non-k-nearest neighbor samples. For the k-nearest neighbor samples, which are closer to the target point, their contribution to the target's local density is more significant. Therefore, the exponential function is used to describe the local density of the K nearest neighbor samples, and the weight of the density contribution is adjusted by multiplying by α to ensure that the density contribution can quickly decay with the increase of distance, and the value of α is 1/2. However, for non-k-nearest neighbor samples, since they are far away from the target point, their contribution to the local density is relatively small, so their local density is denoted by $1/(k \cdot d_{ij} + 1)$, which decays slowly with the increase of distance. The influence of all sample pairs, including k-nearest neighbor samples and non-k-nearest neighbor samples, is comprehensively considered, so that the k-nearest neighbor local density calculation is more comprehensive.

Definition 6:

For a given dataset $X = \{x1, x2, x3...xn\}$, for $\forall \mathbf{x_i} \in \mathbf{X}$,the local density of x_i is as follows.

$$\rho_i = kmnn_i + \frac{1}{n}\sum_{1}^{n} R(x_i, x_j)) \tag{12}$$

Formula (12) defines the improved local density. DPC identifies cluster centers by the principle that they have higher local density than neighbors. This paper enhances this by incorporating global density distribution via k-MNN and k-local density.k-MNN integrates k-NN and k-INN, with density increasing with surrounding points. k-local density mitigates regional density disparities through adaptive adjustments, and combined with Formula (5), identifies cluster centers in sparse areas.

Algorithm 1:

Input: original dataset, k-nearest neighbor value

Output: denoised dataset, cluster center

Step1: Data normalization, calculate the sample Euclidean distance matrix

Step2: Calculate the average distance AvgDist between each point and its K nearest neighbors according to Equations (6) and (7), and use the average distance to calculate the standard deviation σ.

Step3: According to Formula (8), find the points whose average k-nearest neighbor distance is greater than the average of all k-nearest neighbor distance plus N times the standard deviation, and mark them as noise points.

Step4: The noise points are removed and the local density ρ_i is calculated according to equations (9) and (12).

Step5: Select the cluster center C by constructing the decision diagram and calculating γ_i through ρ_i and δ_i according to equation (5).

The Local Outlier Factor (LOF) algorithm identifies outliers by comparing the local density of each data point with that of its neighbors. While LOF effectively detects local outliers in heterogeneous density regions without assuming global distribution patterns and demonstrates effectiveness in multidimensional data, it has inherent limitations. Specifically, LOF assesses only local neighborhood relationships without accounting for global data structure, potentially failing to identify outliers that substantially deviate from the entire dataset (e.g., specific noise points in Figure 1a). Traditional Density Peak Clustering methods typically utilize fixed cutoff distances or Gaussian kernel functions for noise detection, which may struggle to adapt to heterogeneous data densities. Although recent improvements have enhanced noise identification capabilities, they still insufficiently incorporate global density features. In contrast, the K-DPC algorithm computes the standard deviation of average k-nearest neighbor distances, enabling noise identification by setting distance thresholds relative to this deviation. This approach effectively integrates local and global density information, adapting to dataset density variations while accurately differentiating noise from valid data points. As illustrated in Figures 1b and 1d, K-DPC precisely identifies noise points without misclassifying normal data, demonstrating superior noise recognition performance.

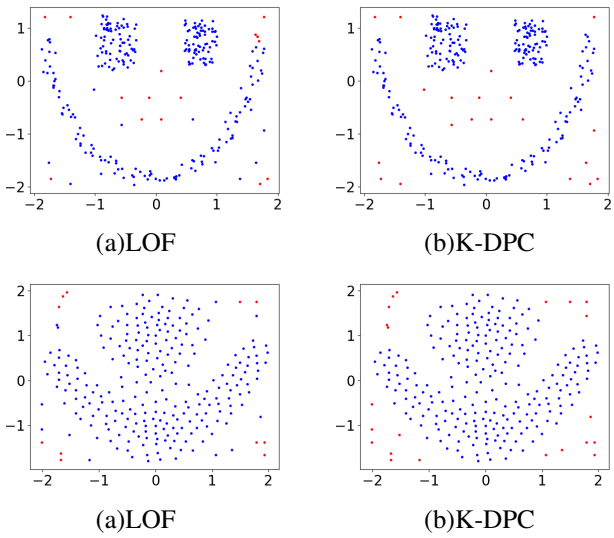


Fig. 1. Comparison of LOF and K-DPC algorithms for noise point detection

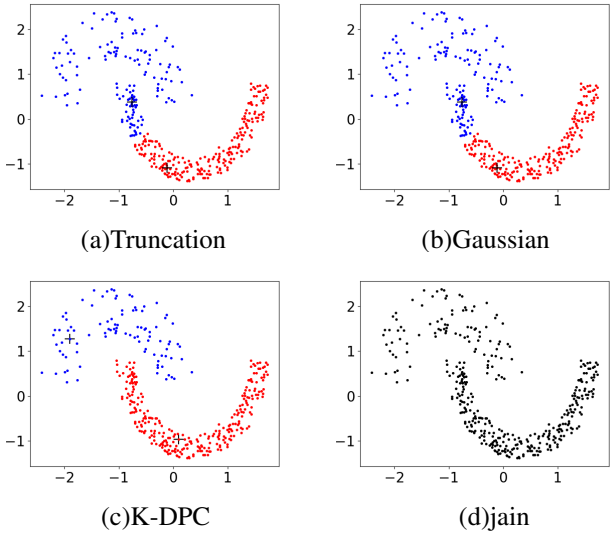


Fig. 2. Cluster centers of different density calculation methods on Jain dataset

The figure highlights notable limitations in the traditional DPC algorithm's cluster center identification, largely stemming from its strong dependency on local density measures.

This dependency often results in an inadequate characterization of the global data structure, particularly in datasets with pronounced inter-cluster density variations. In such cases, local density metrics frequently fail to accurately reflect the global data distribution, thereby compromising clustering performance. For instance, when processing datasets with heterogeneous density distributions, the traditional DPC algorithm exhibits two critical drawbacks: (1) over-segmentation of high-density regions into multiple clusters, and (2) misclassification of low-density points into adjacent clusters or incorrect labeling as noise. These limitations are clearly illustrated in Figures 2(a) and 2(b) using the Jain dataset.

To address these challenges, the K-DPC algorithm integrates a k-mutual nearest neighbor (k-MNN) approach to redefine local density computation. By incorporating density information from k-nearest neighbors, K-DPC adaptively adjusts density estimates, effectively mitigating disparities between sparse and dense regions. This advanced method achieves three key improvements: (1) precise cluster center identification even in low-density regions, (2) elimination of the constraints imposed by the traditional cutoff distance parameter, and (3) establishment of a more robust and flexible cluster center detection framework.

C. Remaining points assignment

This paper proposes an enhanced non-central point assignment strategy that integrates k-nearest neighbors and weight aggregation mechanisms to optimize the clustering process. The strategy first constructs an initial clustering set using cluster centers and their k-nearest neighbors. For high-density points, it identifies classified neighbors within their k-neighborhood, quantifies the point-cluster association degree via logarithmic transformation and distance summation, and assigns points to the cluster with the maximum transformed distance sum. For non-dense points, a density-weighted model based on Gaussian kernel functions is developed to aggregate weights within the k-neighborhood, assigning points to the cluster with the highest cumulative weight sum—this approach facilitates rational assignment of data points in low-density regions. By fusing kNN connectivity and weight aggregation, the method effectively mitigates the domino effect of single-point misassignment, enhances the robustness of the assignment process, and significantly improves clustering accuracy for heterogeneous density datasets, offering an efficient solution for complex data distributions.

Definition 7:

High-density point: For each point, compare its k-nearest neighbors and k-inverse neighbors. If the number of intersections between its k-nearest neighbors and k-inverse neighbors is greater than or equal to K-1, the point is defined as a high-density point.

$$H\left(x_i\right) = Sum\left(KNN\left(x_i\right) \cap NNN\left(x_i\right)\right) \geq K-1 \quad (13)$$

Definition 8:

The sum of reciprocal distances is calculated where C_i is the ith cluster. This formula simply calculates the sum of the reciprocal distances from point p to the points in cluster C_i. Setting distance threshold: In order to remove very small distance values, define a threshold ϵ,ϵ =1.

$$d'_{pq} = max\left(d_{pq}, \epsilon\right) \quad (14)$$

$$distsum'\left(p, c_i\right) = \sum_{q \in KNN_{(p) \cap c_i}} log\left(\frac{1}{d'_{pq}}\right) \quad (15)$$

Definition 9:

For each point k-nearest neighbor data $K = \{k1, k2, k3...kn\}$, for $\forall \mathbf{k_i}, \in K$,different density weights are set according to the distance from the point.According to the distance from the point through the Gaussian formula, the basic weight and additional weight are calculated to set different density weights.

$$baseweight = exp\left(-k\left(\frac{d_{ij}}{bandwidth}\right)^2\right) \quad (16)$$

$$addweight = \frac{1}{1+d_{ij}} \quad (17)$$

$$\alpha_i = baseweight_i \times addweight_i \quad (18)$$

By constructing a distance-sensitive weight mechanism for neighboring points, those with closer proximity to unassigned points are assigned higher weights during the assignment process. Furthermore, a multi-cluster neighborhood influence factor is introduced to explicitly model heterogeneous membership probabilities of neighbors, enabling more accurate point assignment decisions. The strategy of co-constructing an initial clustering set based on center points and their k-nearest neighbors effectively reduces the iterative optimization search space. By prestructuring the neighborhood structure, the search complexity in the subsequent assignment process is minimized, thus enhancing algorithm efficiency while ensuring assignment rationality.

