Spatial Pattern Matching: A Survey

Nicole R. Schneider[1] , **Kent O'Sullivan**[2] and **Hanan Samet**[1]
[1] University of Maryland
[2] University of Sydney
nsch@umd.edu, kosu0918@uni.sydney.edu.au, hjs@umd.edu

Abstract

Recent developments in Artificial Intelligence (AI) have led to new ways for users to search through vast information. However, users may have questions that are grounded in the real world that require spatial inference, for which language models are not well suited. Conversely, traditional spatial search methods, like spatial pattern matching, answer spatial reasoning questions correctly but are noise-intolerant, slow, and brittle. Presently, there are opportunities to integrate AI and spatial pattern matching to enable robust and flexible spatial search. This paper surveys existing spatial pattern matching methods, including the few that apply machine learning to the problem, discussing their efficiency and limitations, and describing opportunities to further enable spatial search through AI.

1 Introduction

Motivation. In the modern era, people rely on Global Positioning System (GPS)-based mapping applications to search for places they want to visit or want to recall to find again. Although this kind of search is inherently spatial in nature, modern search engines are designed for keyword search, not spatial search [Purves *et al.*, 2018]. Even popular mapping applications, like Google Maps [1] only support top-k keyword queries, which just return objects in a region of interest that are relevant to the keywords the user enters [Chen *et al.*, 2022].

Suppose a user wanted to know which restaurants in Washington D.C. are located next to a park and within half a kilometer of a metro stop. The most common form of spatial search, top-k search, would answer such a query with any restaurants, parks, and metro stops in the area of interest, ignoring the desired constraint that they be co-located and fulfill specific distance and topological relationships with respect to each other. Another form of spatial search, spatial group keyword search, would find groups of those entities in a minimum enclosing circle, but still would not satisfy the topological constraint (restaurant adjacent to park) [Cao *et al.*, 2015; Mahmood and Aref, 2017; Mahmood and Aref, 2019; Schneider *et al.*, 2025b]. A third form of spatial search, spatial pattern matching, can address all the constraints in the question, but its computational burden and complicated query format prohibits its use in most applications. The computational complexity stems from the quadratic number of relations that can be defined on a set of entities, which makes the process of matching query and data patterns NP-hard [Fang *et al.*, 2019]. Unlike spatial group keyword search, which is also NP-hard, there are no approximate methods that reduce the complexity of spatial pattern matching broadly.

Although it is regarded as intractable for large search spaces, spatial pattern matching has many useful applications, including in urban planning, region recommendation, scene recognition [Deng *et al.*, 2017], and similar region search [Liu *et al.*, 2018]. Tailored spatial pattern matching systems like GESTALT [O'Sullivan *et al.*, 2023a], Spacekey [Fang *et al.*, 2019], and SketchMapia [Schwering *et al.*, 2014] have been developed to handle use cases like housing search (e.g. "Find all the houses in an area that are within 10km of a school and greater than 5km from any highway."), targeted landmark-based search (e.g. "Find restaurants that have an ocean to the East and a golf course to the South."), and more. While these systems can handle some spatial pattern matching queries, they do not handle every type of spatial constraint across every type of spatial entity. And critically, they require input as a sketch map or pictorial query rather than a text description, which necessitates a custom interface that cannot be easily integrated into generic search engines.

In other words, spatial pattern matching is a powerful form of spatial search capable of answering questions that most search engines cannot currently handle, but its widespread adoption is inhibited by several challenges that we identify and discuss in this paper. Specifically, we highlight

1. The noisy and incomplete nature of geospatial data,
2. The computational challenges associated with resolving spatial pattern matching queries,
3. The lack of flexible, generic methods that work across all entity/relation types, and
4. The requirement that queries be issued in pictorial, sketch, or graph form.

