ASYNCHRONOUS CPU-GPU DENSITY PEAK CLUSTERING VIA GRANULAR-BALL REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Density Peak Clustering (DPC) relies on extensive distance computations, leading to high time and memory costs when applied to large-scale data. Although existing granular-ball methods reduce the computational scale through coarse-grained representation, their scalability is still limited by the quality of granular-ball construction and the sequential execution of subsequent computations. To address these issues, we propose an incremental CPU-GPU parallel density peak clustering method based on granular-ball representation, termed ICG-DPC. First, the data space is partitioned into local regions, where granular balls are adaptively refined according to local density and structural dispersion. The balls are further split along the principal direction of the data distribution to better preserve complex spatial structures, thereby compressing the original data into a much smaller set of granular balls. Then, an asynchronous CPU-GPU collaborative framework is developed, in which the CPU dynamically performs granular-ball refinement and structural maintenance, while the GPU parallelizes local neighborhood construction, density estimation, and density-peak computation. A version-consistency mechanism is further employed to coordinate dynamic structural updates with parallel computation. Experimental results on large-scale synthetic and real-world datasets demonstrate that ICG-DPC maintains competitive clustering performance while providing favorable computational efficiency and scalability.

1 INTRODUCTION

Clustering is a fundamental unsupervised learning technique for discovering intrinsic structures in unlabeled data and has been widely applied in image analysis, information retrieval, social networks, and bioinformatics Jain et al. (1999); Wang et al. (2024). With the rapid growth of data volume, modern clustering tasks increasingly involve millions of samples, making computational scalability a critical requirement.

Density Peak Clustering (DPC) identifies cluster centers according to local density and the relative distance to higher-density samples Rodriguez & Laio (2014). Although DPC can detect clusters with complex structures without iterative centroid updating, its pairwise distance computation incurs $O(n^2)$ computational and memory costsRodriguez & Laio (2014); Chen et al. (2020). Existing improvements, including local-neighborhood search, spatial indexingChen et al. (2020), samplingDing et al. (2023), and GPU accelerationLiu et al. (2023), can reduce this burden. However, most still operate at the sample level, so their computational scale remains strongly dependent on the original data size.

Granular-ball computing (GBC) offers a coarse-grained alternative by representing n samples with a much smaller set of K granular balls, where $K \ll n$ Xia et al. (2019). This representation transfers subsequent computation from the sample level to the granule level and has shown strong potential for large-scale learning. Granular-ball-based DPC further reduces conventional DPC costs by performing density-peak computation on granular balls rather than individual samplesCheng et al. (2023).

Nevertheless, two limitations remain. First, clustering quality depends strongly on granular-ball construction, and conventional partitioning strategies may inadequately represent elongated, anisotropic, or heterogeneous-density structures Zhang et al. (2025). Second, existing granular-ball-

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

based DPC methods are mainly executed sequentially, and GPU acceleration has not been effectively coordinated with dynamic granular-ball refinement.

To address these issues, we propose ICG-DPC, an asynchronous CPU-GPU density peak clustering method based on granular-ball representation. The method adaptively refines granular balls using local density, structural dispersion, and principal-direction splitting to improve representation quality while reducing the computational scale. An asynchronous heterogeneous framework is further developed in which the CPU performs structural refinement and maintenance, while the GPU executes neighborhood construction, density estimation, and density-peak computation in parallel. This design enables structure-dependent and data-parallel operations to proceed concurrently.

The main contributions of this work are summarized as follows:

1. We develop a structure-aware granular-ball refinement strategy that jointly considers local density and structural dispersion. A principal-direction-based splitting mechanism is further introduced to preserve complex and anisotropic spatial structures while maintaining a compact representation.
2. We develop an asynchronous CPU-GPU framework for large-scale density peak clustering. A producer-consumer mechanism enables CPU-side granular-ball refinement and GPU-side density-peak computation to proceed concurrently, while version consistency ensures reliable coordination under dynamic structural updates.

2 Related Work

2.1 Granular-Ball-Based Density Peak Clustering

Density Peak Clustering (DPC) identifies cluster centers according to local density and the relative distance to higher-density samples **?**. Although DPC can detect clusters with complex structures without iterative centroid updating, its pairwise distance computation incurs quadratic computational and memory costs, which limits its applicability to large-scale data. Existing improvements reduce this burden through local neighborhood search, spatial indexing, sampling, and parallel computation Chen et al. (2020). However, most of these methods still operate at the sample level, and their computational scale therefore remains closely related to the original data size.

Granular-ball computing (GBC) provides an alternative by representing a large number of samples with a smaller set of granular balls **?**. By transferring distance and statistical computations from the sample level to the granule level, GBC can substantially reduce the effective computational scale while preserving major structural information. This idea has been incorporated into DPC, where granular-ball-level density and relative-distance calculations are used to reduce the computational burden of conventional DPC **?**. Subsequent studies further introduced adaptive granular-ball construction and neighborhood-based density estimation to improve clustering performance **??**.