Algorithm 2:

Input: Denoised dataset, cluster center

Output: Clustering results

Step1: Identify dense points using formula (13). Iteratively classify dense points by integrating formulas (14) and (15).

Step2: For non-dense points, assign each point to the cluster with the maximum weight sum via formulas (16) $\sim$ (18), and repeat this process iteratively.

Step3: Post-iteration, if a point's weight distribution is uniform across clusters, assign it to the cluster containing its nearest neighbor.

Step4: Output the final clustering results.

Computational time complexity:

Assuming that the total number of samples is N, the time complexity of K-DPC algorithm consists of the following six steps:1) The time complexity of calculating the Euclidean distance between samples is $\mathrm{O}(N^2)$, 2) perform the calculation of the mean and standard deviation, which is A linear operation, and its time complexity is $\mathrm{O}(N)$, 3) sort the Euclidean distance and determine the time complexity of the first K nearest neighbor points of each sample is $\mathrm{O}(nlog_2n)$,4) the time complexity of calculating the local density of K of each sample is $\mathrm{O}(N^2)$, 5) The time complexity is $\mathrm{O}(N^2)$ when classifying high density points, and 6) the time complexity of performing density-based weight and label assignment is $\mathrm{O}(N)$. Therefore, the overall time complexity of the K-DPC algorithm is $\mathrm{O}(N^2)$.

IV. Experimental results and analysis

A. Experiment Preparation

In order to confirm the practicability of K-DPC algorithm, this study selected data sets with multiple dimensions, sizes and additional noise for testing, and compared it with eight clustering algorithms including DPC, DBSCAN, K-means, mean-shift, NNODPC, KKDPC, and DK-means.The evaluation criteria are NMI (normalized mutual information) [25], RI (Rand index) [26], ARI (adjusted Rand index) [26], AMI (adjusted mutual information) [27] and FMI (Fowlkes-Mallows index) [26]. The ideal value of these indicators is 1, and the closer to 1 indicates the better performance of the algorithm.The hardware configuration for the experiments consisted of using an Intel(R) Core(TM) i7-7700 processor, 8 GB memory, a computer running Windows 11 64-bit operating system, and PyCharm 2023.2 as the programming environment.

TABLE I
Artificial datasets

Dataset	Number of instances	Number of features	Number of clusters	Literature
Spiral	312	2	3	[28]
Flame	240	2	2	[29]
Zelink3	266	2	3	[30]
Aggregation	788	2	7	[31]
Jain	373	2	2	[32]
R15	600	2	15	[33]
Ring	1000	2	2	[22]
Compound	399	2	6	[35]
Pathbased	300	2	3	[36]

TABLE II
Real datasets

Dataset	Number of instances	Number of features	Number of clusters	Literature
Iris	150	4	3	[34]
Wine	178	1	3	[34]
Sonar	208	60	2	[34]
Vihecle	846	18	4	[34]
Zoo	101	17	7	[34]
Abalone	4177	8	3	[34]
Ecoli	326	8	8	[34]
pima	768	8	2	[34]
yeast	1484	8	10	[34]
Parkinsons	195	23	2	[34]
Ionosphere	351	34	2	[34]
dermatology	366	34	6	[34]
Libras	360	91	51	[34]
Balancescale	625	4	3	[34]

B. Experimental results on synthetic datasets

This paper introduces K-DPC, a novel clustering algorithm, and evaluates its performance against existing methods across multiple datasets. K-DPC is systematically assessed on nine artificial datasets using standard metrics: Normalized Mutual Information (NMI), Rand Index (RI), Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI), and Fowlkes–Mallows Index (FMI). The algorithm is compared with DPC, DBSCAN, K-means, mean-shift, KNN, KKDPC, and DK-means to validate its effectiveness under diverse data distributions. Experimental results provide a comprehensive comparison, demonstrating K-DPC's superior robustness and accuracy relative to state-of-the-art clustering approaches.

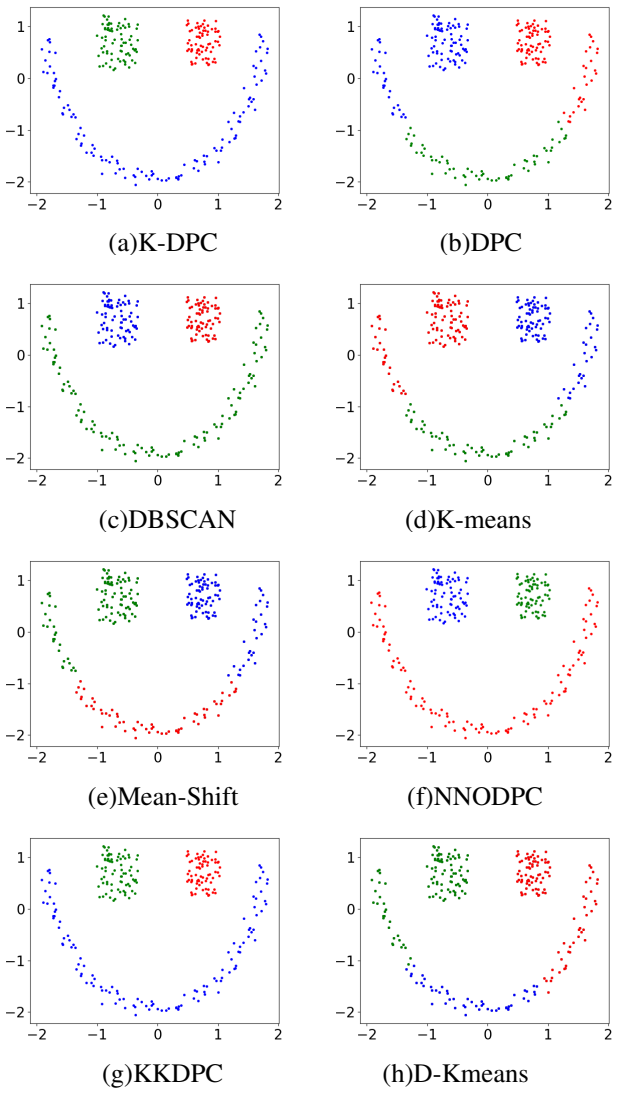
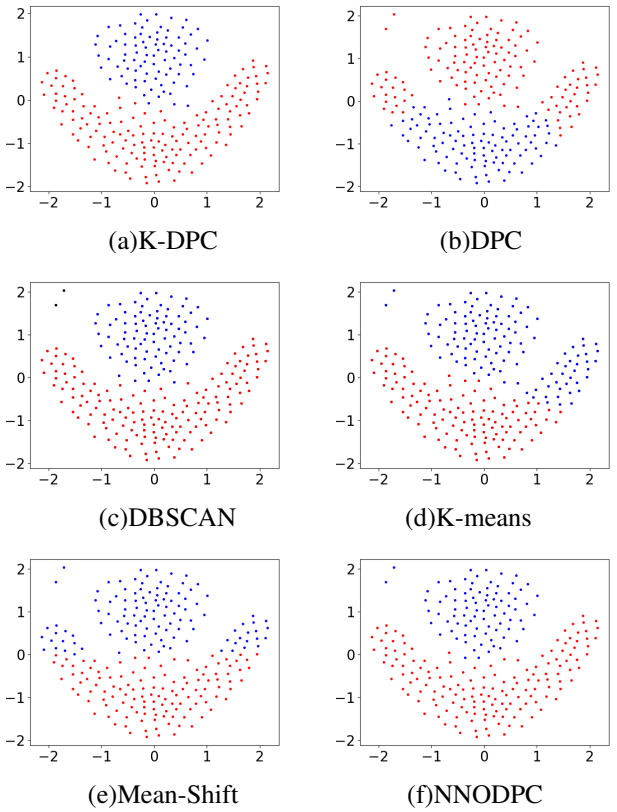
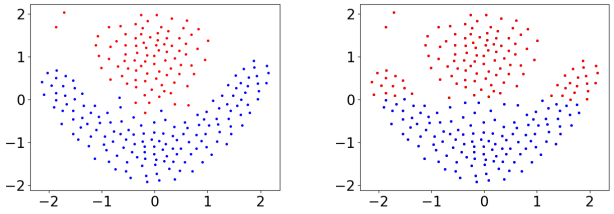


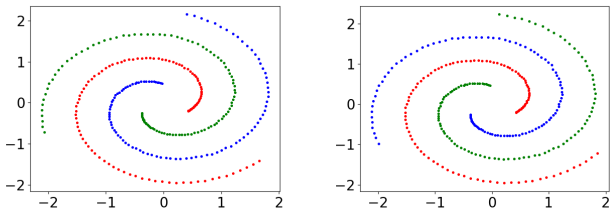
Fig. 3. Clustering results on the Zelnik3 dataset



