International Journal of Geo-Information

MDPI

Article

A Spatial Co-Location Pattern Mining Method Based on Hausdorff Distance Alignment

Xichen Liu [1,2], Yajie Li [1,2] and Muquan Zou [1,2,*]

[1] School of Information Engineering, Kunming University, Kunming 650214, China; liuxichen52@gmail.com (X.L.); objectmars0119@gmail.com (Y.L.)
[2] Yunnan Key Laboratory of Intelligent Logistics Equipment and Systems, Kunming 650214, China
* Correspondence: ynu_zmq@foxmail.com

Abstract

Spatial co-location patterns are used to describe the spatial associations between features, finding wide applications in geographic information systems, urban planning, and other fields. Traditional frameworks for mining spatial features typically consist of two stages: constructing spatial proximity relationships and discovering frequent patterns. However, existing methods have limitations: the construction of proximity relationships relies on fixed distance thresholds or clustering centers, making it difficult to adapt to spatial density heterogeneity; meanwhile, frequency metrics overly depend on participation indices, lacking quantitative analysis of the strength of geometric associations between features. To address these issues, a spatial co-location pattern mining method based on Hausdorff distance is proposed. Drawing on the concept of Hausdorff distance, this method employs Voronoi tessellation to achieve data-adaptive partitioning of the spatial domain. Combined with a K-dimensional tree, it adopts an iterative strategy of direct allocation, proportional allocation, and residual allocation to align instances, generating a spatial proximity relationship graph. Additionally, a new frequency metric based on instance distribution—alignment rate—is introduced, leveraging the decreasing trend of alignment rate in conjunction with a pruning optimization algorithm. Experimental results demonstrate that this method excels in handling noise points, effectively addressing the challenges of uneven data density distribution while enhancing the identification of weakly associated yet potentially valuable patterns.

Keywords: spatial co-location pattern; hausdorff distance; voronoi; k-dimensional tree; alignment rate

check for updates

Academic Editors: Wolfgang Kainz and Maria Antonia Brovelli

Received: 28 June 2025
Revised: 5 August 2025
Accepted: 10 August 2025
Published: 26 August 2025

Citation: Liu, X.; Li, Y.; Zou, M. A Spatial Co-Location Pattern Mining Method Based on Hausdorff Distance Alignment. *ISPRS Int. J. Geo-Inf.* **2025**, *14*, 331. https://doi.org/10.3390/ijgi14090331

Copyright: © 2025 by the authors. Published by MDPI on behalf of the International Society for Photogrammetry and Remote Sensing. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Spatial co-location pattern mining uncovers co-occurrence patterns among diverse features in spatial data, providing data-driven insights for decision-making. Advances in data collection and storage have generated vast spatial datasets rich with information. This approach analyzes frequent feature co-occurrences to reveal synergistic distributions in geographic space [1]. For instance, in urban planning, frequent co-occurrences of hospitals and pharmacies may indicate complementary facility functions, optimizing urban layouts. Typically, spatial co-location pattern mining involves two phases: constructing proximity relationships and quantifying their frequency.

Conventional methods rely on fixed distance thresholds to define proximity and participation ratios to measure frequency [2]. However, these approaches face several limitations.

First, fixed thresholds are sensitive to noise and uneven distributions, often misidentifying noise as proximate or missing genuine relationships. Second, participation ratios are susceptible to outliers, where extreme values may inflate results, obscuring true pattern strength. Third, these methods exhibit low efficiency on large-scale datasets, limiting practical applicability. Prior studies have enhanced proximity generation using density-based clustering [3] or tree-based algorithms [4,5], yet they remain threshold-dependent. Efforts to improve frequency measurement with weighted or composite metrics [6] increase complexity and reduce interpretability. Efficiency improvements via graph structures [7] or indexing [8] still encounter computational bottlenecks. Despite these advances, significant opportunities remain to enhance accuracy, robustness, and applicability.

This study proposes a threshold-free, robust, and efficient method for spatial co-location pattern mining, addressing the aforementioned challenges with the following contributions:

1. Proximity Relationship Generation : Conventional methods use fixed distance thresholds or single k-Nearest Neighbor (k-NN) algorithms, which are sensitive to noise and density variations, hindering the capture of complex spatial structures. The proposed method employs Voronoi tessellation to delineate distribution boundaries and constructs KD-trees for feature subsets in each region, eliminating threshold reliance. A dynamic k-NN algorithm with a three-stage allocation strategy (direct, proportional, and residual) adaptively aligns instances across sparse and dense regions. Drawing on the Hausdorff distance, this approach evaluates point set similarity, ensuring accurate and robust proximity relationships. Local relationships are refined by converting directed edges to undirected edges and applying a transfer substitution strategy, maintaining robustness under noise or density fluctuations, effectively filtering noise while preserving true patterns.
2. Frequency Measurement: Traditional participation ratios overlook synergistic feature alignment, making them prone to distortions from local anomalies. The proposed alignment ratio, a novel metric measuring the proportion of aligned feature instances, offers a comprehensive frequency measure. A cumulative multiplication strategy captures k-order pattern effects, preserving global information and mitigating anomaly distortions, thus providing an intuitive and accurate representation of associations.

The rest of the paper is organized as follows: Section 4 details the proposed method, Section 5 presents experimental results, and Section 6 concludes the study with future work.

2. Related Work

Spatial co-location pattern mining has been extensively studied, with the participation ratio proposed by Huang et al. [9] as a foundational metric for pattern frequency. Its anti-monotonicity reduces the pattern search space, enhancing mining efficiency. Subsequent refinements led to algorithms such as the Join-based Co-location Pattern Mining Algorithm (Join-based) and the Partial-join Co-location Pattern Mining Algorithm (Partial-join) [10]. The latter improves efficiency by minimizing join operations. However, the computational overhead of joins limits performance on large-scale datasets. To address this, the Join-less Co-location Pattern Mining Algorithm (Join-less) [11] employs a lookup strategy, significantly boosting performance. Further optimizations include tree-based algorithms like the Co-location Pattern Index Tree (CPI-tree) [12], the Improved CPI-tree (iCPI-tree) [13], and the Order-Clique-Based Co-location Pattern Mining Algorithm [14]. These leverage prefix-sharing to reduce storage redundancy and integrate proximity relationship generation, improving both precision and efficiency.

Traditional methods for generating proximity relationships rely on distance-based metrics, such as Euclidean [15] and Manhattan distances [16], using fixed thresholds. These

approaches struggle with irregularly distributed data or complex topological structures due to their sensitivity to data distribution. Topology-based methods, such as those using Voronoi diagrams [17,18] or Delaunay triangulation [14,19], address these limitations by partitioning space and adapting to its structure, making them suitable for complex spatial data. Network-index-based methods, including k-Nearest Neighbor (k-NN) algorithms [20,21], fuzzy grid clustering [8], and GPU-accelerated approaches [22], enhance efficiency for large-scale datasets through optimized indexing structures. Recent theoretical advances in spatial graph modeling, such as graph neural networks (GNNs) and spatial autoregressive models, provide robust frameworks for capturing complex spatial dependencies. GNNs represent spatial data as graphs, using node and edge features to model intricate relationships across heterogeneous topologies [23]. Spatial autoregressive models incorporate spatial correlation structures to handle dynamic and irregular spatial configurations [24]. These methods offer theoretical rigor for modeling complex spatial interactions but often require intensive computational resources. Our HD-SCPM algorithm complements these approaches by integrating Voronoi tessellation and KD-tree indexing, providing an efficient theoretical framework for proximity relationship generation with reduced computational complexity.

Theoretically, multi-scale spatial pattern mining addresses the challenge of detecting patterns across varying spatial resolutions through frameworks like multi-resolution grid partitioning and hierarchical clustering. Multi-resolution approaches decompose spatial data into nested grids, enabling the identification of patterns at both fine and coarse scales [25]. Hierarchical clustering constructs layered representations of spatial relationships, capturing dependencies across different levels of granularity [26]. Additionally, wavelet-based transformations provide a theoretical basis for analyzing spatial patterns at multiple scales by decomposing data into frequency components [27]. These frameworks are particularly effective for modeling spatial heterogeneity in complex environments. HD-SCPM's alignment ratio metric is theoretically designed to adapt to multi-scale distributions by dynamically adjusting proximity thresholds, ensuring robust pattern detection across diverse spatial granularities without requiring extensive empirical validation.

Drawing on the Hausdorff distance, this work optimizes distance metrics for complex, irregularly shaped datasets, outperforming traditional metrics in noisy or uneven distributions [28,29]. Additionally, a novel alignment ratio replaces conventional frequency metrics, such as the maximum-minimum [30,31] and weighted participation ratios [32]. This metric better captures relationships between spatial features, offering superior accuracy and robustness, particularly for irregular topological structures. The choice of Hausdorff distance is justified by its ability to capture global geometric relationships, making it robust for noisy and irregular datasets like *Shanghai_POI*. Compared to alternative similarity metrics, such as Dynamic Time Warping (DTW), which excels in measuring temporal sequence similarity for spatio-temporal data [33], Hausdorff distance is more efficient for static spatial patterns, with a complexity of $O(n \log n)$ versus DTW's $O(nm)$ for sequences of lengths n and m. For spatio-temporal extensions, HD-SCPM could potentially integrate DTW to analyze dynamic urban patterns, such as time-varying POI distributions, but this would require additional temporal data and optimization to manage DTW's higher computational cost. The current focus on static spatial patterns limits the direct applicability of DTW, though future work could explore hybrid metrics to enhance HD-SCPM's temporal robustness. Compared to existing algorithms, HD-SCPM achieves competitive efficiency for large-scale, heterogeneously distributed datasets. Its time complexity, as detailed in Section 4.1 (Equation (25)), is $O(|F|^3 n \log |F| + p|F|)$, where $|F|$ is the average number of instances per feature, n is the number of instances, and p is the number of frequent patterns. This is comparable to Join-less and CPI-tree ($O(|F|n \log n)$) in moderately sized datasets and

outperforms Join-based ($O(|F|^2n^2)$) in scenarios with irregular distributions due to its use of Voronoi tessellation and KD-tree indexing. HD-SCPM excels in multi-scale settings by dynamically adjusting proximity thresholds, unlike the static structures of CPI-tree. However, its scalability is limited in high-dimensional spaces due to KD-tree inefficiencies and in dense datasets where the $|F|^3$ term dominates. These trade-offs make HD-SCPM particularly suitable for urban datasets (e.g., *Shanghai_POI*) with complex topologies and moderate feature density. This work advances spatial co-location pattern mining by integrating Voronoi tessellation, KD-tree indexing, and a novel alignment ratio, achieving superior accuracy and efficiency for complex spatial datasets, while positioning itself at the intersection of spatial graph modeling and multi-scale pattern mining to theoretically address diverse spatial data challenges.