Nevertheless, the effectiveness of granular-ball-based DPC still depends strongly on the quality of granular-ball construction, especially for data with heterogeneous densities, elongated distributions, or anisotropic local structures. In addition, density estimation and density-peak computation are generally performed after granular-ball construction in a sequential manner, leaving further room for computational acceleration.

2.2 GPU-Accelerated and Heterogeneous Clustering

GPU acceleration has been widely used to improve the efficiency of clustering algorithms because distance calculation, neighborhood construction, and kernel-based density estimation exhibit strong data parallelism. In DPC, GPU-based methods mainly accelerate distance computation and density-peak evaluation by parallelizing these numerical operations **?**. Spatial indexing and hierarchical computation have also been combined with GPU processing to further improve scalability on large datasets.

However, most GPU-accelerated DPC methods primarily focus on numerical kernels Ma et al. (2024), while CPU-side preprocessing and GPU-side computation are still organized as successive stages. This limits the overlap between structural processing and numerical computation. Granular-

ball-based DPC naturally contains heterogeneous workloads: granular-ball refinement and spatial index maintenance involve dynamic updates and irregular control flows, whereas neighborhood search, density estimation, and peak evaluation are highly data-parallel.

These characteristics motivate a coordinated CPU-GPU execution framework in which structure-dependent operations are handled by the CPU and density-related numerical operations are processed by the GPU. Based on this observation, ICG-DPC combines structure-aware granular-ball refinement with asynchronous CPU-GPU density peak computation to reduce the effective computational scale while improving hardware utilization for large-scale clustering Conway et al. (2025).

3 METHOD

3.1 STRUCTURE-AWARE GRANULAR-BALL REPRESENTATION AND INITIALIZATION

Consider a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$. Directly initializing all samples as a single granular ball is inefficient for large-scale data and may result in an overly coarse representation. Therefore, a KD-tree is first used to partition the data space into M non-overlapping local regions, computed as follows,

$$\mathbf{X} = \bigcup_{j=1}^{M} \mathbf{X}_j, \quad \mathbf{X}_j \cap \mathbf{X}_t = \varnothing,\ j \neq t, \tag{1}$$

where $\mathbf{X}_j$ denotes the sample set in the j-th spatial region, $n_j = |\mathbf{X}_j|$ is the number of samples contained therein.

Within each non-empty region $\mathbf{X}_j$, m_j representative samples are uniformly drawn without replacement to construct the initial regional representation. Each sample therefore has an equal inclusion probability of m_j/n_j. The formal definition of the representative subset is given in Appendix A.1.

To balance distribution preservation and computational efficiency, the sampling strategy is determined by the dataset size. For $n \leq 10^5$, all samples are retained, i.e., $m_j = n_j$. For $n > 10^5$, uniform sampling is performed independently within each KD-tree region, with the number of sampled points allocated to each region is calculated as follows,

$$m_j = \max\left(1, \lfloor \alpha \cdot n_j \rfloor\right), \tag{2}$$

where n_j denotes the number of data points in the j-th region, m_j denotes the corresponding number of sampled points, and $\alpha \in (0, 1)$ represents the sampling ratio.

The representative subset $\mathbf{R}_j$ is then used to initialize a granular ball $g_j = (\mathbf{R}_j, w_j)$ for each spatial region, where the initial weight is defined as $w_j = n_j = |\mathbf{X}_j|$. Thus, although the geometric structure of the granular ball is estimated from the representative subset, its weight retains the sample-scale information of the original region. The aggregation of regional representative subsets is formally defined in Appendix A.2.

During subsequent refinement, granular-ball splitting is performed only on the representative samples to avoid repeatedly accessing the complete original dataset. When a parent granular ball g_s is divided into two child balls $g_s^{(1)}$ and $g_s^{(2)}$, its weight is distributed to the two children in proportion to the numbers of representative samples assigned to them. Consequently, the total statistical weight is preserved throughout refinement, i.e., $w_s^{(1)} + w_s^{(2)} = w_s$.

This initialization and weight-propagation strategy preserves coarse spatial structure and sample-scale information while reducing the construction overhead for large-scale datasets. Moreover, splitting can be performed independently across spatial regions, facilitating scalable and asynchronous CPU-GPU processing.

For each granular ball $g_k = (\mathbf{R}_k, w_k)$, its structure is characterized by the center $\mathbf{c}_k$, RMS radius r_k, and weight w_k, which describe its spatial position, internal dispersion, and represented sample mass, respectively. The center and radius are estimated from the representative subset $\mathbf{R}_k$, whereas the weight preserves the statistical contribution of the original data through recursive weight propagation. Their detailed definitions and update rules are provided in Appendix A.3.

3.2 STRUCTURE-AWARE GRANULAR-BALL SPLITTING

Since the initialized granular balls provide only a coarse representation of the data distribution, some may contain samples from different intrinsic clusters. Therefore, a structure-aware refinement strategy is employed to adaptively determine whether a granular ball should be further split according to its internal spatial dispersion and local density.