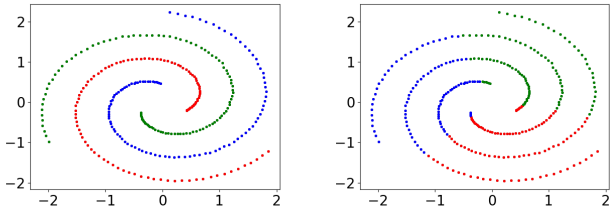


(g)KKDPC (h)D-Kmeans

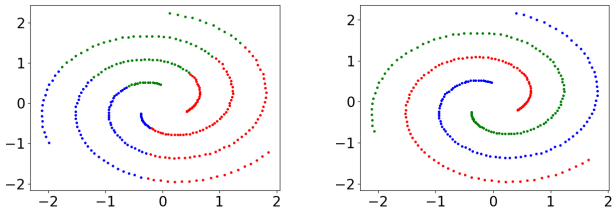
Fig. 4. Clustering results on the Flame dataset



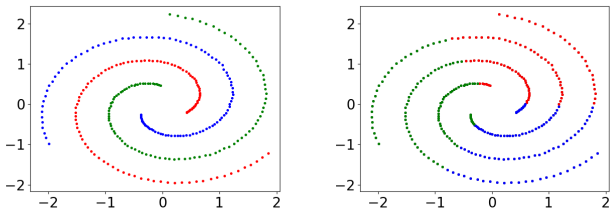
(a)K-DPC (b)DPC



(c)DBSCAN (d)K-means

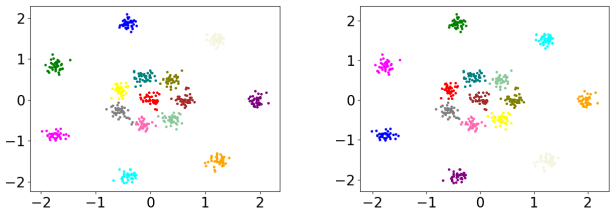


(e)Mean-Shift (f)NNODPC

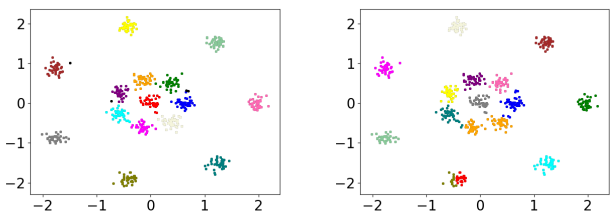


(g)KKDPC (h)D-Kmeans

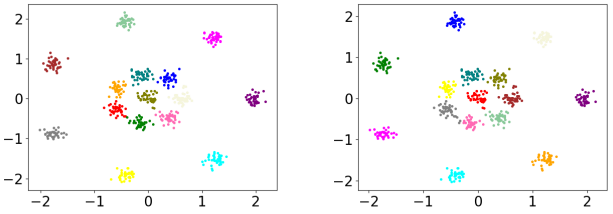
Fig. 5. Clustering results on the Spiral dataset



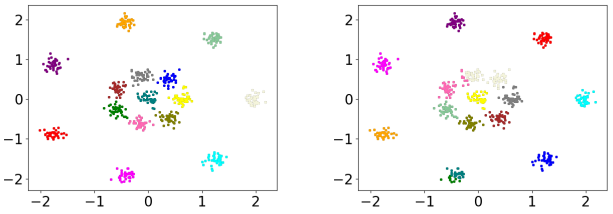
(a)K-DPC (b)DPC



(c)DBSCAN (d)K-means

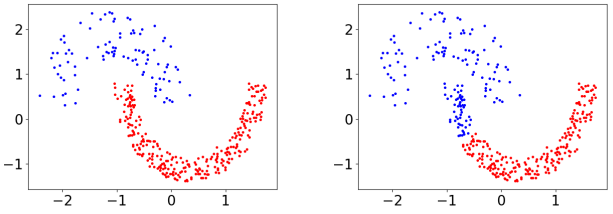


(e)Mean-Shift (f)NNODPC

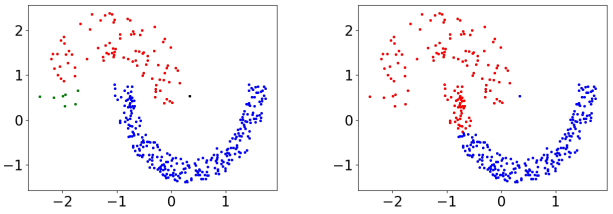


(g)KKDPC (h)D-Kmeans

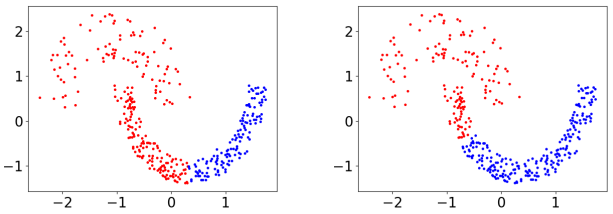
Fig. 6. Clustering results on the R15 dataset



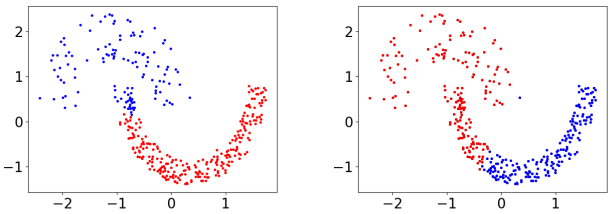
(a)K-DPC (b)DPC



(c)DBSCAN (d)K-means

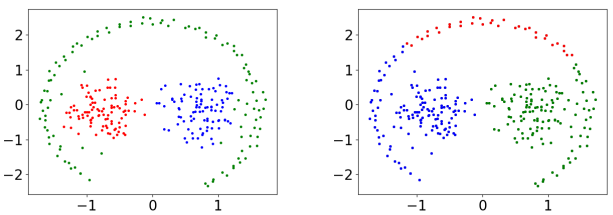


(e)Mean-Shift (f)NNODPC



(g)KKDPC (h)D-Kmeans

Fig. 7. Clustering results on Jain dataset



(a)K-DPC (b)DPC

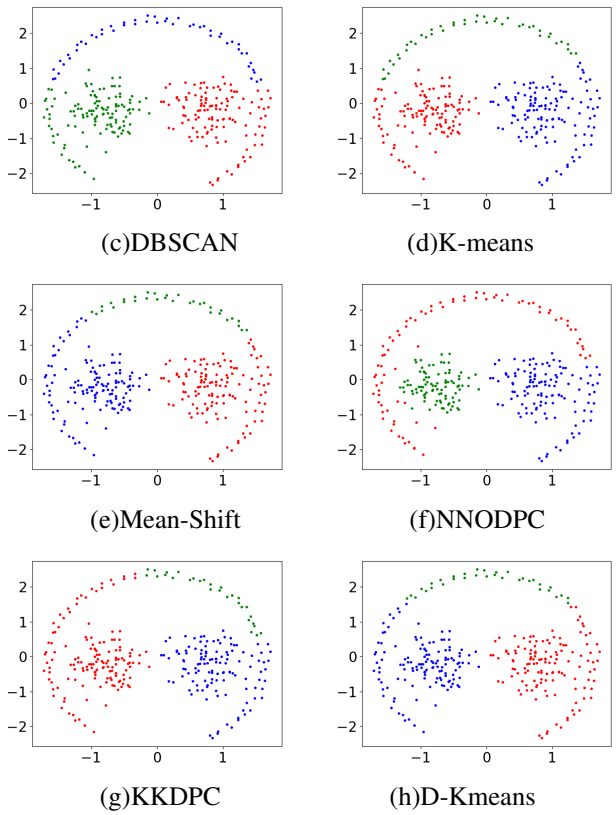


Fig. 8. Clustering results on the Pathbased dataset

1) Analysis of experimental results on artificial data sets:

First, the experimental results for the Zelink3, Spiral, and Pathbased datasets were systematically analyzed. For Zelink3, characterized by a sparse ring cluster and two dense clusters, K-DPC, DBSCAN, NNODPC, and KKDPC achieved precise clustering outcomes, successfully identifying all structural components. In contrast, DPC and K-means exhibited misclassification issues, erroneously assigning portions of the ring cluster to adjacent dense clusters, thereby potentially compromising clustering accuracy.

For the Spiral dataset, density-based algorithms such as K-DPC and DPC exhibited superior performance, effectively capturing helical structures and accurately classifying data points. This efficacy arises from their adaptive neighborhood definitions, which facilitate flexible boundary detection in non-linear structures. Conversely, K-means, Mean-Shift, and DK-means performed suboptimally owing to their reliance on Euclidean distance metrics and parametric assumptions regarding data distribution. Specifically, K-means and DK-means, which optimize cluster assignments based on centroid means and variances, exhibited systematic misclassification when confronted with the spiral's non-Gaussian, intertwined topology—errors that highlighted the limitations of centroid-based approaches. The Mean-Shift algorithm, which relies on gradient ascent to find local density maxima, faced difficulties in resolving overlapping density peaks, resulting in inconsistent determination of cluster centers.

On the Pathbased dataset, K-DPC achieved optimal results by integrating k-mutual nearest neighbor (k-MNN) connectivity to characterize hierarchical density landscapes. In contrast, the DPC algorithm generated erroneous clustering outcomes due to its heuristic initial center selection, which failed to accommodate the dataset's elongated, path-like structures. K-means and DK-means exhibited subpar allocation performance, as their iterative reallocation strategies—reliant on global centroid calculations—could not adapt to the dataset's local density fluctuations.