3. Related Definitions

3.1. Basic Concepts

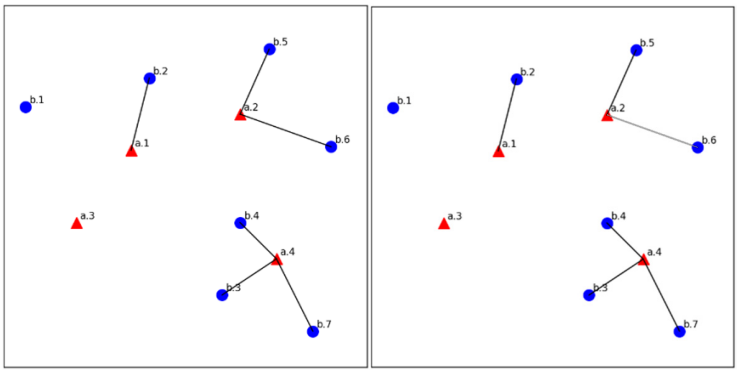
Spatial features represent distinct categories of objects characterized by specific attributes within a spatial domain, such as buildings, rivers, or roads. Let $F = \{f_1, f_2, \ldots, f_n\}$ denote a set of spatial features, where each f_i ($i \in \mathbb{N}, 1 \leq i \leq n$) corresponds to a feature type. Each feature f_i is associated with a set of instances $O_i = \{o_{i1}, o_{i2}, \ldots, o_{im_i}\}$, where each instance o_{ij} ($j \in \mathbb{N}, 1 \leq j \leq m_i$) is positioned at $\mathrm{loc}(o_{ij}) \in \mathbb{R}^d$, with d representing the spatial dimensionality (typically $d = 2$ or 3).

Given two instances $o_{in} \in O_i$ and $o_{jm} \in O_j$, a proximity relationship exists if the distance between their locations satisfies a predefined condition:

$$D(\mathrm{loc}(o_{in}), \mathrm{loc}(o_{jm})) \leq \delta, \tag{1}$$

where $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ is a distance metric (e.g., Euclidean distance), and $\delta \in \mathbb{R}_{>0}$ is a user-defined threshold. This relationship is denoted as $o_{in} \sim o_{jm}$. A set $cl = \{o_1, o_2, \ldots, o_k\} \subseteq \bigcup_{i=1}^{n} O_i$ forms a *spatial clique* if every pair $(o_p, o_q) \in cl \times cl$ (with $p \neq q$) satisfies Equation (1). A spatial clique is *maximal* if it is not contained within any larger clique satisfying the same condition.

For example, in Figure 1a (subfigure (a) of Figure 1), the traditional threshold-based method identifies proximity relationships, such as between instances a_2 and b_6, marked by a black solid line, based solely on the distance threshold δ.



(**a**) Traditional Method (**b**) HD-SCPM Method

Figure 1. Comparison of proximity relationships between traditional threshold-based method and HD-SCPM. (**a**) Traditional method, where instances a_2 and b_6 are connected by a black solid line indicating a proximity relationship. (**b**) HD-SCPM method, where a_2 and b_6 are connected by a light gray solid line, indicating they do not satisfy the bidirectional alignment condition and are not considered neighbors.

A *spatial co-location pattern* $c \subseteq F$ is a subset of features $c = \{f_{i_1}, f_{i_2}, \ldots, f_{i_k}\}$, where k is the pattern's order. A spatial clique cl containing instances of all features in c, with each instance belonging to a distinct feature, is called a *row instance* of c. The collection of all row instances forms the *table instance* $T(c)$. The *participation ratio* for a feature $f_i \in c$ is defined as:

$$\mathrm{PR}(c, f_i) = \frac{|\{o \in O_i \mid \exists cl \in T(c), o \in cl\}|}{|O_i|}, \tag{2}$$

where $|\cdot|$ denotes the cardinality of a set. The *participation index* of pattern c is the minimum participation ratio across all features:

$$\mathrm{PI}(c) = \min_{f_i \in c} \mathrm{PR}(c, f_i). \tag{3}$$

A pattern c is deemed *frequent* if $\mathrm{PI}(c) \geq \theta$, where $\theta \in (0, 1]$ is a user-specified threshold. For example, in Figure 1a, for the pattern $c = \{a, b\}$, the participation ratios are $\mathrm{PR}(c, a) = \frac{3}{4} = 0.75$ and $\mathrm{PR}(c, b) = \frac{6}{7} \approx 0.857$, yielding $\mathrm{PI}(c) = \min(0.75, 0.857) = 0.75$. If $\theta = 0.3$, then c is frequent.

3.2. Instance Alignment Based on Hausdorff Distance

To enhance spatial proximity mining, we propose a unified alignment framework inspired by the Hausdorff distance, enabling adaptive instance pairing across heterogeneous spatial distributions. Let $A = \{a_1, a_2, \ldots, a_m\} \subseteq \mathbb{R}^d$ and $B = \{b_1, b_2, \ldots, b_n\} \subseteq \mathbb{R}^d$ represent instance sets of features f_i and f_j. The directed Hausdorff distance is:

$$h(A, B) = \sup_{a \in A} \inf_{b \in B} d(a, b), \tag{4}$$

where $d(a, b) = \|a - b\|_2$ is the Euclidean norm in $\mathbb{R}^d$. The Hausdorff distance is:

$$D_H(A, B) = \max\{h(A, B), h(B, A)\}. \tag{5}$$

For a singleton set $\{o_i\}$, the directed Hausdorff distance simplifies to the distance to its nearest neighbor in B. The alignment function $\Phi : O_i \to O_j$ maps instances of f_i to f_j, constrained by a threshold δ_H:

$$\Phi(o_i) = o_j \implies d(o_i, o_j) \leq \delta_H, \quad o_j = \arg\min_{o' \in O_j} d(o_i, o'). \tag{6}$$

The threshold δ_H is set based on the dataset's spatial density, e.g., 100 m for *Shanghai_POI* to align with urban co-location distances.

The alignment process employs a three-stage strategy (direct, proportional, and residual allocation) to handle varying instance cardinalities. The unified allocation set is:

$$\mathcal{A}(f_i, f_j) = \mathcal{A}_{\mathrm{direct}}(f_i, f_j) \cup \mathcal{A}_{\mathrm{prop}}(f_i, f_j) \cup \mathcal{A}_{\mathrm{resid}}(f_i, f_j), \tag{7}$$

with total cost:

$$C(f_i, f_j) = \sum_{(o_i, o_j) \in \mathcal{A}(f_i, f_j)} d(o_i, o_j). \tag{8}$$

The HD-SCPM algorithm uses this framework to construct a proximity graph with bidirectional edges:

$$E_{\mathrm{proximity}} = \{(o_i, o_j) \in \mathcal{A}(f_i, f_j) \mid (o_j, o_i) \in \mathcal{A}(f_j, f_i)\}. \tag{9}$$

This process takes instance sets O_i, O_j, and threshold δ_H as input, outputting the proximity graph $E_{\mathrm{proximity}}$. It uses spatial partitioning to mimic the inf and sup operations of the

Hausdorff distance, ensuring efficiency and adaptability. The spatial partitioning employs a randomized Voronoi tessellation, where feature instances serve as seed points to generate Voronoi cells, approximating nearest-neighbor searches for efficient alignment. To control potential variability introduced by randomization, we use fixed seed points derived from the dataset's instance locations and apply multiple sampling iterations (e.g., Monte Carlo methods) to stabilize results, ensuring consistent and reproducible proximity relationships.

To illustrate the evolution of the three-stage alignment process, we present a four-panel figure (Figure 2) that integrates the direct, proportional, and residual allocation stages, culminating in the final proximity graph. Figure 2a depicts direct allocation using green dashed arrows to indicate the nearest-neighbor assignments from instances of feature f_i to f_j, clarifying the directional pairing process. Figure 2b illustrates proportional allocation, adding orange dashed arrows (with a distinct style) to show balanced assignments based on the proportionality factor. Figure 2c incorporates residual allocation, using purple solid arrows to highlight the assignment of remaining instances, ensuring all instances are allocated. Figure 2d presents the final proximity graph with bidirectional edges, consistent with the original visualization. The panels are arranged in a sequential order (top-left, top-right, bottom-right, bottom-left) to clearly demonstrate the step-by-step evolution of the allocation process.

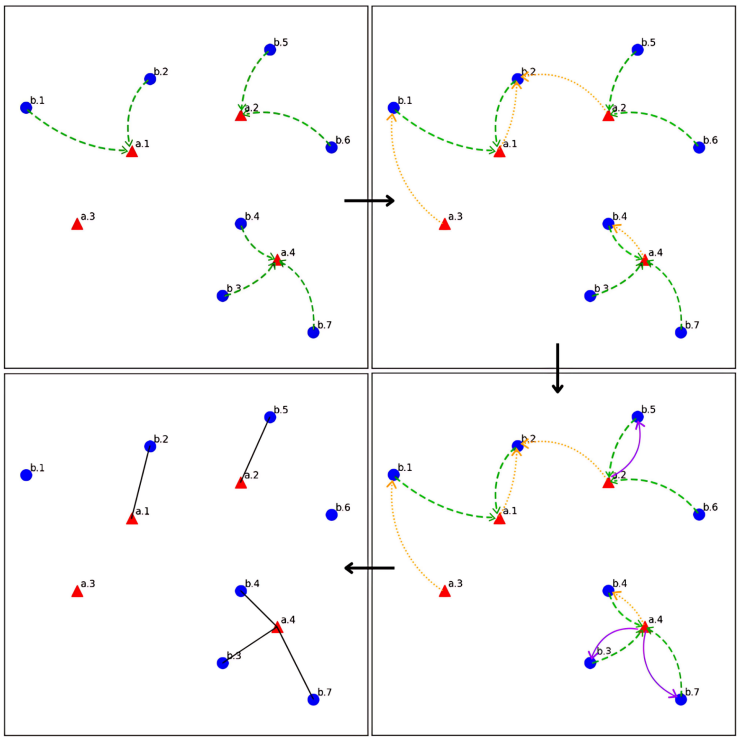


Figure 2. Evolution of the three-stage alignment process: (**a**) Direct allocation with green dashed arrows, (**b**) Proportional allocation with orange dashed arrows, (**c**) Residual allocation with purple solid arrows, (**d**) Final proximity graph.