To characterize the internal spatial dispersion of a granular ball, we define the structural dispersion degree S_k using the mean absolute deviation (MAD) from its center. Unlike the RMS radius r_k, which emphasizes larger deviations through squared distances, S_k reflects the typical deviation of samples from the center. Their ratio $S_k/r_k \in (0, 1]$ further characterizes the internal dispersion pattern and is used to guide subsequent granular-ball refinement. The structural dispersion degree is computed as follows,

$$S_k = \frac{1}{|g_k|} \sum_{\mathbf{x}_i \in g_k} |\mathbf{x}_i - \mathbf{c}_k|. \tag{3}$$

Let K denote the current number of granular balls. To quantify granular-ball compactness, a dimension-independent variance-normalized density is defined as follows,

$$\rho_k = \frac{w_k}{r_k^2 + \epsilon}, \tag{4}$$

where $\epsilon = 10^{-8}$ avoids division by zero. The average density over all granular balls is used as a global reference to guide adaptive splitting according to the relative density of each granular ball.

Since density information is informative for heterogeneous-density regions but less discriminative when granular-ball densities are similar, two complementary splitting criteria are adopted: a density-guided adaptive threshold and a structure-ratio criterion for approximately uniform-density cases. Both use the same PCA-based splitting operator described below.

For heterogeneous-density distributions, the internal dispersion S_k and relative density are jointly used to define the adaptive threshold as follows,

$$T_r = \eta \cdot S_k \cdot \sqrt{\frac{\rho_{\text{global}}}{\rho_k}}, \tag{5}$$

where $\eta \in [0.8, 3.0]$ controls the refinement granularity. A granular ball is further split when $r_k > T_r$, and the ball with the largest $r_k - T_r$ is prioritized at each refinement iteration.

When granular-ball densities are approximately uniform, i.e., $\rho_k \approx \rho_{\text{global}}$, the density-based adjustment becomes less discriminative with respect to structural differences. Therefore, the relative internal dispersion of each granular ball is additionally considered for structural refinement. Granular balls with relatively large internal dispersion are further refined. Combined with the density-guided mechanism, this strategy enables effective granular-ball refinement under both heterogeneous- and approximately uniform-density distributions.

Once a granular ball is selected for refinement, it is split using a PCA-guided operator. Specifically, the dominant direction is determined from the principal eigenvector of the local covariance matrix, and samples are projected onto this direction. The granular ball is then divided into two subsets according to the median projection value. Detailed formulations of the covariance estimation, projection, and median-based partition are provided in Appendix A.4.

By splitting along the principal direction determined by the covariance structure, the proposed operator better adapts to anisotropic data distributions and avoids the randomness associated with k-means ($k = 2$) initialization, leading to more stable granular-ball refinement.

Each refinement step divides one granular ball into two non-empty sub-balls, and the process continues until no ball satisfies the splitting criterion. Since each granular ball contains at least one sample, the number of balls is bounded by n, ensuring termination after at most $n - 1$ splits. As refinement proceeds, the resulting granular balls generally become more compact, facilitating convergence of the splitting process.

To avoid redundant computation, descriptors of unchanged granular balls are directly reused, while only newly generated granular balls are updated after each split. Specifically, when a granular ball

g_s is divided into $g_s^{(1)}$ and $g_s^{(2)}$, the center, RMS radius, weight, and density of the two child balls are recomputed locally, while all unaffected granular balls retain their cached $(\mathbf{c}_k, r_k, w_k, \rho_k)$. The detailed update formulas are provided in Appendix **??**.

The global average density is maintained incrementally. Let S_ρ denote the accumulated density of all current granular balls as follows,

$$S_\rho = \sum_{k=1}^{K} \rho_k. \tag{6}$$

After splitting g_s, it is updated as follows,

$$S'_\rho = S_\rho - \rho_s + \rho_s^{(1)} + \rho_s^{(2)}, \tag{7}$$

$$\rho'_{\text{global}} = \frac{S'_\rho}{K+1}. \tag{8}$$

To maintain the evolving granular-ball structure, a KD-tree is constructed over the center set $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$. When g_s is split, its node is replaced by those of $g_s^{(1)}$ and $g_s^{(2)}$, avoiding reconstruction of the entire spatial index. During refinement, the CPU identifies the affected regions $\mathcal{R}_{\text{changed}}$ and asynchronously pushes them into the task queue $\mathcal{Q}$ while continuing structural updates. After convergence, a stable granular-ball set $\mathcal{G} = \{g_1, g_2, \ldots, g_K\}$ with $K \ll n$ and descriptors $\{(\mathbf{c}_k, r_k, w_k, \rho_k)\}_{k=1}^{K}$ is obtained and passed to the parallel density-peak computation stage.