Analysis of the Jain, Flame, and R15 datasets revealed that K-DPC demonstrates superior clustering performance on Jain—a dataset characterized by two intersecting crescent-shaped clusters. In contrast, traditional DPC and related algorithms erroneously assigned cluster centers to the higher-density lower-half cluster, leading to systematic misclassifications of the sparser upper-half points. KKDPC outperformed traditional DPC by integrating kernel-based density estimation for initial cluster point selection but failed to effectively capture the Jain dataset's non-convex geometry—highlighting the need for adaptive neighborhood mechanisms such as those employed in K-DPC.

For the Flame dataset, both K-DPC and KKDPC demonstrate robust performance in detecting cluster structures. Conversely, DBSCAN's accuracy is compromised by its sensitivity to noise, while K-means and nnopc exhibit suboptimal performance requiring improvement. On the artificial R15 dataset, most algorithms achieve satisfactory results: DBSCAN accurately identifies the number of clusters despite noise constraints, whereas K-means and DK-means struggle with tightly connected clusters, resulting in incorrect point assignments and diminished performance.

K-DPC achieves significantly higher average scores across multiple metrics on this artificial dataset—0.9706 (NMI), 0.9905 (RI), 0.9759 (ARI), 0.9701 (AMI), and 0.9830 (FMI)—compared to DPC (0.6435, 0.7992, 0.5733, 0.6272, 0.7676), with performance improvements ranging from 0.191 to 0.402. This underscores its superior capability in handling artificial data distributions.

Table IV employs the Friedman test, a non-parametric statistical method, to comprehensively evaluate the performance of eight clustering algorithms on artificial datasets. A lower rank mean (calculated by averaging performance ranks, where rank 1 denotes the best and rank 8 the worst) indicates superior performance relative to other methods. The K-DPC algorithm demonstrates an outstanding rank mean below 2 across all five metrics—NMI, RI, ARI, AMI, and FMI—significantly outperforming comparative algorithms. For instance, its rank mean of 1.89 surpasses DPC (3.50–3.72), DBSCAN (5.61–5.72), and K-means (6.00–6.39) by at least 3.39 points, highlighting its consistent top-tier performance.

2) Analysis of experimental results on real data sets:

Table VI shows K-DPC outperforms classical and state-of-the-art clustering methods on 14 real-world datasets across NMI, RI, ARI, AMI, and FMI. Compared to DPC and DBSCAN, K-DPC demonstrates remarkable superiority, especially on Iris, Wine, and Parkinsons datasets, where it achieves the highest scores in all metrics. K-DPC's mean values exceed DPC's by 0.112 to 0.178, confirming its advantage in dealing with various data distributions.

The performance gain of K-DPC originates from its precise initial cluster center selection, which effectively mitigates the domino effect in point allocation. Although K-DPC

TABLE III
EXPERIMENTAL RESULTS ON ARTIFICIAL DATASETS

Dataset	Metrics	K-DPC	DPC	DBSCAN	K-means	Mean-Shift	NNODPC	KKDPC	DK-means	Avg
Zelink3	NMI	**1.0**	0.4779	**1.0**	0.5405	0.6755	**1.0**	**1.0**	0.5348	0.7786
	RI	**1.0**	0.6860	**1.0**	0.7316	0.8157	**1.0**	**1.0**	0.7281	0.8702
	ARI	**1.0**	0.3345	**1.0**	0.4145	0.5635	**1.0**	**1.0**	0.4073	0.7150
	AMI	**1.0**	0.4739	**1.0**	0.5372	0.6716	**1.0**	**1.0**	0.5314	0.7768
	FMI	**1.0**	0.5863	**1.0**	0.6228	0.6998	**1.0**	**1.0**	0.6185	0.8159
	Arg-	6	0.15	0.28/6	3	0.28	15	0.3	0.002	-
Flame	NMI	**0.9630**	0.4131	0.8749	0.3939	0.4267	0.8994	0.9354	0.4267	0.6666
	RI	**0.9916**	0.6639	0.9695	0.7155	0.7324	0.9752	0.9834	0.7324	0.8455
	ARI	**0.9831**	0.3269	0.9388	0.4312	0.4649	0.8990	0.9666	0.4649	0.6844
	AMI	**0.9628**	0.4112	0.8741	0.3920	0.4249	0.9502	0.9353	0.4249	0.6719
	FMI	**0.9922**	0.6786	0.9712	0.7253	0.7417	0.9768	0.9846	0.7417	0.8515
	Arg-	8	0.28	0.3/5	2	0.36	5.2	0.06	0.085	-
Spiral	NMI	**1.0**	**1.0**	**1.0**	0.0002	0.0026	**1.0**	**1.0**	0.0003	0.6254
	RI	**1.0**	**1.0**	**1.0**	0.5541	0.5553	**1.0**	**1.0**	0.5541	0.8329
	ARI	**1.0**	**1.0**	**1.0**	-0.0062	-0.0035	**1.0**	**1.0**	-0.0060	0.6230
	AMI	**1.0**	**1.0**	**1.0**	-0.0057	-0.0033	**1.0**	**1.0**	-0.0055	0.6232
	FMI	**1.0**	**1.0**	**1.0**	0.3272	0.3290	**1.0**	**1.0**	0.3274	0.7479
	Arg-	5	0.25	0.3/4	3	0.2	0.25	0.25	0.07	-
R15	NMI	**0.9942**	**0.9942**	0.9701	0.9331	**0.9942**	0.9745	**0.9942**	0.9580	0.9766
	RI	**0.9991**	**0.9991**	0.9958	0.9762	**0.9991**	0.9990	**0.9991**	0.9870	0.9943
	ARI	**0.9928**	**0.9928**	0.9651	0.8208	**0.9928**	0.9727	**0.9928**	0.8974	0.9534
	AMI	**0.9938**	**0.9938**	0.9677	0.9282	**0.9938**	0.9224	**0.9938**	0.9550	0.9686
	FMI	**0.9932**	**0.9932**	0.9675	0.8367	**0.9932**	0.9294	**0.9932**	0.9050	0.9514
	Arg-	15	0.11	0.11/8	15	0.061	15	0.5	0.022	-
Jain	NMI	**1.0**	0.5037	0.9329	0.3443	0.2889	0.5037	0.6508	0.3443	0.6312
	RI	**1.0**	0.7589	0.9897	0.6407	0.5864	0.7589	0.8604	0.6407	0.8091
	ARI	**1.0**	0.5133	0.9783	0.2817	0.2300	0.5026	0.7136	0.2817	0.6229
	AMI	**1.0**	0.5026	0.9324	0.3429	0.2865	0.5026	0.6501	0.3429	0.6311
	FMI	**1.0**	0.7905	0.9916	0.6815	0.5925	0.7905	0.8819	0.6815	0.8270
	Arg-	6	0.18	0.31/4	2	0.27	9.2	0.58	0.058	-
Ring	NMI	**1.0**	0.1314	**1.0**	0.0007	0.3601	**1.0**	0.2401	0.0006	0.4666
	RI	**1.0**	0.4995	**1.0**	0.5000	0.6328	**1.0**	0.5599	0.4999	0.7115
	ARI	**1.0**	0.0007	**1.0**	0.0035	0.2657	**1.0**	0.1201	0.0001	0.4238
	AMI	**1.0**	0.0007	**1.0**	0.0028	0.3592	**1.0**	0.2394	0.0001	0.4503
	FMI	**1.0**	0.5590	**1.0**	0.5005	0.6464	**1.0**	0.6473	0.5010	0.7318
	Arg-	6	0.2	0.5/4	2	0.58	55	0.5	0.022	-
Aggregation	NMI	0.9822	0.9957	0.9784	0.8395	0.8120	0.8733	**1.0**	0.8429	0.9155
	RI	0.9958	0.9993	0.9947	0.9190	0.9170	0.9270	**1.0**	0.9183	0.9589
	ARI	0.9876	0.9978	0.9843	0.7382	0.7829	0.8716	**1.0**	0.7357	0.8873
	AMI	0.9820	0.9956	0.9780	0.8374	0.8105	0.8030	**1.0**	0.8407	0.9059
	FMI	0.9903	0.9983	0.9877	0.7948	0.8463	0.8561	**1.0**	0.7928	0.9083
	Arg-	7	1.85	0.16/8	7	0.08	13	0.03	0.062	-
Compound	NMI	**0.8678**	0.7335	0.8356	0.6460	0.8585	0.8315	0.8015	0.7490	0.7904
	RI	**0.9505**	0.8414	0.9224	0.8055	0.9337	0.8996	0.9131	0.8417	0.8885
	ARI	**0.8711**	0.5366	0.8087	0.4178	0.8323	0.8252	0.7740	0.5676	0.7042
	AMI	**0.8642**	0.7281	0.8333	0.6388	0.8552	0.7337	0.7970	0.7435	0.7742
	FMI	**0.9052**	0.6417	0.8674	0.5449	0.8804	0.8010	0.8333	0.6721	0.7683
	Arg-	5	0.18	0.3/4	6	0.12	7	0.2	0.11	-
Pathbased	NMI	**0.9283**	0.5417	0.6810	0.5508	0.5485	0.7304	0.5014	0.5508	0.6291
	RI	**0.9772**	0.7445	0.8639	0.7515	0.7497	0.8589	0.7124	0.7515	0.8012
	ARI	**0.9488**	0.4574	0.6734	0.4687	0.4658	0.7287	0.4162	0.4687	0.5785
	AMI	**0.9279**	0.5387	0.6745	0.5478	0.5455	0.6846	0.4975	0.5478	0.6205
	FMI	**0.9659**	0.6612	0.7745	0.6656	0.6644	0.7911	0.6558	0.6656	0.7305
	Arg-	13	0.22	0.23/3	3	0.2	9	0.05	0.06	-