From residual allocation to the final proximity graph, the HD-SCPM algorithm transforms the allocation results into a proximity graph by retaining only bidirectional relationships. Specifically, instance pairs (o_i, o_j) with bidirectional arrows (i.e., $(o_i, o_j) \in \mathcal{A}(f_i, f_j)$ and $(o_j, o_i) \in \mathcal{A}(f_j, f_i)$) are represented as undirected black solid lines in Figure 2d, indicating a confirmed proximity relationship. Instance pairs with only unidirectional arrows (i.e., $(o_i, o_j) \in \mathcal{A}(f_i, f_j)$ but $(o_j, o_i) \notin \mathcal{A}(f_j, f_i)$) are removed, as they do not satisfy the bidirectional condition for proximity, and thus are not considered neighbors. This process,

transitioning from the purple solid arrows in Figure 2c to the black solid lines in Figure 2d, ensures that only robust spatial relationships are retained in the proximity graph.

To highlight the advantages of the HD-SCPM algorithm over traditional threshold-based methods, we present a side-by-side comparison in Figure 1. Subfigure (a) illustrates the proximity relationships identified by the traditional method, where instances such as a_2 and b_6 are connected by a black solid line, indicating a proximity relationship based solely on the distance threshold δ. Subfigure (b) shows the HD-SCPM method's results, where the same instance pair a_2 and b_6 is connected by a light gray solid line, indicating that it does not satisfy the bidirectional alignment condition and is thus not considered a proximity relationship. This comparison demonstrates that HD-SCPM's requirement for bidirectional alignment reduces false positives, ensuring more robust and accurate proximity relationships.

3.3. Direct Allocation

When $|O_i| \geq |O_j|$, direct allocation assigns each instance $o_i \in O_i$ to its nearest neighbor in O_j:

$$\mathcal{A}_{\text{direct}}(f_i, f_j) = \{(o_i, o_j) \mid o_i \in O_i, o_j = \Phi(o_i)\}, \tag{10}$$

with $\Phi(o_i)$ as defined in Equation (6). Ties are resolved by instance indices. This process is visualized in Figure 2a, where green dashed arrows indicate the directional nearest-neighbor assignments.

Proof. Direct allocation minimizes the total distance by greedily assigning each o_i to its nearest neighbor, ensuring local optimality. By induction, this is globally optimal for $|O_i| \geq |O_j|$. □

3.4. Proportional Allocation

When $|O_i| < |O_j|$, proportional allocation balances the assignment with proportionality factor:

$$k = \left\lfloor \frac{|O_j|}{|O_i|} \right\rfloor, \tag{11}$$

assigning k instances of O_j to each $o_i \in O_i$:

$$\mathcal{A}_{\text{prop}}(f_i, f_j) = \{(o_i, o_j) \mid o_j \in O_j, o_i \in \text{NN}_k(o_j, O_i)\}, \tag{12}$$

where $\text{NN}_k(o_j, O_i)$ is the set of k-nearest neighbors of o_j in O_i. This is depicted in Figure 2b, with orange dashed arrows (distinct from direct allocation) illustrating the balanced assignments.

Proof. Proportional allocation ensures balanced distribution, assigning approximately k instances of O_j per $o_i \in O_i$. The number of assignments is $m_i \cdot k \leq m_j$, with $r = m_j - m_i \cdot k$ unallocated instances, ensuring fairness. □

3.5. Residual Allocation

Residual allocation assigns the remaining r instances of O_j:

$$O_j^{\text{unalloc}} = O_j \setminus \{o_j \mid \exists o_i, (o_i, o_j) \in \mathcal{A}_{\text{prop}}(f_i, f_j)\}, \tag{13}$$

$$\mathcal{A}_{\text{resid}}(f_i, f_j) = \{(o_i, o_j) \mid o_j \in \text{Top}_r(O_j^{\text{unalloc}}, o_i)\}, \tag{14}$$

where $\text{Top}_r(O_j^{\text{unalloc}}, o_i)$ selects the r closest unallocated instances to o_i. This is shown in Figure 2c, where purple solid arrows indicate the assignment of remaining instances.

Proof. Residual allocation ensures all instances are assigned:

$$|\mathcal{A}_{\text{prop}}(f_i, f_j)| + |\mathcal{A}_{\text{resid}}(f_i, f_j)| = m_j. \tag{15}$$

Greedy selection of the r closest instances minimizes the allocation cost. □

3.6. Alignment Ratio of Instances

3.6.1. Alignment Ratio Between Two Spatial Features

The alignment ratio for features f_i and f_j quantifies the proportion of aligned instance pairs:

$$\text{AR}(f_i, f_j) = \frac{|\mathcal{A}_{\text{aligned}}(f_i, f_j)|}{|O_i| + |O_j|}, \tag{16}$$

where $\mathcal{A}_{\text{aligned}}(f_i, f_j)$ is the set of aligned pairs. A pattern $\{f_i, f_j\}$ is frequent if:

$$\text{AR}(f_i, f_j) \geq \theta_{\text{AR}}, \tag{17}$$

with $\theta_{\text{AR}} \in (0, 1]$ as a user-defined threshold. The aligned pairs are defined as:

$$\mathcal{A}_{\text{aligned}}(f_i, f_j) = \{(o_i, o_j) \in E_{\text{proximity}} \mid d(o_i, o_j) \leq \delta_H\}, \tag{18}$$

where $d(o_i, o_j)$ is the Euclidean distance defined in Equation (4). The thresholds δ_H and θ_{AR} are calibrated to ensure statistically significant patterns. For *spatial_data_A*, $\delta_H = 5$ and $\theta_{\text{AR}} = 0.3$ are set based on average nearest neighbor distances and frequent pattern significance. For *Shanghai_POI*, $\delta_H = 100$ meters and $\theta_{\text{AR}} = 0.5$ reflect urban co-occurrence rates and pattern frequency analysis. For example, in Figure 2d, the pattern $\{a, b\}$ is frequent as $\text{AR}(a, b) \approx 0.4545 \geq 0.3$.

3.6.2. Alignment Ratio of k-Order Spatial Co-Location Patterns

For a k-order pattern $c_k = \{f_1, f_2, \ldots, f_k\}$, the alignment ratio is:

$$\text{AR}(c_k) = \prod_{1 \leq i < j \leq k} \text{AR}(f_i, f_j)^{\omega_{ij}}, \tag{19}$$

with $\omega_{ij} = 1/k$, chosen to ensure equal contribution of each feature pair to the overall alignment ratio, reflecting the balanced influence of all pairwise relationships in a k-order pattern. This uniform weighting, where the sum of weights across all $k(k-1)/2$ feature pairs approximates 1, maintains consistency with the participation index and simplifies computation for heterogeneous spatial distributions. or alternatively:

$$\text{AR}(c_k) = \frac{|\mathcal{A}(c_k)|}{|I(c_k)|}, \tag{20}$$

where $\mathcal{A}(c_k) = \{cl \in T(c_k) \mid \forall (o_i, o_j) \in cl, d(o_i, o_j) \leq \delta_H\}$, and $I(c_k) = \bigcup_{f_i \in c_k} O_i$.

3.6.3. Decreasing Property of Alignment Ratio for k-Order Patterns

For a $(k+1)$-order pattern $c_{k+1} = c_k \cup \{f_{k+1}\}$, the aligned instance set satisfies:

$$\mathcal{A}(c_{k+1}) \subseteq \mathcal{A}(c_k), \tag{21}$$

and the total instance set grows:

$$I(c_k) \subseteq I(c_{k+1}). \tag{22}$$

Thus:

$$|\mathcal{A}(c_{k+1})| \leq |\mathcal{A}(c_k)|, \quad |I(c_{k+1})| \geq |I(c_k)|. \tag{23}$$

From Equation (20), the alignment ratio is:

$$\mathrm{AR}(c_{k+1}) = \frac{|\mathcal{A}(c_{k+1})|}{|I(c_{k+1})|}, \tag{24}$$

implying $\mathrm{AR}(c_{k+1}) \leq \mathrm{AR}(c_k)$.

Proof. Since $|\mathcal{A}(c_{k+1})| \leq |\mathcal{A}(c_k)|$ and $|I(c_{k+1})| \geq |I(c_k)|$, the alignment ratio is non-increasing, as $\frac{|\mathcal{A}(c_{k+1})|}{|I(c_{k+1})|} \leq \frac{|\mathcal{A}(c_k)|}{|I(c_k)|}$. □

4. Algorithm and Analysis

This study transforms traditional spatial co-location pattern mining into an alignment problem between instances of spatial features to enhance characterization accuracy and algorithmic performance. First, Voronoi tessellation is applied based on random spatial features, and on this basis, a KD-tree is constructed for each feature subset within each region. Leveraging this, a three-stage allocation strategy (direct allocation, proportional allocation, and residual allocation) is employed in conjunction with a dynamic k-Nearest Neighbor (k-NN) algorithm to initialize instance alignment, mitigating the imbalance in handling sparse and dense regions. When constructing the spatial proximity relationship, directed edges are converted into undirected edges to improve relationship stability, and a transfer substitution strategy is applied to adjust instance pairs that locally violate "spatial proximity," ensuring geographical rationality. Ultimately, a Spatial Proximity Relationship Graph (SNRG) is generated, providing the foundational structure for pattern mining. The alignment ratio is adopted as the frequency metric, and the Moran's index [14] is used to validate spatial autocorrelation, enabling the identification of high-frequency proximity patterns while optimizing computational efficiency.