3.3 Asynchronous CPU-GPU Parallel Density Peak Computation

After the structure-aware refinement, a compact granular-ball representation $\mathcal{G} = \{g_1, g_2, \ldots, g_K\}$ is obtained. Density peak computation is then performed to estimate the local density of granular balls and identify representative density centers. Since neighborhood construction and distance-based kernel evaluation become computationally expensive for large-scale datasets, an asynchronous CPU-GPU framework is introduced. Specifically, the CPU handles structure-dependent operations, including granular-ball refinement, statistical descriptor updating, and spatial structure maintenance, which involve dynamic data changes and irregular control flows. In contrast, the GPU performs highly parallel density peak computation, including distance calculation, kernel density estimation, and peak score evaluation.

The framework follows a producer-consumer architecture based on a single-producer single-consumer (SPSC) queue $\mathcal{Q}$. The CPU generates regional density-computation tasks, each associated with a spatial region and its structural version, and inserts them into $\mathcal{Q}$. It then continues structural refinement without waiting for GPU completion. Meanwhile, the GPU retrieves available tasks and performs local density peak computation. This asynchronous execution allows CPU-side structural refinement and GPU-side density computation to proceed concurrently. The formal task representation is provided in Appendix A.5.

To improve GPU load balance, the granular-ball space is partitioned into independent spatial regions before density computation. Specifically, the KD-tree constructed on granular-ball centers is traversed to generate candidate regions. Small regions are merged with neighboring ones, while oversized regions are recursively divided as follows,

$$N_{\min} \leq |\mathcal{R}_p| \leq N_{\max}, \tag{9}$$

where $N_{\min}$ and $N_{\max}$ control the computational granularity of GPU tasks.

The resulting regions are non-overlapping and collectively cover the complete granular-ball set, their formal definition is given in Appendix A.5. This controlled partition helps reduce workload imbalance among GPU tasks.

For each spatial region $\mathcal{R}_p$, the GPU performs density peak computation using the local granular-ball structure. To avoid exhaustive pairwise distance calculations, a local k_n-nearest-neighbor graph is constructed to capture the dominant spatial relationships among nearby granular balls.

Based on the local neighborhood graph, density estimation is restricted to neighboring granular balls, reducing redundant distance computations. To adapt to different regional spatial scales, the kernel bandwidth is determined from the neighborhood distance distribution as follows,

$$\sigma_p = \operatorname{median} d_{ij} \mid j \in \mathcal{N}_{k_n}(i), \tag{10}$$

where d_{ij} denotes the distance between granular balls g_i and g_j. This adaptive bandwidth eliminates the need for a manually specified global scale parameter and adjusts the density estimation to the local compactness of each region.

With the adaptive neighborhood scale, the density of each granular ball is estimated using a weighted Gaussian kernel as follows,

$$\hat{\rho}_i = \sum_{j \in \mathcal{N}_{k_n}(i)} \omega_j \exp\left(-\frac{d_{ij}^2}{\sigma_p^2}\right), \tag{11}$$

where ω_j denotes the number of original samples contained in granular ball g_j. This weighting allows granular balls containing more samples to contribute more strongly to the density estimate, thereby retaining sample-count information after granular-ball compression.

Based on the estimated local densities, the separation of each granular ball from higher-density granular balls is evaluated according to the conventional density peak criterion. The local density and separation are then jointly used to measure peak significance. Granular balls with relatively high peak significance are retained as local density peak candidates for subsequent global center determination. The detailed definitions of the separation and peak significance are provided in Appendix A.6.

Since CPU refinement and GPU density computation proceed asynchronously, the granular-ball structure may change after a task is submitted. To maintain consistency, each task is assigned a version identifier V_p. The CPU updates the global version after each structural change, and the GPU compares the task version with the latest version upon completion. If the version difference exceeds a predefined tolerance, the corresponding region is considered outdated and recomputed. This mechanism preserves consistency between structural refinement and density peak computation during asynchronous execution.

3.4 Boundary Correction, Global Center Selection, and Label Assignment

After region-level density peak computation, the GPU returns the local candidate centers $\mathbf{C}_p^{\mathrm{loc}}$ from each region. Since both density estimation and candidate detection are performed within individual regions, granular balls near region boundaries may suffer from incomplete parent connections due to the absence of cross-region neighborhood information. In addition, local density peaks detected independently from different regions may contain redundant candidates. Therefore, boundary correction and global candidate reevaluation are performed to obtain a globally consistent clustering structure.

For granular balls located near region boundaries, neighboring granular balls from other regions are first examined. A granular ball g_i is treated as a boundary ball when a granular ball from another region lies within the local neighborhood scale of the region containing g_i. For each boundary ball, only its k nearest cross-region neighbors are considered, and their connectivity is measured as follows,

$$S_{ij} = \frac{\hat{\rho}_j}{d_{ij} + \epsilon}. \tag{12}$$

Among the cross-region neighbors with densities higher than that of g_i, the neighbor with the largest connectivity is regarded as a candidate parent. The original parent relation is updated only when the candidate provides stronger connectivity than the current parent; otherwise, the existing relation is retained. The resulting corrected parent set is denoted by $\mathcal{P}^*$. Since parent updates are restricted to higher-density neighbors, this correction restores missing cross-region connections while preserving the density-increasing property of the density-peak forest.