TABLE IV
RANK MEAN OF THE METRICS ON THE ARTIFICIAL DATASET

Dataset	NMI	RI	ARI	AMI	FMI
K-DPC	7.11	7.11	7.11	7.11	7.11
DPC	3.72	3.39	3.50	3.61	3.61
DBSCAN	5.61	5.72	5.61	5.72	5.72
K-means	2.00	2.39	2.22	2.33	2.00
Mean Shift	3.78	3.78	4.00	3.89	3.89
NNODPC	5.83	5.61	5.83	5.06	5.61
KKDPC	5.33	5.44	5.33	5.33	5.56
DK-means	2.61	2.56	2.39	2.94	2.50

TABLE V
RANK MEAN OF THE METRICS ON THE REAL DATASET

Dataset	**NMI**	**RI**	**ARI**	**AMI**	**FMI**
K-DPC	7.57	6.93	7.50	7.64	6.07
DPC	4.21	3.86	4.07	4.79	4.14
DBSCAN	4.71	3.64	3.64	4.43	4.04
K-means	3.86	4.39	4.21	4.36	3.43
Mean Shift	3.82	4.39	3.68	3.18	3.96
NNODPC	4.54	3.54	4.46	3.07	4.39
KKDPC	3.75	3.32	3.68	4.29	5.68
DK-means	3.54	5.39	4.75	4.25	4.21

TABLE VI
EXPERIMENTAL RESULTS OF THE REAL DATASETS

Dataset	Metrics	K-DPC	DPC	DBSCAN	K-means	Mean-Shift	NNODPC	KKDPC	DK-means	Avg
Iris	NMI	**0.9115**	0.6532	0.6618	0.5815	0.6539	0.7822	0.6964	0.6539	0.6993
	RI	**0.9641**	0.7261	0.7693	0.7149	0.7607	0.8720	0.7598	0.7607	0.7910
	ARI	**0.9188**	0.4531	0.5413	0.4200	0.5350	0.7795	0.5312	0.5350	0.5892
	AMI	**0.9103**	0.6483	0.6564	0.5757	0.6512	0.7159	0.6912	0.6512	0.6875
	FMI	**0.9455**	0.6856	0.7418	0.6565	0.7483	0.8135	0.7440	0.7483	0.7604
	Arg-	4	0.41	0.82/2	3	0.35	7	0.6	0.02	-
Wine	NMI	**0.8659**	0.4193	0.5421	0.4288	0.4637	0.8211	0.5522	0.4288	0.5652
	RI	**0.9543**	0.7191	0.7274	0.7187	0.7199	0.9284	0.7075	0.7187	0.7743
	ARI	**0.8978**	0.3715	0.4140	0.3711	0.4507	0.8181	0.4310	0.3711	0.5157
	AMI	**0.8645**	0.4131	0.5252	0.4227	0.4602	0.8380	0.5456	0.4227	0.5615
	FMI	**0.9323**	0.5834	0.5252	0.5835	0.6949	0.8916	0.6872	0.5835	0.6852
	Arg-	15	2.12	2.3/2	3	0.3	4	0.1	4.41	-
Zoo	NMI	0.8030	0.4673	**0.8678**	0.7485	0.8331	0.5774	0.8422	0.6325	0.7215
	RI	0.9107	0.6689	**0.9774**	0.8578	0.9158	0.7844	0.9554	0.8022	0.8590
	ARI	0.7231	0.1723	**0.9366**	0.5644	0.7315	0.3894	0.8765	0.3846	0.5973
	AMI	0.7733	0.4046	**0.8481**	0.7163	0.7863	0.2046	0.8215	0.5845	0.6424
	FMI	0.7877	0.3976	**0.9513**	0.6582	0.7993	0.3441	0.9057	0.5111	0.6694
	Arg-	3	0.9	3/2	7	0.08	7	0.1	0.2	-
Sonar	NMI	**0.0772**	0.0121	0.0328	0.0053	0.0001	0.0238	0.0201	0.0053	0.0221
	RI	**0.5260**	0.5010	0.4975	0.5002	0.5002	0.4985	0.5018	0.5002	0.5032
	ARI	**0.0520**	0.0019	-0.0051	0.0004	0.0001	0.0157	0.0032	0.0004	0.0086
	AMI	**0.0715**	0.0084	0.0147	0.0018	0.0001	0.0034	0.0112	0.0018	0.0141
	FMI	0.5172	0.5424	0.5565	0.5055	**0.7073**	0.6958	0.7016	0.5055	0.5915
	Arg-	4	5.30	7/2	2	3	16	0.6	0.03	-
Vihecle	NMI	0.2160	0.1872	0.1587	0.1867	**0.3411**	0.0456	0.1362	0.1867	0.1823
	RI	0.6289	0.6315	0.6319	0.6523	**0.7506**	0.4392	0.5142	0.6523	0.6126
	ARI	**0.1433**	0.1168	0.1154	0.1216	0.0001	0.0378	0.0738	0.1216	0.0913
	AMI	**0.2126**	0.1839	0.1500	0.1835	0.0004	0.0225	0.1310	0.1835	0.1334
	FMI	0.4050	0.3733	0.3714	0.3590	0.0033	0.4178	**0.4223**	0.3590	0.3388
	Arg-	16	1.93	1.5/4	8	1.8	7	1.93	0.36	-
Ecoli	NMI	**0.6218**	0.4014	0.4757	0.5680	0.5884	0.5811	0.5811	0.6136	0.5539
	RI	0.8162	**0.6742**	0.7693	0.7853	0.8336	0.7468	0.7468	**0.8702**	0.7803
	ARI	0.5167	0.1736	0.4787	0.3571	0.6177	0.3993	0.3993	**0.6543**	0.4496
	AMI	**0.6088**	0.3740	0.4628	0.5494	0.5677	0.5628	0.5629	0.5964	0.5356
	FMI	0.6407	0.3968	0.6510	0.5012	**0.7431**	0.5803	0.5803	0.7423	0.6044
	Arg-	8	0.85	0.85/5	8	0.3	17	0.6	0.01	-
Abalone	NMI	0.1655	**0.1839**	0.1342	0.1258	0.0474	0.1203	0.0720	0.1258	0.1219
	RI	0.6076	0.5918	0.5962	0.5969	0.4694	**0.6306**	0.3857	0.5969	0.5594
	ARI	**0.1887**	0.1603	0.1024	0.1294	0.0220	0.1181	0.0157	0.1294	0.1082
	AMI	0.1651	**0.1835**	0.1245	0.1254	0.0464	0.1082	0.0713	0.1254	0.1187
	FMI	0.5058	0.4911	0.4089	0.4436	0.4718	0.3628	**0.5504**	0.4436	0.4597
	Arg-	35	0.65	0.3/4	3	1.8	38	0.65	0.19	-
Pima	NMI	**0.0662**	0.0146	0.0177	0.0556	0.0091	0.0481	0.0026	0.0299	0.0305
	RI	**0.5538**	0.5231	0.5440	0.5495	0.5487	0.4996	0.5140	0.5513	0.5355
	ARI	**0.1054**	0.0382	0.0165	0.0956	0.0171	0.0449	0.0138	0.0753	0.0509
	AMI	**0.0652**	0.0136	0.0128	0.0547	0.0074	-0.0438	0.0015	0.0288	0.0175
	FMI	0.5791	0.5629	**0.7061**	0.5780	0.0001	0.6129	0.5678	0.6312	0.5298
	Arg-	12	0.14	0.13/3	2	0.35	7	0.56	0.04	-
Yeast	NMI	**0.2335**	0.0751	0.0600	0.2030	0.0818	0.0117	0.0736	0.0693	0.1010
	RI	**0.7115**	0.4276	0.2554	0.6515	0.2738	0.2567	0.2852	0.4916	0.4192
	ARI	**0.1369**	0.0375	0.0102	0.1095	0.0206	0.0072	0.0007	0.0406	0.0454
	AMI	**0.2216**	0.0597	0.0487	0.1965	0.0717	-0.0095	0.0570	0.0649	0.0888
	FMI	0.3198	0.0001	0.0001	0.3426	0.0001	**0.4464**	0.4453	0.3410	0.2369
	Arg-	6	0.09	0.1/20	10	0.05	5	0.2	0.3	-
Parkinsons	NMI	**0.3523**	0.2516	0.1399	0.1088	0.1848	0.1562	0.0833	0.1253	0.1753
	RI	**0.7334**	0.7027	0.4923	0.5094	0.3761	0.5613	0.5829	0.6154	0.5717
	ARI	**0.3838**	0.2677	0.0728	-0.0965	0.0018	0.1481	0.1489	0.2115	0.1423
	AMI	**0.3481**	0.2464	0.1136	0.1042	0.0274	0.1717	0.0793	0.1215	0.1515
	FMI	**0.8200**	0.8131	0.0001	0.6338	0.0001	0.5819	0.6437	0.6740	0.5208
	Arg-	5	0.07	0.3/5	2	0.7	10	0.46	0.12	-
Ionosphere	NMI	0.2117	0.0630	**0.2803**	0.1299	0.0001	0.1652	0.0509	0.1320	0.1291
	RI	**0.6015**	0.4988	0.5716	0.5865	0.5385	0.4892	0.4986	0.5865	0.5464
	ARI	**0.2024**	-0.0315	0.1622	0.1727	0.0001	0.0884	-0.0307	0.1728	0.0921
	AMI	0.2100	0.0606	**0.2737**	0.1280	0.0001	0.0430	0.0486	0.1301	0.1118
	FMI	0.6181	0.5939	0.5311	0.6031	**0.7338**	0.2451	0.5911	0.6028	0.5649
	Arg-	7	0.31	0.8/4	2	1	3	0.5	0.35	-