4.1. Algorithm Implementation

The first part of the algorithm (Alignment Process) generates the Spatial Proximity Relationship Graph (SNRG) through six steps, as illustrated in Figure 3: First, it initializes the edge sets; then, it generates a Voronoi diagram to partition the spatial domain based on random features; next, it constructs a KD-tree for the feature instance set in each region; subsequently, for each pair of features, it performs direct allocation (Equation (10)), proportional allocation (Equation (12)), and residual allocation (Equation (14)), utilizing the dynamic k-NN algorithm and the Hausdorff distance concept Equation (4)) to adaptively align instances and generate directed edges; afterward, it filters bidirectional edges to convert them into undirected edges; finally, it returns the SNRG.

The second part of the algorithm (Alignment Ratio Calculation) mines frequent patterns through five steps, as shown in Figure 4: First, it initializes the frequent pattern set; then, it calculates the alignment ratio for each pair of features (Equation (16)), filtering second-order frequent patterns; next, it sets the initial order $k = 2$; subsequently, it iteratively generates higher-order candidate patterns, pruning with the alignment ratio monotonicity (Equation (21)), calculating the k-order alignment ratio (Equation (19)) only for candidates whose sub-patterns are all frequent, and adding them to the frequent pattern set if the threshold is met; finally, it returns all frequent patterns.

Algorithm 1 Alignment Process Algorithm

Input: a spatial dataset
Output: neighborhood relationship graph SNRG

```
Initialize edges ← ∅, undirected_edges ← ∅
voronoi ← Voronoi(F[random])
for each region in voronoi:
    build feature_trees[region] ← KDTree(F in region)
for each pair (f_i, f_j) in F:
    if |O_i| > |O_j|:
        edges ← nearest_assign(O_i, O_j, feature_trees)
    else if |O_j| > |O_i|:
        edges ← nearest_assign(O_j, O_i, feature_trees)
    k ← floor(max(|O_i|, |O_j|) / min(|O_i|, |O_j|))
    edges ← proportional_assign(O_min, O_max, k, feature_trees)
    edges ← remainder_assign(O_max, O_min, feature_trees)
undirected_edges ← { (a, b) | (a, b) ∈ edges ∧ (b, a) ∈ edges }
return SNRG ← undirected_edges
```

Figure 3. Algorithm 1.

Algorithm 2 Alignment Rate Compute Algorithm

Input: a spatial dataset, SNRG, alignment rate threshold T
Output: frequent patterns

```
frequent_patterns ← ∅
for each pair (f_i, f_j) in F:
    AI ← alignment_rate(O_i, O_j, SNRG)
    if AI > T:
        frequent_patterns.add((f_i, f_j))
k ← 2
while frequent_patterns has patterns of size k-1:
    candidates ← generate_candidates(frequent_patterns, k)
    for each candidate in candidates:
        if all sub_patterns in frequent_patterns:
            AI_ck ← product_of_pairwise_AI(candidate)
            if AI_ck > T:
                frequent_patterns.add(candidate)
    k ← k + 1
return frequent_patterns
```

Figure 4. Algorithm 2.

The time complexity of constructing the Voronoi diagram is $O(n \log n)$, where n is the number of instances of the randomly selected features. The time complexity of building a local KD-tree is $O(|F| \log |F|)$, where $|F|$ is the average number of instances in each Voronoi region. Since a KD-tree is built for each feature in each region, and there are $O(n)$ regions in total, the total complexity is $O(n|F| \log |F|)$. The time complexity of nearest neighbor search is $O(\log |F|)$. In direct allocation, for each point in each region queried against other features, the complexity per region is $O(|F| \log |F|)$, resulting in a total complexity of $O(|F|^3 n \log |F|)$. The complexities of proportional allocation and residual allocation are also $O(|F|^3 n \log |F|)$. The complexity of generating directed and undirected edges is $O(e)$, where e is the number of edges, and in the worst case, e is $O(|F|^2 n)$. The time complexity of calculating the alignment ratio is $O(|F|^2 n)$, as it requires counting the number of edges between each pair of features. The complexity of identifying frequent patterns is $O(p|F|)$, where p is the number of frequent patterns. In summary, the total time complexity of the algorithm is:

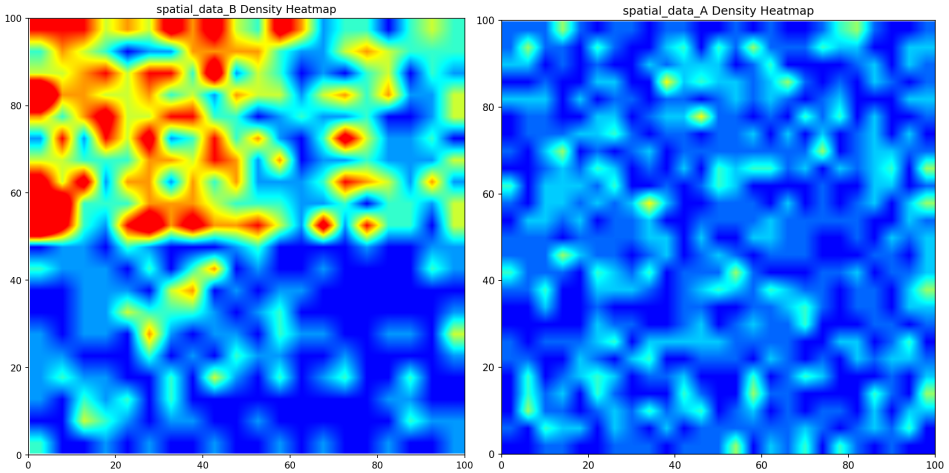
$$O(|F|^3 n \log |F| + p|F|) \tag{25}$$

5. Experimental Results Analysis

This section analyzes the runtime efficiency and mining results of the proposed algorithm through experiments, comparing it with the traditional join-based pattern mining algorithm from [1] and the CPI-tree mining algorithm from [4]. The algorithm was implemented in Python3.10, and all programs were run on a PC with an Intel Core i9 CPU, 32 GB RAM, and Windows 11 Professional.

5.1. Experimental Dataset and Parameter Settings

To analyze the efficiency of the algorithm on datasets of varying scales and distributions, this paper uses a randomly generated spatial feature dataset, *spatial_data_A* (containing 4 features), with 20% of the data sampled as *sample_spatial_data* for experiments. To enhance visual aesthetics, the heatmaps of *spatial_data_A* and *spatial_data_B* are presented as subfigures in Figure 5, illustrating their density variations with higher energy in high-density regions. Based on *spatial_data_A*, a non-uniformly distributed dataset *spatial_data_B* is generated through transformation. Additionally, a real-world dataset, Shanghai Point of Interest (POI) (where each POI category is treated as a feature), is used. The *Shanghai_POI* dataset includes 217 diverse feature categories (e.g., supermarkets, subway stations, banks, pharmacies), capturing complex urban environments with varied functional zones such as commercial, industrial, and residential areas. Due to the large data volume (242,557 instances), two representative features (primary schools and farmers' markets) are selected for visualization in Figure 6 to ensure clarity, while the full dataset with all features is used for pattern mining. The dataset information is detailed in Table 1.



(**a**) *spatial_data_A* (**b**) *spatial_data_B*

Figure 5. Heatmaps of *spatial_data_A* and *spatial_data_B*. Cool tones represent sparse data distribution, warm tones represent dense data distribution, and the transition from cool to warm tones indicates the process of data distribution from sparse to dense. (**a**) Heatmap of *spatial_data_A* showing density variations with higher energy in high-density regions. (**b**) Heatmap of *spatial_data_B* illustrating density variations of the non-uniformly distributed dataset.

Table 1. Dataset information.

Dataset	Number of Features	Number of Instances	Spatial Range
spatial_data_A	4	700	100×100
sample_spatial_data	4	140	100×100
Shanghai_POI	217	242,557	$120° \sim 122$ °E, $30° \sim 32$ °N

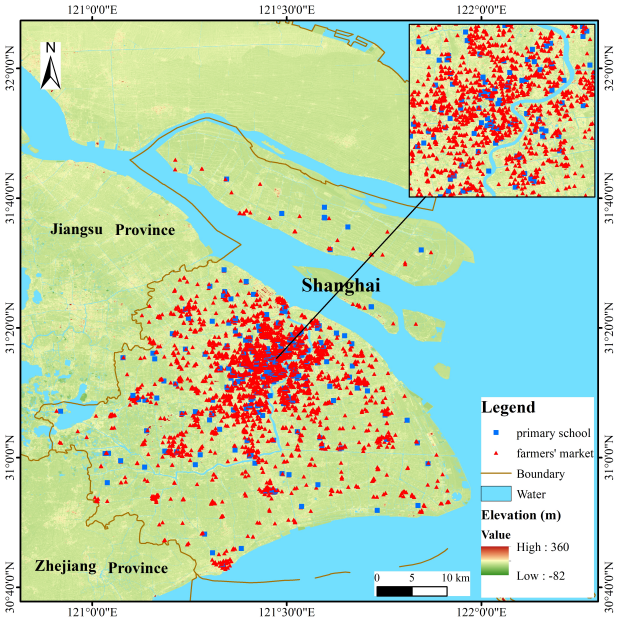


Figure 6. Distribution of two representative features (primary schools and farmers' markets) from the *Shanghai_POI* dataset, selected for visualization to avoid overcrowding due to the high density of the full 217-feature dataset.