After boundary correction, the local candidate centers produced by all regions are merged into the global candidate set $\mathbf{C}^{\mathrm{loc}}$. Candidate centers from structurally unchanged regions are directly reused,

whereas only those from regions affected by granular-ball splitting or structural updates are recomputed, thereby avoiding unnecessary repeated computation.

To eliminate redundant local density peaks across regions, the aggregated candidates are further reevaluated at the global level. The global kernel scale is first adaptively determined from the pairwise distances among candidate granular-ball centers computed as follows,

$$\sigma_g = \operatorname{median}\left\{d_{ij} \mid i < j,\, \mathbf{c}_i, \mathbf{c}_j \in \mathbf{C}^{\mathrm{loc}}\right\}. \tag{13}$$

The global density of each candidate granular ball is then recomputed as follows,

$$\rho_i^g = \sum_j \omega_j \exp\left(-\frac{d_{ij}^2}{\sigma_g^2}\right). \tag{14}$$

Based on the global densities, the separation and density-peak decision value of each candidate are computed using the same definitions as those adopted in the region-level density peak calculation, with $\hat{\rho}_i$ replaced by ρ_i^g. For the candidate with the highest global density, the separation is defined by the maximum distance within the candidate set. The corresponding definitions are provided in Appendix A. The globally reevaluated candidates are then filtered according to their decision values to obtain the final cluster-center set $\mathbf{C}^*$, whose cardinality determines the estimated number of clusters K_c.

After the final centers are determined, cluster labels are propagated directly through the corrected parent structure $\mathcal{P}^*$ rather than by reconstructing a new parent relation. Local root candidates that are not retained as final centers are reconnected to valid candidates with higher global density so that every parent path eventually reaches a center $c_r \in \mathbf{C}^*$. Each granular ball then inherits the cluster label of the final center reached through the corrected density-peak forest.

Because a small number of boundary granular balls may still be affected by local density fluctuations, a local structural-consistency refinement is further applied after the initial label propagation. The neighboring labels of each granular ball are examined, and its label is corrected when a clear local consensus conflicts with the initially assigned label.

Finally, the granular-ball labels are mapped back to the original samples. All samples contained in the same granular ball inherit the final label assigned to that granular ball.

4 EXPERIMENTS

In this section, ten datasets are used to evaluate the effectiveness and efficiency of the ICG-DPC method. The real world datasets are obtained from publicly available machine learning repositories, including PenDigits, Letter, MNIST, Covertype, SUSY and HIGCS. Synthetic datasets include T4, TB, SF, CG, Flower and TB_10M. Table 1 provide a brief description of all the datasets used in the experiments. Twelve clustering algorithms are selected as baselines for comparison, including k-means, DPC, LSC-K Cai & Chen (2014), LSC-R Cai & Chen (2014), USPEC Huang et al. (2019), GB-USC Cheng et al. (2024) and GB-SC Xie et al. (2023). The experimental results are presented in Table 2, where the best performance for each dataset is highlighted for clarity. Clustering performance is evaluated using ACC Xu et al. (2003)and NMIStrehl & Ghosh (2002), with higher values indicating better results.

4.1 EXPERIMENT ON SYNTHETIC DATASETS

Table 2 compares the clustering performance of ICG-DPC with representative baseline methods on the five synthetic datasets. Overall, ICG-DPC shows a clear advantage in clustering accuracy over the granular-ball-based and large-scale clustering baselines, while maintaining competitive NMI across different data distributions. The improvement is particularly evident on datasets with complex spatial structures, such as CG and Flower, suggesting that the proposed structure-aware granular-ball refinement can preserve informative local geometry more effectively than conventional coarse-grained representations.

Table 1: Details of datasets in the experiments

TYPE	DATASET	OBJECTS	DIMENSIONS	CLASSES
Synthetic	TB	100,000	2	2
	SF	100,000	2	2
	CG	100,000	2	11
	Flower	100,000	2	2
	TB_10M	10,000,000	2	2
Real	Letter	20,000	16	26
	MNIST	70,000	784	10
	SUSY	5,000,000	18	2
	Covertype	581,012	54	7
	HIGGS	11,000,000	28	2

Table 2: ACC and NMI of different algorithms on synthetic and real-world datasets

	TB		**SF**		**CG**		**FLOWER**		**TB_10M**	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
k-means	0.786	0.261	0.649	0.467	0.527	0.589	0.539	0.659	0.750	0.197
DPC	0.915	0.904	0.552	0.609	0.457	0.502	0.407	0.452	0.933	0.927
FCAG	–	–	–	–	–	–	–	–	–	–
LSC-R	0.519	0.160	0.704	0.676	0.693	0.785	0.640	0.796	–	–
LSC-K	0.537	0.600	0.689	0.657	0.677	0.775	0.678	0.788	–	–
USPEC	**0.995**	0.956	0.791	0.765	0.635	0.783	0.842	0.876	0.911	0.937
GB-USC	0.896	0.863	0.793	0.811	0.680	**0.815**	0.856	**0.894**	0.891	0.890
GB-SC	–	–	–	–	–	–	–	–	–	–
ICG-DPC	0.966	**0.958**	**0.814**	**0.796**	**0.809**	0.706	**0.910**	0.878	**0.986**	**0.995**