TABLE VII
EXPERIMENTAL RESULTS OF THE REAL DATASETS (CONTINUE)

Dataset	Metrics	K-DPC	DPC	DBSCAN	K-means	Mean-Shift	NNODPC	KKDPC	DK-means	Avg
Dermatology	NMI	**0.7777**	0.7295	0.5238	0.7076	0.4594	0.4453	0.6538	0.2124	0.5637
	RI	0.8377	**0.8985**	0.7495	0.7729	0.8076	0.7996	0.7643	0.7154	0.7932
	ARI	0.5666	**0.6903**	0.3132	0.4012	0.0863	0.0015	0.4281	0.0889	0.3220
	AMI	**0.7721**	0.7234	0.4965	0.7006	0.3231	0.0001	0.6452	0.1951	0.4820
	FMI	0.6810	**0.7546**	0.4781	0.5557	0.0001	0.0088	0.5976	0.2653	0.4177
	Arg-	6	0.35	1/12	6	0.2	5	0.3	0.14	-
Libras	NMI	**0.6011**	0.5702	0.5852	0.5319	0.3920	0.5732	0.5697	0.5436	0.5459
	RI	0.8922	0.8669	0.8591	0.8907	0.8237	0.8715	0.8817	**0.8952**	0.8726
	ARI	0.3132	0.2496	0.1128	0.2268	0.1636	**0.4883**	0.2529	0.2617	0.2586
	AMI	**0.5442**	0.5125	0.4076	0.4676	0.3427	0.2366	0.5124	0.4823	0.4382
	FMI	**0.3801**	0.3338	0.1908	0.2880	0.0001	0.3151	0.3237	0.3212	0.2691
	Arg-	15	0.08	0.82/2	15	0.8	15	0.37	0.56	-
Balancescale	NMI	**0.1408**	0.0993	0.0001	0.0619	0.0745	0.1109	0.0466	0.0068	0.0676
	RI	0.5541	0.3046	0.1987	**0.6179**	0.5318	0.2944	0.2633	0.5968	0.4202
	ARI	**0.0846**	0.0084	0.0001	0.0455	0.0628	0.0270	0.0026	-0.0042	0.0284
	AMI	**0.1361**	0.0927	0.0001	0.0574	0.0720	0.0018	0.0390	-0.0048	0.0493
	FMI	0.3752	0.4143	**0.4458**	0.2932	0.0001	0.4109	0.4242	0.0001	0.2955
	Arg-	5	0.4	0.2/3	3	0.17	8	0.06	0.1	-

generally outperforms algorithms such as NNODPC, certain metrics on specific datasets may be surpassed by alternative methods. K-DPC leverages k-mutual nearest neighbor technology to determine cluster centers and employs an optimized local density definition for hierarchical density evaluation, thereby enhancing the accuracy of cluster center identification. Furthermore, the algorithm integrates k-nearest neighbor connectivity and weight-sum allocation to replace the original strategy, incorporating the spatial distribution characteristics of samples. These design choices enable K-DPC to deliver exceptional clustering results across both artificial and real-world datasets, underscoring its strong generalization capability and broad applicability.

Table V presents the application of the Friedman test to comprehensively assess the performance of eight algorithms on real-world datasets. A higher rank mean reflects superior performance compared to other methods. As illustrated in the table, K-DPC achieves a significantly higher rank mean than all comparative algorithms, showcasing its superiority across five evaluation metrics NMI, RI, ARI, AMI, and FMI. The results of the Friedman test further substantiate that K-DPC exhibits stability and clear advantages when handling real-world datasets with varying characteristics.

The Friedman test results further validate the robustness and superiority of K-DPC in managing datasets with diverse density distributions. This exceptional performance is attributed to its innovative density estimation approach and adaptive neighborhood mechanism, which enable precise identification of cluster centers even under complex distribution scenarios. In contrast to algorithms like K-means and DK-means that struggle with non-Gaussian structures, K-DPC demonstrates remarkable resilience in defining cluster boundaries and mitigating noise, as evidenced by its consistently high rank mean across all evaluation metrics. These findings establish K-DPC as an optimal solution for clustering tasks that require both accuracy and computational efficiency

C. Dataset with artificially added noise points

It is common to not specifically include noisy data in regular datasets.Therefore, we manipulated each of the datasets listed below, adding an additional 20 disturbing noise data points to be processed accordingly.