The *spatial_data_A* dataset exhibits a uniform spatial distribution, suitable for controlled testing, while *spatial_data_B* introduces non-uniform clustering to simulate varied density patterns. In contrast, the *Shanghai_POI* dataset reflects complex urban spatial patterns with diverse functional zones, such as high-density commercial districts and lower-density industrial areas, making it representative of real-world heterogeneity. The synthetic datasets, while effective for validating algorithmic robustness, lack the intricate spatial interactions of real urban environments. The *Shanghai_POI* dataset, with its large scale and high feature diversity, poses computational challenges due to its volume and complexity. HD-SCPM excels in detecting patterns in high-density, heterogeneous urban settings like Shanghai's Pudong district, where diverse facilities coexist. However, its performance may be limited in sparse regions (e.g., suburban areas) with fewer co-location opportunities or in single-function zones where patterns are less prevalent. Additionally, the computational complexity of $O(|F|^3 n \log |F|)$ (Section 4.1, Equation (25)) restricts scalability in extremely high-dimensional or dense datasets, indicating a need for optimization in such scenarios.

In the experiments, the default settings of the algorithm's required parameters—distance threshold, participation threshold, and alignment ratio threshold—across different datasets are listed in Table 2. The join-based and CPI-tree algorithms maintain the same distance and participation thresholds.

Table 2. Parameter settings.

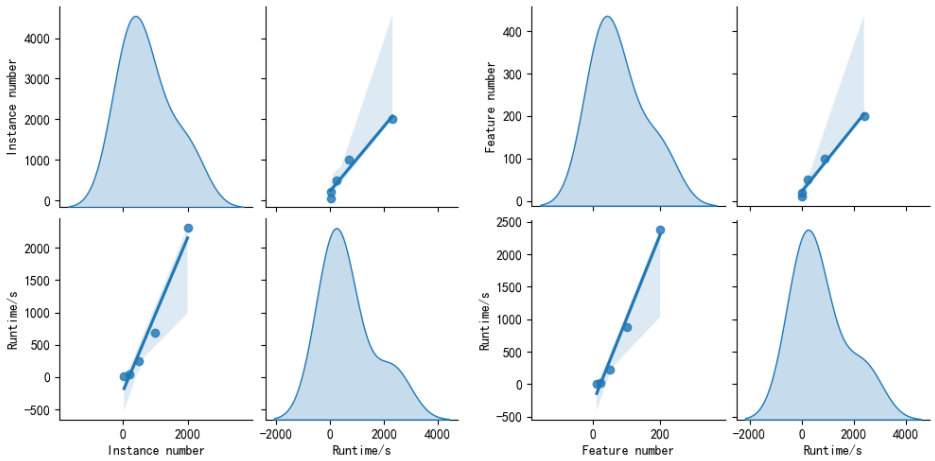
Dataset	Distance Threshold	Participation Threshold	Alignment Ratio Threshold
spatial_data_A	5	0.3	0.3
sample_spatial_data	5	0.3	0.3
Shanghai_POI	100	0.5	0.5

5.2. Experimental Evaluation

The research method proposed in this paper, High-Dimensional Spatial Co-location Pattern Mining (HD-SCPM), differs from traditional spatial co-location pattern mining algorithms. It analyzes the impact of data scale and complexity on the algorithm from two aspects: (1) the impact of the number of instances on algorithm efficiency, and (2) the impact of the number of features on algorithm efficiency. Additionally, it compares HD-SCPM with join-based [1] and CPI-tree [4] algorithms in terms of runtime efficiency, mining accuracy, noise tolerance, robustness, and pattern mining quality.

5.2.1. Impact of the Number of Instances and Features on the Runtime Efficiency of the HD-SCPM Algorithm

When the number of features is fixed at 10 and the number of instances is 50, 200, 500, 1000, and 2000, respectively, the algorithm's runtime is shown in Figure 7a. When the number of instances is fixed at 50 and the number of features is 10, 20, 50, 100, and 200, respectively, the runtime is shown in Figure 7b.



(**a**) Fixed number of features (**b**) Fixed number of instances

Figure 7. Impact of instance and feature quantity on runtime efficiency of HD-SCPM. (**a**) Runtime with varying number of instances and fixed number of features. (**b**) Runtime with varying number of features and fixed number of instances.

Figure 7 reveals that as the number of instances and features increases, the runtime of the HD-SCPM algorithm significantly grows. This is primarily because the computational complexity of the HD-SCPM algorithm mainly stems from instance alignment and alignment ratio computation. Specifically, an increase in the number of instances leads to a multiplicative increase in distance calculations and traversal operations. Meanwhile, an increase in the number of features causes an exponential rise in the number of feature pair combinations. This, in turn, affects the construction of the proximity relationship graph, alignment ratio computation, and frequent pattern determination. Consequently, when both the number of instances and features increase simultaneously, the runtime growth becomes more pronounced. This highlights the importance of optimizing these processes to enhance the algorithm's efficiency in handling large-scale and high-dimensional data.

5.2.2. Impact of Dataset Scale on Runtime Efficiency

The experiment evaluates the runtime efficiency differences of the three algorithms on the *Shanghai_POI* dataset as the dataset size continuously increases. When the number of features increases by 5 each time and the number of instances per feature increases by 50 each time, the runtime of the three algorithms is shown in Figure 8.

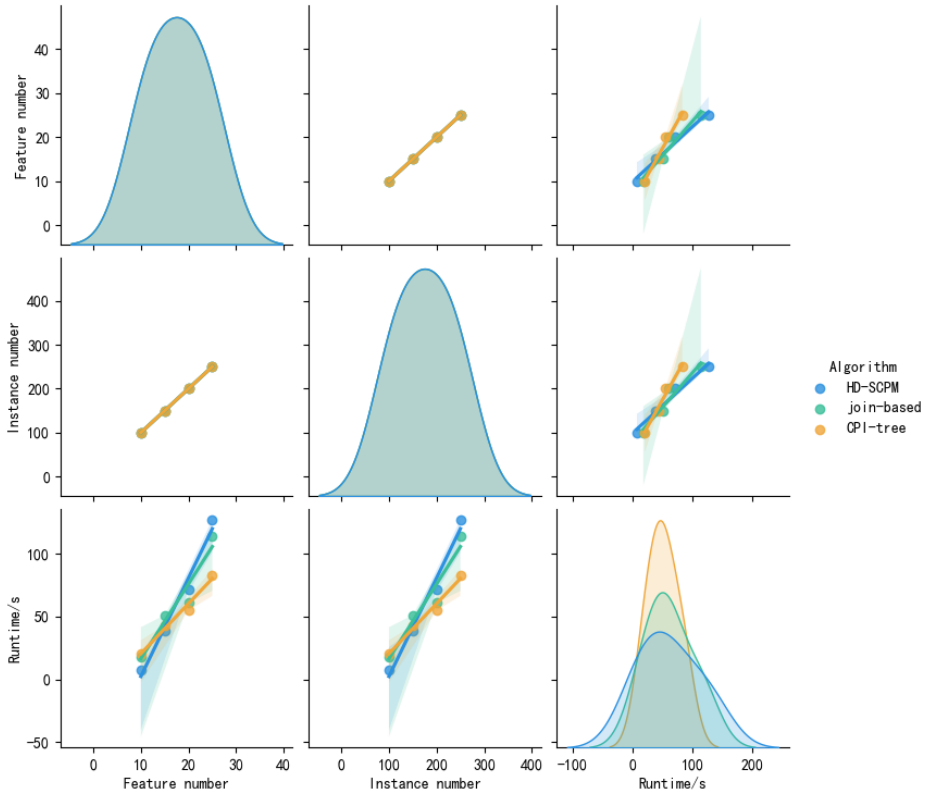
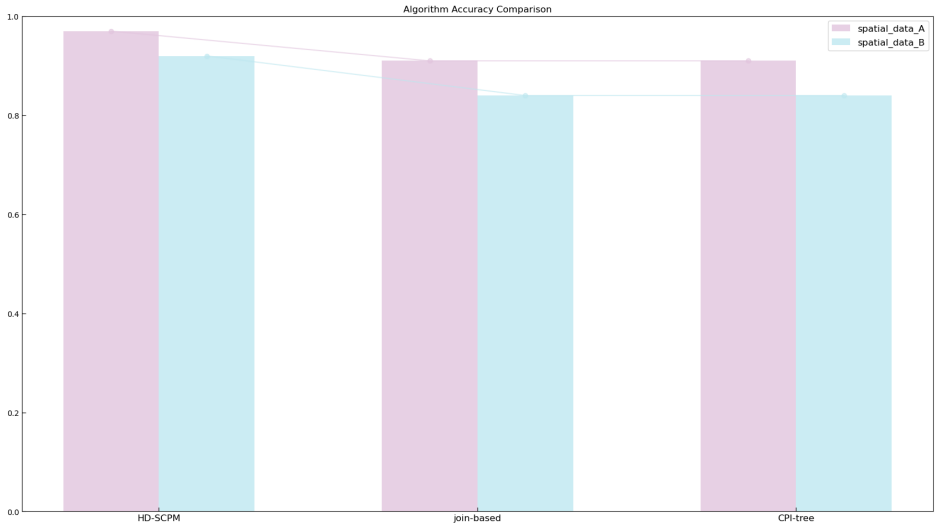


Figure 8. Impact of dataset size on algorithm efficiency.