	LETTER		**MNIST**		**SUSY**		**COVERTYPE**		**HIGGS**	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
k-means	0.255	0.333	0.458	0.405	0.562	0.216	0.239	0.075	0.612	0.251
DPC	0.359	0.458	0.365	0.404	0.396	0.282	0.527	0.318	0.593	0.339
FCAG	0.316	0.432	0.669	0.673	0.420	0.335	–	–	0.519	0.473
LSC-R	0.298	0.396	0.754	0.729	0.554	0.318	0.227	0.086	0.511	0.293
LSC-K	0.317	0.427	**0.760**	**0.731**	0.546	0.241	0.302	0.107	0.602	0.299
USPEC	0.315	0.422	0.698	0.683	0.531	0.359	0.243	0.086	0.583	0.377
GB-USC	0.306	0.425	0.538	0.653	0.635	0.302	0.255	0.085	0.505	0.335
GB-SC	0.128	0.184	–	–	0.540	0.370	–	–	0.511	0.279
ICG-DPC	**0.446**	**0.534**	0.743	0.695	**0.692**	**0.865**	**0.860**	**0.373**	**0.971**	**0.935**

Compared with GB-USC and USPEC, ICG-DPC achieves higher average ACC while maintaining a comparable or slightly better average NMI. This indicates that the proposed method improves cluster assignment accuracy without sacrificing the consistency of the discovered partition structure. The performance on TB_10M further shows that this advantage can be retained when the data scale increases to the order of millions, supporting the scalability of the proposed granular-ball representation and asynchronous computation framework. Nevertheless, ICG-DPC does not dominate all evaluation metrics; several baselines achieve better results on individual datasets, particularly in NMI. These results suggest that ICG-DPC provides a favorable balance between clustering quality and adaptability across different synthetic structures rather than being uniformly optimal for every distribution.

Table 3: Running time of different algorithms on synthetic and real-world datasets (s)

	TB	SF	CG	FLOWER	TB_10M
k-means	0.47	0.60	0.78	0.81	4.27
DPC	0.59	0.71	0.79	1.62	–
FCAG	–	–	–	–	–
LSC-R	22.32	30.46	41.91	53.07	–
LSC-K	37.24	50.40	65.87	82.00	–
USPEC	9.30	11.36	24.96	25.98	113.60
GB-USC	3.56	5.95	22.38	20.19	43.99
GB-SC	–	–	–	–	–
ICG-DPC	3.19	4.69	9.51	17.44	45.71

	LETTER	MNIST	SUSY	COVERTYPE	HIGGS
k-means	0.02	0.38	2.71	0.56	7.19
DPC	0.22	0.51	–	3.19	–
FCAG	1.38	36.90	60.00	–	192.10
LSC-R	4.69	4.61	53.11	32.51	162.00
LSC-K	2.41	6.75	58.20	38.97	170.91
USPEC	1.29	4.14	107.24	12.11	188.02
GB-USC	1.26	16.09	73.68	17.16	131.33
GB-SC	71.33	–	182.50	–	–
ICG-DPC	2.49	6.81	17.16	16.05	25.97

4.2 Experiment on Real-World Datasets

Table 2 further compares the methods on five real-world datasets with different dimensionalities and data scales. Compared with GB-USC and USPEC, ICG-DPC achieves substantially better average performance in both ACC and NMI, indicating that the proposed representation is effective not only for synthetic spatial structures but also for more heterogeneous real-world data. The advantage is particularly pronounced on Covertype and HIGGS, where ICG-DPC provides markedly better clustering results than the competing granular-ball and large-scale clustering methods.

The improvement on these datasets suggests that the combination of adaptive granular-ball refinement and density-peak computation is effective in retaining discriminative structure after data compression. In addition, the strong performance on HIGGS, which contains more than ten million samples, demonstrates that reducing sample-level computation to granular-ball-level processing does not necessarily lead to a substantial loss of clustering quality. However, the results on MNIST and SUSY also show that the advantage of ICG-DPC is metric- and dataset-dependent: several baseline methods obtain higher ACC on these datasets, although ICG-DPC remains competitive in terms of NMI. Overall, the real-world experiments demonstrate that ICG-DPC provides a favorable trade-off between clustering effectiveness and large-scale computational applicability across datasets with substantially different characteristics.

5 Conclusion

This paper proposed ICG-DPC, an asynchronous CPU-GPU granular-ball density peak clustering framework for large-scale data. By integrating structure-aware granular-ball refinement, asynchronous computation, and cross-region correction, ICG-DPC achieves compact representation and efficient density-peak clustering. Experiments on synthetic and real-world datasets demonstrate competitive clustering performance and favorable scalability. A current limitation is that regional sampling may lose fine-grained structural information in high-dimensional or highly complex data. Future work will focus on more adaptive representations and extensions to streaming and dynamically evolving data.