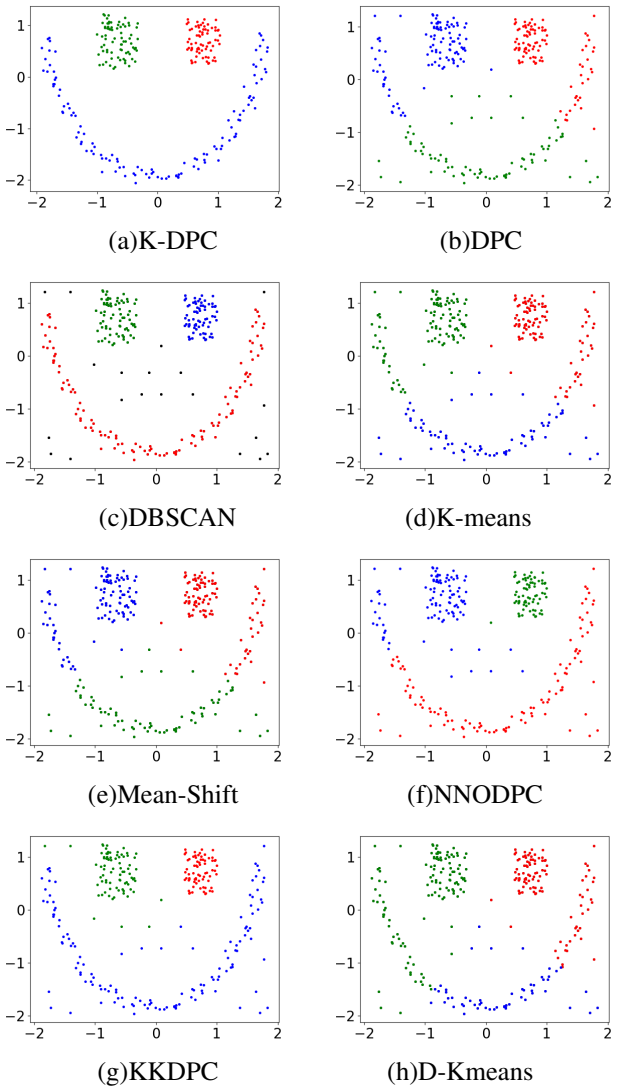


Fig. 9. Clustering results on the Zelnik3 dataset with noise

TABLE VIII
EXPERIMENTAL RESULTS OF NOISY DATA SETS

Dataset	Metrics	K-DPC	DPC	DBSCAN	K-means	Mean-Shift	NNODPC	KKDPC	DK-means	Avg
Spiral	NMI	**1.0**	0.6709	0.9294	0.0011	0.0012	0.6633	0.6780	0.0001	0.4930
	RI	**1.0**	0.8596	0.9614	0.5536	0.5522	0.8159	0.8374	0.5530	0.7666
	ARI	**1.0**	0.6864	0.9108	-0.0046	-0.0049	0.6614	0.6388	-0.0060	0.4852
	AMI	**1.0**	0.6691	0.9289	-0.0044	-0.0044	0.5993	0.6762	-0.0056	0.4824
	FMI	**1.0**	0.7925	0.9406	0.3302	0.3319	0.7428	0.7625	0.3292	0.6537
	Arg-	5	0.25	0.3/4	3	0.14	9	0.46	0.12	-
Flame	NMI	**0.9630**	0.3372	0.7765	0.3091	0.3199	0.6618	0.6911	0.3223	0.5441
	RI	**0.9916**	0.7076	0.9101	0.6929	0.7027	0.8773	0.8706	0.7076	0.8065
	ARI	**0.9831**	0.4153	0.8209	0.3859	0.4053	0.6608	0.7406	0.4151	0.6013
	AMI	**0.9628**	0.3353	0.7754	0.3071	0.3180	0.7542	0.6902	0.3204	0.5544
	FMI	**0.9922**	0.7113	0.9084	0.6969	0.7071	0.8829	0.8786	0.7130	0.8104
	Arg-	8	0.29	0.3/5	2	0.28	7	0.06	0.085	-
Zelink3	NMI	**1.0**	0.4463	0.9272	0.4365	0.4331	0.5502	0.8471	0.3890	0.6287
	RI	**1.0**	0.7338	0.9639	0.7140	0.7206	0.7611	0.9400	0.6937	0.8159
	ARI	**1.0**	0.4061	0.9178	0.3737	0.3821	0.5456	0.8673	0.3288	0.6027
	AMI	**1.0**	0.4427	0.9266	0.4327	0.4294	0.4719	0.8461	0.3850	0.6168
	FMI	**1.0**	0.6075	0.9458	0.5946	0.5956	0.6544	0.9132	0.5651	0.7345
	Arg-	6	0.15	0.28/6	3	0.265	10	0.3	0.002	-
R15	NMI	**0.9942**	0.9578	0.9150	0.8782	0.9647	0.9364	0.9571	0.9546	0.9448
	RI	**0.9991**	0.9912	0.9821	0.9630	0.9936	0.9826	0.9912	0.9912	0.9868
	ARI	**0.9928**	0.9285	0.8510	0.7374	0.9464	0.9323	0.9283	0.9280	0.9056
	AMI	**0.9938**	0.9549	0.9086	0.8703	0.9610	0.8663	0.9540	0.9514	0.9325
	FMI	**0.9932**	0.9332	0.8610	0.7637	0.9506	0.8769	0.9330	0.9328	0.9056
	Arg-	15	0.11	0.11/8	15	0.061	18	0.5	0.08	-

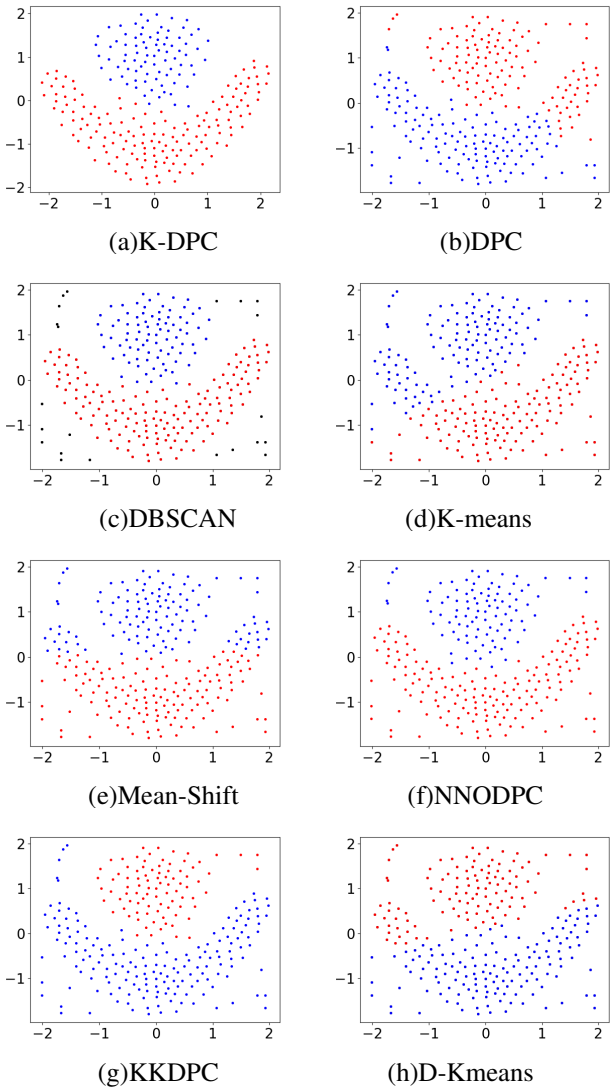


Fig. 10. Clustering results on the Flame dataset with noise

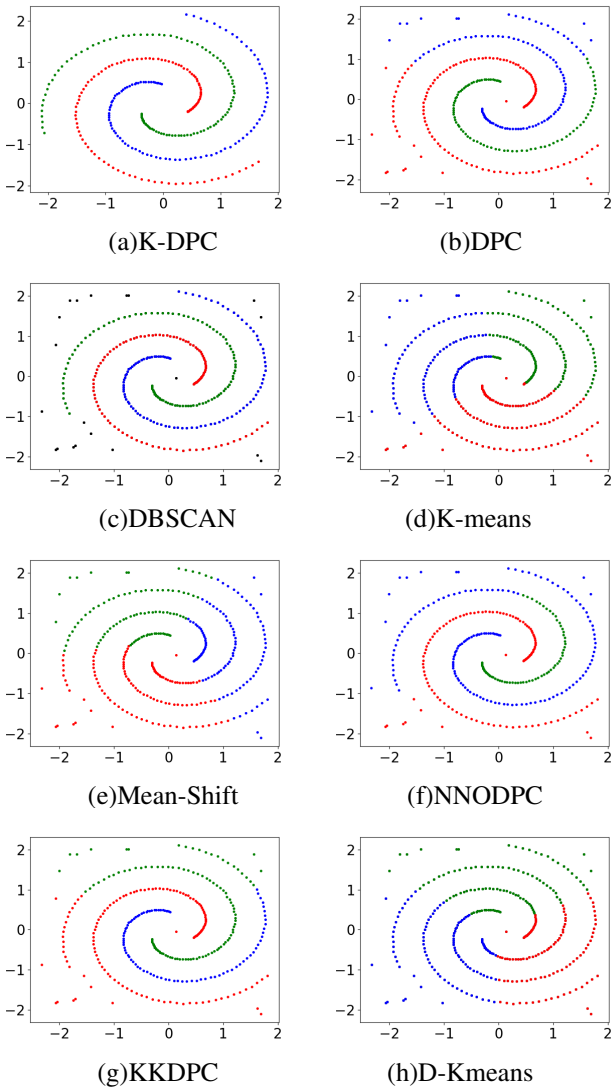


Fig. 11. Clustering results on Spiral dataset with noise

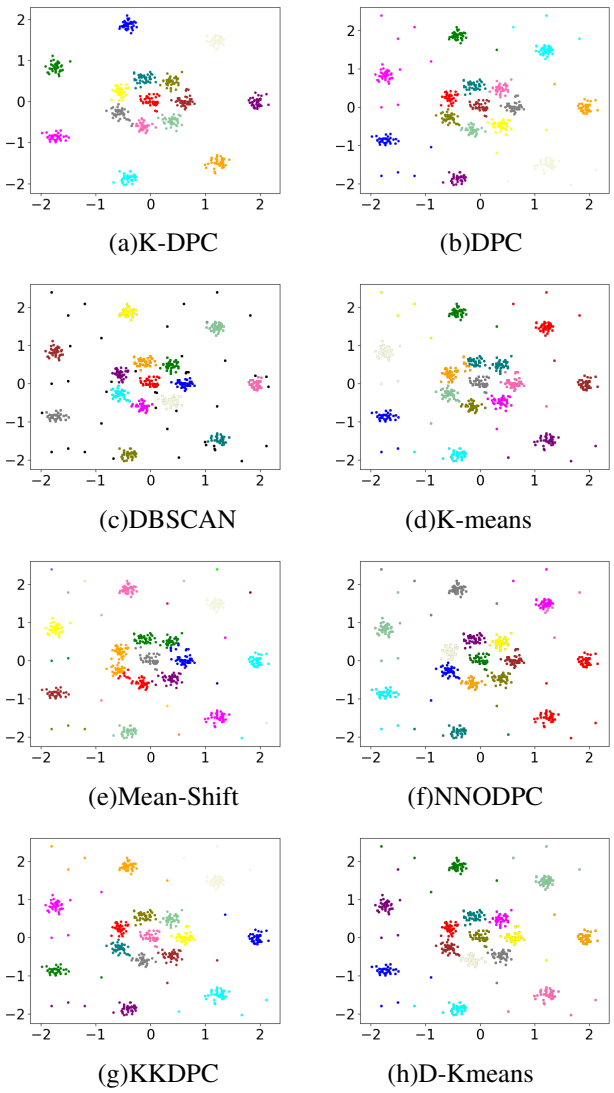


Fig. 12. Clustering results on R15 dataset with noise

The above analysis demonstrates that the K-DPC algorithm exhibits high robustness against noisy data. When applied to datasets with artificially added noise, K-DPC successfully excluded all 20 newly introduced noise points. The algorithm also effectively removes inherent noise in the Flame dataset. Furthermore, in noisy datasets, K-DPC outperforms classical clustering algorithms—including the traditional DPC, DBSCAN, and several improved methods—across evaluation metrics: NMI, RI, ARI, AMI, and FMI.