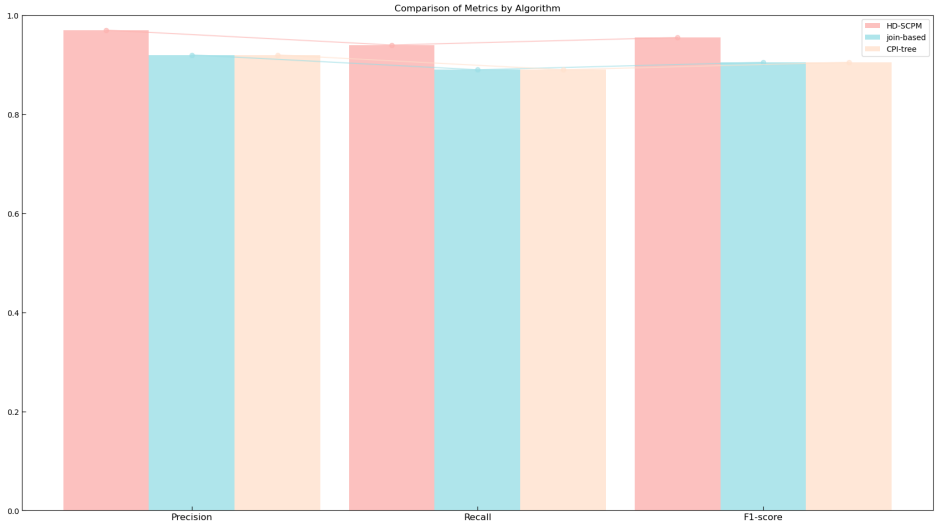
Figure 8 presents a 3 × 3 matrix visualization, comparing the performance of three algorithms—HD-SCPM (blue), Join-based (green), and CPI-tree (orange)—across three variables: Feature number (range 0–50, peak around 40), Instance number (range 0–500, peak around 400), and Runtime/s (range 0–200 s, peak 0–50 s). The diagonal panels display the kernel density estimation (KDE) distribution for each variable, while the off-diagonal panels show scatter plots of pairwise variable relationships, with points colored by algorithm. The figure indicates that, on small-scale datasets, both HD-SCPM and the Join-based algorithm [1] exhibit lower runtimes compared to the CPI-tree algorithm [4], with HD-SCPM slightly outperforming the Join-based method. However, as the dataset scale increases (with rising Feature number and Instance number), the efficiency of HD-SCPM declines due to time-consuming alignment operations, and the Join-based algorithm also experiences reduced efficiency due to an increase in intermediate results, though it remains slightly better than HD-SCPM on medium-to-small-scale datasets. The CPI-tree algorithm, through reducing redundancy and employing efficient pruning techniques, shows better adaptability on some large-scale datasets, but its runtime growth trend remains noticeable within the current data range, requiring further validation of its overall performance on large scales.

5.2.3. Differences in Pattern Mining Accuracy Among the Three Algorithms

Accuracy comparison experiments were conducted on the *sample_spatial_data* dataset (containing 29 frequent patterns) and the *spatial_data_A* dataset (containing 174 frequent patterns). The accuracy of the three algorithms was initially analyzed, followed by further evaluation using precision, recall, and F1-score metrics. The *spatial_data_B* dataset contains 119 frequent patterns. The accuracy of the three algorithms on *spatial_data_A* and *spatial_data_B* is shown in Figure 9a. The evaluation metrics for mining frequent patterns on *spatial_data_A* are presented in Figure 9b, while those for *sample_spatial_data* are shown in Figure 9c. In all panels, the vertical axis represents proportional values ranging from 0 to 1, rounded to one decimal place, and uniformly displayed as percentage data.

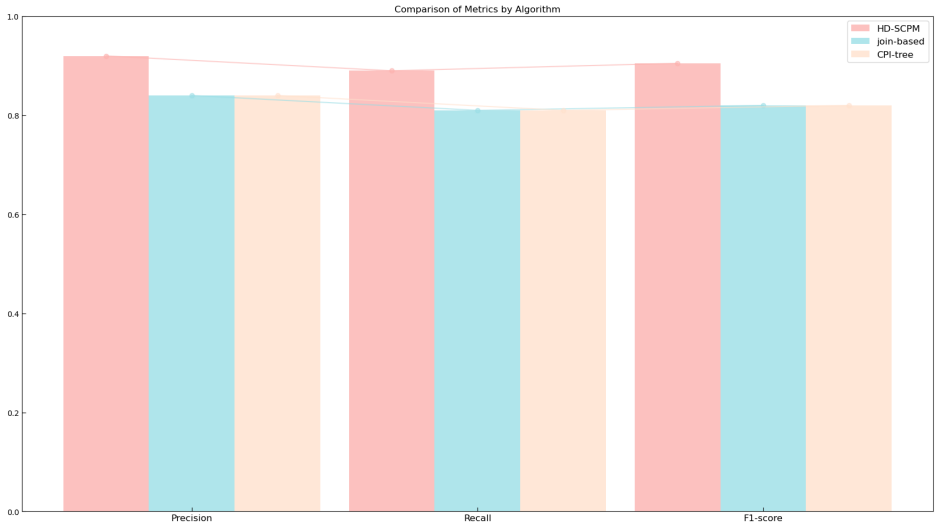


(**a**) Accuracy on *spatial_data_A* and *spatial_data_B*



(**b**) Metrics on *spatial_data_A*

Figure 9. *Cont.*



(**c**) Metrics on *sample_spatial_data*

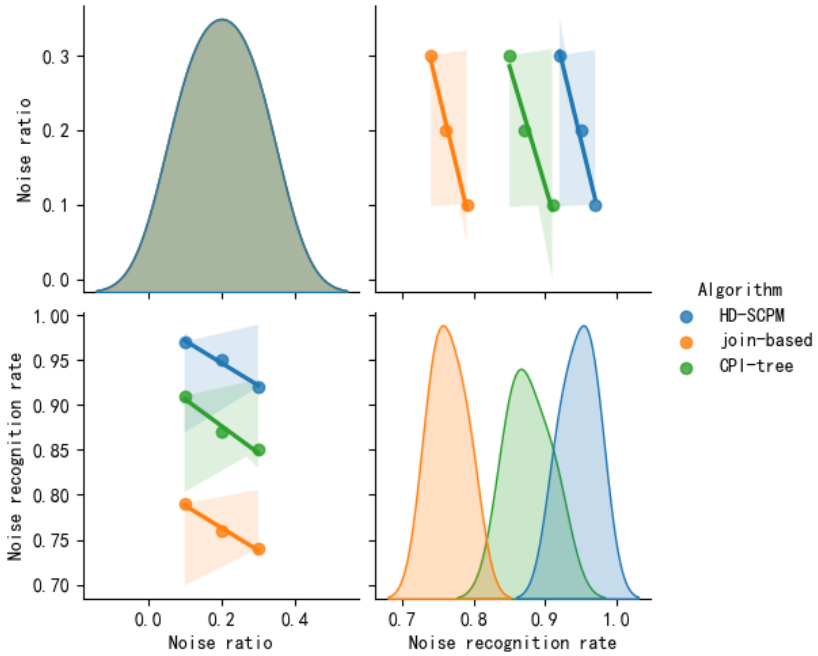
Figure 9. Pattern mining accuracy comparison across different datasets. (**a**) Accuracy of the three algorithms on *spatial_data_A* and *spatial_data_B*. (**b**) Precision, recall, and F1-score on *spatial_data_A*. (**c**) Precision, recall, and F1-score on *sample_spatial_data*.

Figure 9 shows that the HD-SCPM algorithm outperforms both the join-based [1] and CPI-tree [4] algorithms on both small-scale and large-scale datasets. HD-SCPM maintains high accuracy on both balanced and imbalanced datasets, although there is a slight decrease in accuracy on imbalanced data, demonstrating its adaptability to different data distributions. On small-scale data, HD-SCPM achieves high precision but slightly lower recall. In contrast, the join-based and CPI-tree algorithms exhibit high precision, low recall, and consistent performance across different datasets. On large-scale data, HD-SCPM demonstrates high recall with a slight drop in precision, while the other two algorithms experience declines in both precision and recall. Overall, HD-SCPM excels in accuracy, precision, and recall, particularly showing superior performance in complex pattern mining tasks.

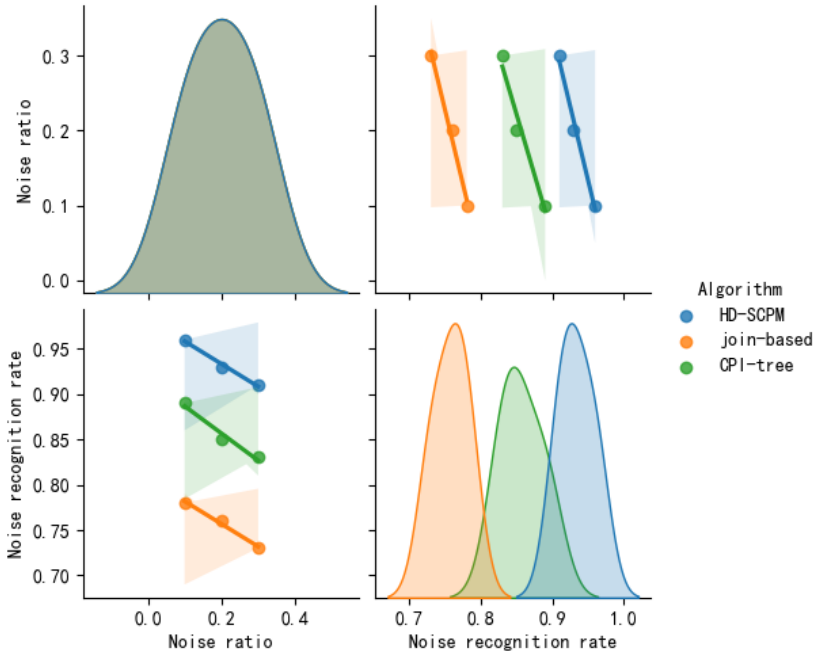
5.2.4. Differences in Noise Tolerance Among the Three Algorithms

Noise tolerance comparison experiments were conducted on the *sample_spatial_data* and *spatial_data_A* datasets, building upon the previous results, to evaluate the proportion of noise points mistakenly identified as frequent pattern points by the three algorithms after adding noise. The noise levels were set at 10%, 20%, and 30%, and the algorithms' effectiveness in noisy environments was assessed using the noise identification rate, defined as the ratio of identified noise points to the total number of noise points. The noise identification rates of the three algorithms on *sample_spatial_data* are shown in Figure 10a, while those on *spatial_data_A* are presented in Figure 10b.

Figure 10 demonstrates that the HD-SCPM algorithm leverages the Hausdorff distance concept for instance alignment, exhibiting strong robustness and effectively resisting noise. Although its identification rate slightly decreases as noise levels increase, it still maintains a high level, demonstrating excellent performance. In contrast, the join-based algorithm [1] is highly sensitive to noise. Its pattern recognition accuracy significantly declines in noisy environments, resulting in poor performance. The CPI-tree algorithm [4] mitigates noise through pruning techniques, but its performance still degrades under high noise levels, showing overall moderate effectiveness.