AI USE STATEMENT

In this work, generative AI tools were used to assist with language editing, presentation, and organization of parts of the manuscript, including the Introduction, Related Work, and portions of the descriptions of the method and experimental results. Generative AI tools were not used to develop the core methodology, design or conduct the experiments, or generate experimental data or results. All AI-assisted content was reviewed and revised by the authors. The authors take full responsibility for the final content of this work, including its text, research claims, and experimental results.

REFERENCES

Deng Cai and Xinlei Chen. Large scale spectral clustering via landmark-based sparse representation. *IEEE transactions on cybernetics*, 45(8):1669–1680, 2014.

Yewang Chen, Xiaoliang Hu, Wentao Fan, Lianlian Shen, Zheng Zhang, Xin Liu, Jixiang Du, Haibo Li, Yi Chen, and Hailin Li. Fast density peak clustering for large scale data based on knn. *Knowledge-Based Systems*, 187:104824, 2020.

Dongdong Cheng, Ya Li, Shuyin Xia, Guoyin Wang, Jinlong Huang, and Sulan Zhang. A fast granular-ball-based density peaks clustering algorithm for large-scale data. *IEEE Transactions on Neural Networks and Learning Systems*, 35(12):17202–17215, 2023.

Dongdong Cheng, Shushu Liu, Shuyin Xia, and Guoyin Wang. Granular-ball computing-based manifold clustering algorithms for ultra-scalable data. *Expert Systems with Applications*, 247: 123313, 2024.

Alex Conway, Alex Pothen, Martin Farach-Colton, and Bora Ucar. 2025 proceedings of the conference on applied and computational discrete algorithms (acda), 2025.

Shifei Ding, Chao Li, Xiao Xu, Ling Ding, Jian Zhang, Lili Guo, and Tianhao Shi. A sampling-based density peaks clustering algorithm for large-scale data. *Pattern Recognition*, 136:109238, 2023.

Dong Huang, Chang-Dong Wang, Jian-Sheng Wu, Jian-Huang Lai, and Chee-Keong Kwoh. Ultra-scalable spectral clustering and ensemble clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1212–1226, 2019.

Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

Zhuojin Liu, Shufeng Gong, Yuxuan Su, Changyi Wan, Yanfeng Zhang, and Ge Yu. Improving density peaks clustering through gpu acceleration. *Future Generation Computer Systems*, 141: 399–413, 2023.

Luyao Ma, Geping Yang, Yiyang Yang, Xiang Chen, Juan Lu, Zhiguo Gong, and Zhifeng Hao. Up-dpc: Ultra-scalable parallel density peak clustering. *Information Sciences*, 660:120114, 2024.

Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *science*, 344(6191):1492–1496, 2014.

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

Yizhang Wang, Jiaxin Qian, Muhammad Hassan, Xinyu Zhang, Tao Zhang, Chao Yang, Xingxing Zhou, and Fengjin Jia. Density peak clustering algorithms: A review on the decade 2014–2023. *Expert Systems with Applications*, 238:121860, 2024.

Shuyin Xia, Yunsheng Liu, Xin Ding, Guoyin Wang, Hong Yu, and Yuoguo Luo. Granular ball computing classifiers for efficient, scalable and robust learning. *Information Sciences*, 483:136–152, 2019.

Jiang Xie, Weiyu Kong, Shuyin Xia, Guoyin Wang, and Xinbo Gao. An efficient spectral clustering algorithm based on granular-ball. *IEEE Transactions on Knowledge and Data Engineering*, 35 (9):9743–9753, 2023.

Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 267–273. ACM, 2003. doi: 10.1145/860435.860485.

Xingguo Zhang, Li Xu, and Weikuan Jia. Adaptive granular-ball-based density peak clustering. *Neurocomputing*, pp. 131458, 2025.

A METHOD

A.1 REGIONAL REPRESENTATIVE SAMPLING

For each non-empty spatial region $\mathbf{X}_j$, m_j representative samples are uniformly drawn without replacement to form as follows,

$$\mathbf{R}_j = \left\{ \mathbf{z}_1^{(j)}, \mathbf{z}_2^{(j)}, \ldots, \mathbf{z}_{m_j}^{(j)} \right\} \subseteq \mathbf{X}_j, \tag{15}$$

where $\mathbf{z}_t^{(j)}$ denotes the t-th sample drawn from the j-th region. The selection is unbiased, every sample in $\mathbf{X}_j$ has the same probability m_j/n_j of being included in $\mathbf{R}_j$.