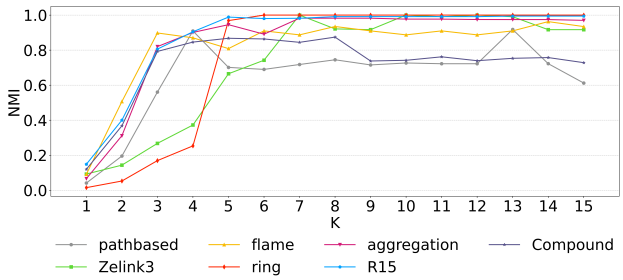


Fig. 13. Clustering effect

D. Parametric analysis

The K-DPC algorithm requires pre-setting parameter k, which defines the number of neighbors for each data point. This parameter is critical for local density calculation, cluster center identification, and data point assignment, directly influencing clustering performance. In this section, select datasets are analyzed experimentally with k values ranging from 1 to 15. An upper bound of 15 is set because larger k values entail higher computational costs while yielding marginal improvements in clustering results. Figure 5 illustrates how varying k values affect K-DPC's clustering performance across the Zelnik3, Flame, R15, Ring, Aggregation, Compound, and Pathbased datasets.

As shown in Figure 13, the K-DPC algorithm achieves optimal clustering performance across target datasets when parameter k is set within the range of 5 to 15. Within this interval, evaluation metrics for these datasets exhibit significant stability. The figure reveals a clear upward trend in metrics for all datasets when (k ¡ 5), which can be attributed to the fact that larger k values enable each data point to integrate more comprehensive k-nearest-neighbor information. Specifically, incorporating more neighboring points into local density calculations and point-assignment processes allows the algorithm to more accurately identify data point interrelationships and distribution characteristics, thereby substantially enhancing clustering accuracy.

These experimental results clearly demonstrate that within a reasonable parameter range, the k-DPC algorithm exhibits low sensitivity to k. In other words, even as k fluctuates within a defined interval, the algorithm consistently yields stable and reliable clustering outcomes, providing strong validation of its effectiveness and robustness in practical applications.

V. Conclusion

This paper addresses the limitations of the DPC algorithm by proposing an enhanced K-DPC clustering algorithm. The novel approach integrates k-nearest neighbor and point assignment strategies to improve clustering performance. During the clustering process, the average distance of a point's k-nearest neighbors and its N-fold standard deviation are calculated to effectively detect and eliminate noise. K-DPC redefines local density by integrating the number of mutual neighbors and the local density of k-nearest neighbors, enhancing the accuracy of initial cluster center determination. Based on this, the principle of direct density reachability among k-nearest neighbors is employed for the preliminary assignment of points. For unassignable data points, cumulative weights are calculated to complete the assignment process. The proposed algorithm mitigates the impact of noise on clustering results, improves the accuracy of initial cluster center identification for density-heterogeneous clusters, and significantly reduces sensitivity to the domino effect.

Several comparative experiments on synthetic, real-world, and noisy datasets demonstrate that the K-DPC algorithm outperforms DPC and other methods across multiple metrics. In noise-free scenarios, K-DPC's mean values for NMI, RI, ARI, AMI, and FMI exceed those of DPC by 0.2113, 0.1426, 0.2672, 0.2202, and 0.1523, respectively—reflecting average improvements of 0.1661, 0.1068, 0.1967, 0.1817, and 0.1338.

On noisy datasets, performance gaps widen: K-DPC's mean metrics outperform DPC by 0.3792 (NMI), 0.1752 (RI), 0.3807 (ARI), 0.3816 (AMI), and 0.2334 (FMI), with average improvements of 0.3296, 0.1516, 0.3411, 0.3356, and 0.2184. These results highlight K-DPC's superior efficiency in handling datasets of diverse sizes and geometries—especially noisy ones—and its enhanced clustering performance relative to traditional and state-of-the-art algorithms.

Looking forward, we plan to further optimize the K-DPC algorithm to address more complex data characteristics and higher-dimensional spaces. Additionally, we will explore its potential in real-time data processing and large-scale dataset scenarios. We also intend to integrate deep learning techniques to enhance the algorithm's adaptability and clustering accuracy in unstructured and dynamic environments.

REFERENCES

[1] Y. Zhang, J. Zhou, Z. H. Deng, et al., "Multi-view fuzzy clustering approach based on medoid invariant constraint," *Ruan Jian Xue Bao (J. Softw)*, vol. 30, no. 2, pp. 282-301, 2019.

[2] R. GeethaRamani and L. Balasubramanian, "Macula segmentation and fovea localization employing image processing and heuristic based clustering for automated retinal screening," *Computer Methods and Programs in Biomedicine*, vol. 160, pp. 153-163, 2018.

[3] G. Liu, Y. Zhang, and A. Wang, "Incorporating adaptive local information into fuzzy clustering for image segmentation," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3990-4000, 2015.

[4] L. Jing, K. Michael, Z. Huang, et al., "An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data," *IEEE Transactions on Knowledge & Data Engineering*, 2007, DOI:10.1109/FKDE.2007.1048.

[5] J. Yuan, X. Li, Y. Qin, et al., "Research on Dynamic Ride and Drop-off Site Setting for Customized Passenger Transport Based on Spatial-temporal Clustering," *Engineering Letters*, vol. 33, no. 5, 2025.

[6] X. Li, Y. Ma, H. Zhong, et al., "A Novel Clustering Method for PV Power Curve Patterns based on Multidimensional Feature, Entropy Weight, and K-means," *Engineering Letters*, vol. 33, no. 4, 2025.

[7] Y. Shen, Y. Ma, H. Zhong, et al., "DTW-based Adaptive K-means Algorithm for Electricity Consumption Pattern Recognition," *Engineering Letters*, vol. 33, no. 1, 2025.

[8] L. Sun, X. Qin, W. Ding, et al., "Density peaks clustering based on k-nearest neighbors and self-recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 1913-1938, 2021.

[9] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, Berkeley, CA: University of California Press, 1967, pp. 281-298.

[10] H. S. Park and C. H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336-3341, 2009.

[11] W. Wang, J. Yang, and R. Muntz, "STING: A statistical information grid approach to spatial data mining," in *VLDB*, 1997, pp. 186-195.

[12] M. Ester, H. P. Kriegel, J. Sander, et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*, 1996, pp. 226-231.

[13] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492-1496, 2014.

[14] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32-40, 1975.

[15] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," *ACM Sigmod Record*, vol. 27, no. 2, pp. 73-84, 1998.

[16] M. Du, S. Ding, and H. Jia, "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," *Knowledge-Based Systems*, vol. 99, pp. 135-145, 2016.

[17] J. Y. Xie, H. C. Gao, W. X. Xie, X. H. Liu, and P. W. Grant, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors," *Information Sciences*, vol. 354, pp. 19-40, 2016.

[18] Y. Yaohui, M. Zhengming, and Y. Fang, "Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy," *Knowledge-Based Systems*, vol. 133, pp. 208-220, Oct. 2017.

[19] R. Liu, H. Wang, and X. Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *Information Sciences*, vol. 450, pp. 200-226, 2018.

[20] A. J. Bai, "K-means algorithm for optimizing initial cluster centers based on improved density peaks," *Shenyang University of Technology*, 2024 [Online]. Available: [Accessed 21 August 2024].

[21] D. Jiang, W. Zang, R. Sun, et al., "Adaptive density peaks clustering based on K-nearest neighbor and Gini coefficient," *IEEE Access*, vol. 8, pp. 113900-113917, 2020.

[22] X. Yuan, H. Yu, J. Liang, et al., "A novel density peaks clustering algorithm based on K nearest neighbors with adaptive merging strategy," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 10, pp. 2825-2841, 2021.

[23] W. C. Chen, J. Zhao, and R. B. **ao, "Density Peaks Clustering Algorithm With Nearest Neighbor Optimization for Data With Uneven Density Distribution," *Control. Decis*, vol. 39, no. 3, pp. 919-928, 2024.

[24] Y. ZHOU, H. XIA, H. LIU, et al., "DPC clustering algorithm based on K-reciprocal Neighbors and kernel density estimation," *Journal of Beihang University*, 2023.

[25] Z. He, X. Xu, and S. Deng, "k-ANMI: A mutual information based clustering algorithm for categorical data," *Information Fusion*, vol. 9, no. 2, pp. 223-233, 2008.

[26] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: is a correction for chance necessary?," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 1073-1080.

[27] S. Ding, C. Li, X. Xu, et al., "A sampling-based density peaks clustering algorithm for large-scale data," *Pattern Recognition*, vol. 136, p. 109238, 2023.

[28] H. Chang and D. Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 191-203, 2008.

[29] L. Fu and E. Medico, "FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data," *BMC Bioinformatics*, vol. 8, p. 1, 2007.

[30] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems*, 2004, vol. 17.

[31] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 4, 2007.

[32] A. K. Jain and M. H. C. Law, "Data clustering: A user's dilemma," in *Pattern Recognition and Machine Intelligence: First International Conference, PReMI 2005, Kolkata, India, December 20-22, 2005. Proceedings 1*, Berlin, Heidelberg: Springer, 2005, pp. 1-10.

[33] F. Ma, T. Gong, F. Yang, et al., "Peak clustering algorithm of grid density based on Zipf distribution," [Online]. Available: [Accessed 21 August 2024].

[34] K. Bache and M. Lichman, "UCI machine learning repository," 2020 [Online]. Available: http://archive.ics.uci.edu/ml.

[35] J. Y. XIE, H. C. GAO, and W. X. XIE, "K-nearest neighbors optimized clustering algorithm by fast search and finding the density peaks of a dataset," *Scientia Sinica Informationis*, vol. 46, no. 2, pp. 258-280, 2016.

[36] H. Chang and D. Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 191-203, 2008.