(**a**) Noise identification rate on *sample_spatial_data*



(**b**) Noise identification rate on *spatial_data_A*

Figure 10. Noise tolerance comparison across different datasets. (**a**) Noise identification rate of the three algorithms on *sample_spatial_data*. (**b**) Noise identification rate on *spatial_data_A*.

5.2.5. Differences in Robustness Among the Three Algorithms

Robustness comparison experiments were conducted on the *spatial_data_A* dataset (containing 174 frequent patterns) and the *spatial_data_B* dataset (containing 119 frequent patterns) to evaluate the performance of the three algorithms under both uniform and non-uniform density distributions. The evaluation is based on the pattern recognition rate, defined as the ratio of the number of identified frequent patterns to the total number of frequent patterns. The experimental results are shown in Figure 11.

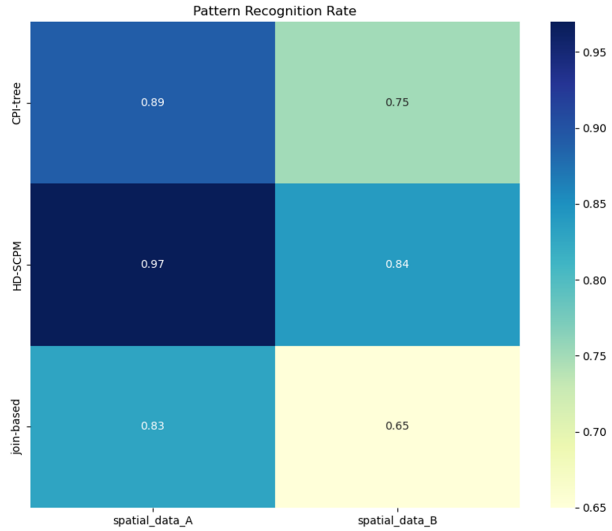


Figure 11. Pattern recognition rates of HD-SCPM, join-based, and CPI-tree algorithms on *spatial_data_A* and *spatial_data_B*.

Figure 11 indicates that HD-SCPM and CPI-tree [4] exhibit similar performance on uniformly distributed datasets, while the join-based algorithm [1] has a lower recognition rate. On non-uniformly distributed datasets, HD-SCPM outperforms both join-based and CPI-tree algorithms, with only a slight but stable decrease in recognition rate. In contrast, both join-based and CPI-tree algorithms experience a significant decline in recognition rate, indicating that HD-SCPM has higher robustness in handling non-uniform density distributions.

5.2.6. Differences in Pattern Mining Quality Among the Three Algorithms

The quality of frequent patterns mined by the three algorithms on real-world datasets is evaluated, focusing on two key aspects: pattern complexity and pattern diversity. Pattern complexity examines whether the three algorithms can deeply mine higher-order frequent patterns, while pattern diversity investigates whether they can uncover rarer and more latent frequent patterns. Comparative experiments were conducted on the *Shanghai_POI* dataset, with 20% of the data randomly sampled for the experiments. The relationships between the number of frequent patterns, the number of categories, and the order of patterns extracted by the three algorithms are shown in Figure 12.

Figure 12 shows that as the order increases, the number and variety of frequent patterns mined by all three algorithms decrease. The join-based [1] and CPI-tree [4] algorithms exhibit similar performance at lower orders, but their performance declines at higher orders (above the fourth order). Specifically, the join-based algorithm produces the fewest effective patterns due to repetitive patterns, while the CPI-tree algorithm remains relatively stable but mines fewer high-order patterns. In contrast, HD-SCPM demonstrates the best performance, excelling notably at the third order and remaining effective at higher orders. Overall, HD-SCPM surpasses both the join-based and CPI-tree algorithms in pattern mining quality, particularly showing superior performance at higher orders.

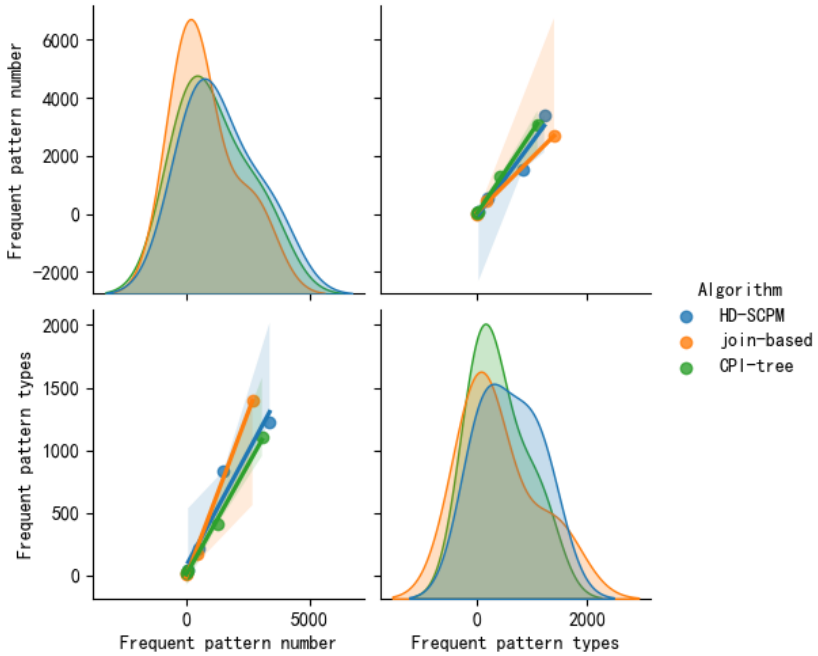


Figure 12. Relationship between pattern mining quality and order for HD-SCPM, join-based, and CPI-tree algorithms on *Shanghai_POI*.

5.3. Pattern Instance Analysis

On the real-world *Shanghai_POI* dataset, we conducted an instance analysis of the patterns mined by the proposed High-Dimensional Spatial Co-location Pattern Mining (HD-SCPM) algorithm and the CPI-tree algorithm [4], with partial mining results shown in Table 3. The traditional join-based algorithm [1] generates candidate patterns and verifies their frequency, which often leads to redundancy, especially for high-order patterns (fourth order and above), resulting in limited practical utility. The CPI-tree algorithm leverages a tree structure and pruning techniques to reduce redundancy and improve efficiency. Although its high-order patterns remain valuable, redundancy control is still insufficient. In contrast, HD-SCPM excels in high-order pattern mining by capturing global relationships through instance alignment, eliminating noise, and adapting to multi-scale distributions. It introduces the Alignment Ratio (AR) as a replacement for frequency counting, recursively filtering meaningful high-order patterns to ensure accuracy and reliability. Compared to CPI-tree, HD-SCPM is more effective in reducing redundancy, focusing on genuine spatial relationships, and offering greater practical value in complex data mining scenarios, particularly in high-density urban environments like Shanghai, where diverse functional areas coexist.

The CPI-tree and HD-SCPM algorithms share similarities but also exhibit differences in mining spatial patterns. Table 3 presents some representative patterns mined by the two algorithms. Both algorithms can mine common third-order patterns, such as {supermarket, public restroom, bus stop}, which reflect typical combinations of public facilities and commercial areas, commonly found in urban residential zones to support daily convenience. However, HD-SCPM demonstrates a unique advantage in capturing high-order and rare patterns, effectively uncovering refined combinations like {industrial park, training institute, gym}. This pattern reveals the spatial synergy between office spaces, skill development facilities, and fitness amenities in industrial parks, directly supporting urban planning by guiding the allocation of complementary services to meet the needs of professionals in knowledge-driven economies. For instance, city planners can use this pattern to optimize

the placement of training institutes and gyms near industrial parks, enhancing employee productivity and well-being.

Furthermore, in fifth-order pattern mining, the pattern extracted by HD-SCPM, {subway station, commercial street, office building, bank, pharmacy}, aligns more closely with real-world urban layouts. This pattern reflects the functional integration of transportation hubs, commercial activities, and essential services in city centers, where pharmacies are strategically located to meet the health demands of large populations in busy areas. Such insights are actionable for urban planners and policymakers, enabling optimized placement of pharmacies to improve public health access or adjust transportation infrastructure to reduce congestion around commercial districts. In contrast, the pattern mined by CPI-tree, {subway station, commercial street, office building, bank, courier service point}, is less reflective of regional functional integration due to the scattered distribution of courier service points, which are often temporary or less tied to urban core functions. This demonstrates that HD-SCPM not only reduces the generation of redundant patterns in high-order mining but also more accurately captures genuine spatial relationships, offering higher practical application value, particularly for optimizing urban service layouts and resource allocation in complex metropolitan areas like Shanghai.

To enhance the practical utility of these findings, we validated the mined patterns through statistical analysis of spatial densities and co-occurrence rates in the *Shanghai_POI* dataset, supplemented by domain knowledge from urban planning experts. For example, the pattern {subway station, commercial street, office building, bank, pharmacy} was cross-verified with Shanghai's urban zoning data, confirming its alignment with high-traffic commercial zones. These patterns can guide actionable decisions, such as prioritizing pharmacy locations near office buildings or enhancing bus stop accessibility near supermarkets. Compared to CPI-tree, HD-SCPM's ability to focus on semantically meaningful patterns reduces false positives and supports precise urban planning and commercial decision-making, as demonstrated in applications like optimizing service facility placements in Shanghai's Pudong district.

Table 3. Partial patterns discovered by CPI-tree and HD-SCPM algorithms.

Pattern Order	CPI-tree Algorithm	HD-SCPM Algorithm
3rd Order	{supermarket, public restroom, bus stop}, {subway station, commercial street, office building}	{supermarket, public restroom, bus stop}, {*industrial park, training institute, gym*}, {subway station, commercial street, office building}
4th Order	{supermarket, public restroom, bus stop, restaurant}, {subway station, commercial street, office building, bank}	{subway station, commercial street, office building, bank}, {*industrial park, training institute, gym, cafe*}
5th Order	{supermarket, public restroom, bus stop, restaurant, convenience store}, {**subway station, commercial street, office building, bank, courier service point**}	{*subway station, commercial street, office building, bank, pharmacy*}, {*industrial park, training institute, gym, cafe, convenience store*}

Note: Italicized patterns are mined by the HD-SCPM algorithm but not by the CPI-tree algorithm, while bold patterns are mined by the CPI-tree algorithm but not by the HD-SCPM algorithm.