A.2 AGGREGATION OF REGIONAL REPRESENTATIVE SUBSETS

After representative sampling, the subsets generated from all KD-tree regions are aggregated as follows,

$$\mathbf{R} = \bigcup_{j=1}^{M} \mathbf{R}_j, \tag{16}$$

with the total number of representative samples computed as follows,

$$|\mathbf{R}| = \sum_{j=1}^{M} m_j. \tag{17}$$

A.3 INCREMENTAL UPDATE OF GRANULAR-BALL DESCRIPTORS

To reduce redundant computation during granular-ball refinement, only newly generated granular balls require descriptor recalculation, whereas unchanged granular balls directly reuse their stored descriptors $(\mathbf{c}_k, r_k, w_k, \rho_k)$.

Suppose that a parent granular ball $g_s = (\mathbf{R}_s, w_s)$ is divided into two child balls $g_s^{(1)} = (\mathbf{R}_s^{(1)}, w_s^{(1)})$ and $g_s^{(2)} = (\mathbf{R}_s^{(2)}, w_s^{(2)})$. For each child granular ball $g_s^{(t)}$, $t \in 1, 2$, its center and RMS radius are recomputed from the corresponding representative subset as follows,

$$\mathbf{c}_s^{(t)} = \frac{1}{|\mathbf{R}_s^{(t)}|} \sum_{\mathbf{x}_i \in \mathbf{R}_s^{(t)}} \mathbf{x}_i, \tag{18}$$

$$r_s^{(t)} = \sqrt{\frac{1}{|\mathbf{R}_s^{(t)}|} \sum_{\mathbf{x}_i \in \mathbf{R}_s^{(t)}} \left\| \mathbf{x}_i - \mathbf{c}_s^{(t)} \right\|_2^2}. \tag{19}$$

Because refinement is performed on the representative samples rather than on all original samples, the parent weight is propagated to the child balls according to their relative representative-sample sizes computed as follows,

$$w_s^{(t)} = w_s \frac{|\mathbf{R}_s^{(t)}|}{|\mathbf{R}_s|}, \qquad t \in \{1, 2\}. \tag{20}$$

Since

$$|\mathbf{R}_s^{(1)}| + |\mathbf{R}_s^{(2)}| = |\mathbf{R}_s|, \tag{21}$$

the weight is conserved during splitting computed as follows,

$$w_s^{(1)} + w_s^{(2)} = w_s. \tag{22}$$

The density of each newly generated child ball is then updated as follows,

$$\rho_s^{(t)} = \frac{w_s^{(t)}}{(r_s^{(t)})^2 + \epsilon}. \tag{23}$$

Therefore, the geometric descriptors are estimated from the representative samples, while the recursively propagated weights preserve the sample-scale contribution of the original data throughout granular-ball refinement.

A.4 PCA-Guided Granular-Ball Splitting

For a granular ball g_k selected for refinement, its covariance matrix is computed as follows,

$$\mathbf{\Sigma}_k = \frac{1}{|g_k|} \sum_{\mathbf{x}_i \in g_k} (\mathbf{x}_i - \mathbf{c}_k)(\mathbf{x}_i - \mathbf{c}_k)^T. \tag{24}$$

The principal splitting direction $\mathbf{v}_k$ is defined as the eigenvector corresponding to the largest eigenvalue of $\mathbf{\Sigma}_k$. Each sample is then projected onto this direction as $p_i = \mathbf{v}_k^T(\mathbf{x}_i - \mathbf{c}_k)$.

Using the median projected value as the splitting point, the two child granular balls are defined as follows,

$$g_k^{(1)} = \{\mathbf{x}_i \in g_k : p_i \leq \text{median}(p_j)\}, \tag{25}$$

$$g_k^{(2)} = \{\mathbf{x}_i \in g_k : p_i > \text{median}(p_j)\}. \tag{26}$$

This median-based division produces two subsets along the dominant structural direction without requiring iterative binary clustering or random center initialization.

A.5 Regional Task and Partition Definitions

In the asynchronous producer-consumer framework, each GPU task is represented as follows,

$$\mathcal{T}_p = (\mathcal{R}_p, V_p), \tag{27}$$

where $\mathcal{R}_p$ denotes the spatial region assigned to the task and V_p records its corresponding structural version.

The regional partition satisfies computed as follows,

$$\bigcup_{p=1}^{P} \mathcal{R}_p = \mathcal{G}, \quad \mathcal{R}_p \cap \mathcal{R}_q = \varnothing, \quad p \neq q. \tag{28}$$

Thus, all granular balls are assigned to mutually disjoint computational regions.

A.6 Local Density Peak Measures

For each granular ball g_i, its separation from higher-density granular balls is defined as follows,

$$\delta_i = \min_{j:\hat{\rho}_j > \hat{\rho}_i} d_{ij}. \tag{29}$$

where δ_i represents the minimum distance from g_i to any granular ball with a higher local density within the corresponding region. For the highest-density granular ball in a region, where no higher-density neighbor exists, δ_i is set to the maximum distance to the other granular balls within that region.

The peak significance of g_i is subsequently defined as follows,

$$\gamma_i = \hat{\rho}_i \delta_i. \tag{30}$$

A relatively large γ_i indicates that the granular ball has both a high local density and a large spatial separation from higher-density granular balls, making it more likely to represent a local density peak.