6. Conclusions and Future Work

The High-Dimensional Spatial Co-location Pattern Mining (HD-SCPM) algorithm demonstrates superior performance in mining spatial co-occurrence patterns with complex

feature combinations. It employs a three-level allocation mechanism—direct allocation, proportional allocation, and residual allocation—which effectively overcomes the limitations of traditional fixed-threshold methods in terms of noise sensitivity and density heterogeneity. Experimental results show that the HD-SCPM algorithm exhibits strong robustness in handling local outliers. The Alignment Ratio (AR) metric, which employs a cumulative multiplication strategy to quantify the global synergistic relationships between features, significantly improves pattern recall compared to traditional participation models [5]. Additionally, leveraging its diminishing property, HD-SCPM effectively suppresses the generation of redundant patterns in high-order pattern mining. Through a tight integration of theoretical innovation and empirical analysis, this study proposes an adaptive spatial pattern mining method that combines robustness and scalability.

Future work will focus on further optimizing the performance of the HD-SCPM algorithm, particularly its processing efficiency on large-scale datasets. We plan to incorporate the tree structure of the CPI-tree algorithm [4] into HD-SCPM to optimize data storage and retrieval, thereby reducing redundant computations and improving overall operational efficiency. Furthermore, we will introduce the pruning strategy of CPI-tree to efficiently filter out irrelevant feature combinations, further enhancing the algorithm's speed while maintaining mining accuracy [3]. Additionally, future research will explore frequent pattern mining at different scales to meet the demand for patterns at various levels in practical applications, drawing on multi-scale spatial analysis techniques [2,6,7]. This will provide more diverse solutions for complex application scenarios, such as urban planning and environmental monitoring.

Author Contributions: Conceptualization, Xichen Liu and Muquan Zou; software, Xichen Liu; validation, Xichen Liu; writing—original draft preparation, Xichen Liu; resources, Yajie Li; writing—review and editing, Muquan Zou and Yajie Li; supervision, Muquan Zou. All authors have read and agreed to the published version of the manuscript.

Funding: This article is the outcome of a decision-making consultation project in Yunnan Province, partially supported by the following grants: the Special Basic Cooperative Research Program of the Yunnan Provincial Association of Universities (Grants 202101BA070001-152 and 202101BA070001-155), the Yunnan Provincial Philosophy and Social Science Planning Think Tank Project (Grant ZK2024YB15), the Science and Technology Project for Key Industries in Yunnan Higher Education (Grant FWCY-QYCT2024016), and the Specialized Project on Teacher Education of Yunnan Provincial Education Science Planning (Grant GJZ2412).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are not publicly available due to institutional ownership restrictions, as they are proprietary laboratory assets restricted to internal research use.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Meshram, S.; Wagh, K.P. Spatial co-location pattern mining—A survey. In *Fourth Congress on Intelligent Systems*; Springer:Singapore, 2023; pp. 265–280.
2. Li, J.; Wang, L.; Yang, P.; Zhou, L. A novel algorithm for efficiently mining spatial multi-level co-location patterns. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*. [CrossRef]

3. Garaeva, A.; Makhmutova, F.; Anikin, I.; Sattler, K.U. A framework for co-location patterns mining in big spatial data. In Proceedings of the 2017 20th IEEE International Conference on Soft Computing and Measurements, SCM 2017, St. Petersburg, Russia, 24–26 May 2017; pp. 477–480.
4. Wang, L.; Bao, Y.; Lu, J.; Yip, J. A new join-less approach for co-location pattern mining. In Proceedings of the 2008 IEEE 8th International Conference on Computer and Information Technology, Sydney, NSW, Australia, 8–11 July 2008; pp. 197–202. [CrossRef]
5. Wang, L.; Bao, Y.; Lu, Z. Efficient discovery of spatial co-location patterns using the iCPI-tree. *Open Inf. Syst. J.* **2009**, *3*, 69–80. [CrossRef]
6. Feng, S.; Wang, L.; Fang, Y. Mining spatial collocation patterns with dominant features based on fuzzy proximity relations. *Comput. Sci. Appl.* **2021**, *11*, 176.
7. Chang, X.; Lu, J.; Chen, S.; Duan, P. Spatial collocation pattern mining method based on improved column calculation. *Appl. Res. Comput.* **2024**, *41*. [CrossRef]
8. Hu, Z.; Wang, L.; Tran, V.; Chen, H. Efficiently mining spatial co-location patterns utilizing fuzzy grid cliques. *Inf. Sci.* **2022**, *592*, 361–388. [CrossRef]
9. Huang, Y.; Shekhar, S.; Xiong, H. Discovering colocation patterns from spatial data sets: A general approach. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1472–1485. [CrossRef]
10. Yoo, J.S.; Shekhar, S. A partial join approach for mining co-location patterns: A summary of results. In Proceedings of the 2005 SIAM International Conference on Data Mining, Newport Beach, CA, USA, 21–23 April 2005; pp. 556–560.
11. Yoo, J.S.; Shekhar, S.; Celik, M. A join-less approach for co-location pattern mining: A summary of results. In Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05), Houston, TX, USA, 27–30 November 2005; pp. 813–816. [CrossRef]
12. Sheshikala, M.; Rao, D.R.; Prakash, R.V. A map-reduce framework for finding clusters of colocation patterns—A summary of results. In Proceedings of the 2017 IEEE 7th International Advance Computing Conference (IACC), Hyderabad, India, 5–7 January 2017; pp. 129–131.
13. Golfarelli, M.; Wrembel, R. Big data analytics and knowledge discovery. *Data Knowl. Eng.* **2023**, *146*, 102197. [CrossRef]
14. Tran, V.; Tran, D.; Le, A.; Ha, T. Efficiently discovering spatial prevalent co-location patterns without distance thresholds. In Proceedings of the 25th International Conference, iiWAS 2023, Bali, Indonesia, 4–6 December 2023; pp. 447–463. [CrossRef]
15. Chen, S.; Lu, J. Mining algorithm of fuzzy instance space collocation pattern based on Voronoi diagram and distance decay effect. *Hans. J. Data Min.* **2024**, *14*, 65–80. [CrossRef]
16. Wang, L.; Fang, Y.; Zhou, L. Mining spatial prevalent co-location patterns based on graph databases. *J. Front. Comput. Sci. Technol.* **2022**, *16*, 48–58. [CrossRef]
17. Zhang, X.I. Mining spatial co-location kernel patterns based on Voronoi diagrams. *Chin. J. Comput.* **2022**, *45*, 1908–1925. [CrossRef]
18. Wang, X.; Wang, L.; Chen, H.; Zeng, L. Identifying relationship between pollution sources and cancer cases with spatial ordered pair patterns. *Data Anal. Knowl. Discov.* **2021**, *5*, 14–31.
19. Tran, D.H.; Le, A.T.; Ha, T.N. Discovering Prevalent Co-Location Patterns in Different Density Spatial Data Without Distance Thresholds. Master's Thesis, FPT University, Hanoi, Vietnam, 2023.
20. Qian, F.; Chiew, K.; He, Q.; Huang, H., Ma, L. Discovery of regional co-location patterns with k-nearest neighbor graph. In *Advances in Knowledge Discovery and Data Mining*, 1st ed.; Springer: Berlin, Germany, 2013; pp. 174–186. [CrossRef]
21. Qian, F.; Chiew, K.; He, Q.; Huang, H.; Ma, L. Mining regional co-location patterns with k-NNG. *J. Intell. Inf. Syst.* **2014**, *42*, 485–505.
22. Liturri, G. Temporal Co-Location Pattern Discovery in Spatiotemporal Data Through Parallel Computing. Ph.D. Dissertation, Politecnico di Torino, Turin, Italy, 2023.
23. Bui, K.H.N.; Cho, J.; Yi, H. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Appl. Intell.* **2022**, *52*, 2763–2774. [CrossRef]
24. Ver Hoef, J.M.; Peterson, E.E.; Hooten, M.B.; Hanks, E.M.; Fortin, M.J. Spatial autoregressive models for statistical inference from ecological data. *Ecol. Monogr.* **2018**, *88*, 36–59. [CrossRef]
25. Liu, S.; Qiao, P.; Ye, Z.; Li, W.; Dou, Y. DistGrid: Scalable scene reconstruction with distributed multi-resolution hash grid. *arXiv* **2024**, arXiv:2405.04416.
26. Ran, X.; Xi, Y.; Lu, Y.; Wang, X.; Lu, Z. Comprehensive survey on hierarchical clustering algorithms and the recent developments. *Artif. Intell. Rev.* **2023**, *56*, 8219–8264. [CrossRef]
27. Seydi, S.T.; Bozorgasl, Z.; Chen, H. Unveiling the power of wavelets: A wavelet-based Kolmogorov-Arnold network for hyperspectral image classification. *arXiv* **2024**, arXiv:2406.07869.
28. van Kreveld, M.; Miltzow, T.; Ophelders, T.; Sonke, W.; Vermeulen, J.L. Between shapes, using the Hausdorff distance. *Comput. Geom.* **2022**, *100*, 101817. [CrossRef]
29. Jungeblut, P.; Kleist, L.; Miltzow, T. The complexity of the Hausdorff distance. *Discrete Comput. Geom.* **2024**, *71*, 177–213. [CrossRef]
30. Huang, Y.; Pei, J.; Xiong, H. Mining co-location patterns with rare spatial features. *Geoinformatica* **2006**, *10*, 239–271. [CrossRef]

31. Shekhar, S.; Lu, C.; Zhang, P. A unified approach to spatial outliers, co-location and clustering. *IEEE Trans. Knowl. Data Eng.* **2003**, *16*, 1316–1329.
32. Qi, L.; Han, J.; He, M. Weighted co-location pattern mining in spatial data sets. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 2012 ; pp. 134–141. [CrossRef]
33. Azab, A.M.; Ahmadi, H.; Mihaylova, L.; Arvaneh, M. Dynamic time warping-based transfer learning for improving common spatial patterns in brain–computer interface. *J. Neural Eng.* **2020**, *17*, 016061. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.