Contribution. As a first step towards making spatial pattern matching more widely usable, this paper surveys the ex-

[1] https://developers.google.com/maps/documentation/places/web-service/search-text

isting state of the art spatial pattern matching approaches. For each method, we derive the theoretical complexity and summarize the types of spatial entities, relations, and other search features that the method supports. Based on our analysis, we identify the gaps in current research that inhibit the use of spatial pattern matching more broadly, including complexity challenges, lack of consistency across types of entities and relations supported by different methods, and complicated query input formats. Finally, we propose ways to address these challenges, such as by leveraging AI to develop approximate solutions, developing benchmarks to measure accuracy and scalability, and using AI to infer missing geospatial data and to interpret natural language spatial pattern matching queries.

2 Related Work

Several previous works have surveyed spatial pattern matching and offered taxonomies that organize the spatial relationships that are possible [Carniel, 2020; Carniel, 2023]. Other works surveyed individual spatial relations in great depth, such as Dylla *et al.*, which covered topological relations defined by qualitative spatial calculi [Dylla *et al.*, 2017]. However, none of these surveys address spatial pattern matching across all entity and relation types, leaving a major gap in knowledge about which aspects of spatial pattern matching prevent it from being applied more broadly. The present paper differs from the aforementioned surveys by covering spatial pattern matching thoroughly and from a practical angle, with the goal of enabling its use in generalized search scenarios. We build on previous surveys and theoretical work that compare spatial pattern matching methods [Schneider *et al.*, 2024a] but instead identify the challenges that prohibit its use in mainstream search systems, highlighting the role AI can play to address many of these issues.

3 Preliminaries

This section formally defines spatial entities and relations, as well as the spatial pattern matching problem. Intuitively, spatial pattern matching, also known as pattern-based keyword search, is the task of matching query keywords and spatial constraints to the data entities that satisfy them. Query constraints can capture metric, topological, directional, and order relationships between entities, based on their position.

3.1 Entities

Spatial entities can be categorized into three main types:

(i) *Points* that consist of an (x, y) coordinate pair in Cartesian space,

(ii) *Lines* that represent the shortest path between two points, and

(iii) *Regions* that represent the area inside a polyline joining several points.

Point entities are typically used to represent objects in the world, like trees, fire hydrants, and bus stops. Line entities are typically used to represent ways, like highways, waterways, and railways. Region entities are typically used to represent the extent of lakes, large buildings, states, and counties. Most of the papers we survey propose methods that operate on point entities, often encoding them as nodes in a graph, with a few methods also supporting line and region entities.

3.2 Relations Between Entities

Points, lines and regions relate to each other spatially through the following types of relationships:

(i) *Metric relations* that describe the distances between spatial entities,

(ii) *Topological relations* that describe how regions, lines, and points interact,

(iii) *Directional relations* that describe how entities are positioned relatively in space,

(iv) *Order relations* that describe the cyclic order in which entities appear relative to a reference point, and

(v) *Hybrid relations* that combine two or more of the previous relation types.

We provide examples of applications for each relation type in Table 1, and formally define each relation type below.

Metric relations

Metric relations characterize how far apart entities are in space. They are often specified using distance intervals and signs representing exclusion in, exclusion out, mutual exclusion, and mutual inclusion [Fang *et al.*, 2019]. They are commonly measured quantitatively, using continuous or discrete values, and less commonly measured qualitatively, using terms like 'near' and 'far' [Papadias *et al.*, 1998]. Formally, a metric relation query Q_μ can be specified using a metric operator $\mu \in \{<, \leq, =, \neq, \geq, >\}$ as

$$Q_\mu(D, dist, c) = \{x, y \in D \mid \langle dist(x, y), c\rangle \in \mu\},$$

where $dist$ is a distance function like Euclidean or Manhattan distance, c is a constant numerical value, and x and y represent a pair of search candidates in database D. Metric relations can be defined over point entities, making them straightforward to measure, or they can be defined over line or region data, requiring a convention to determine which point in the region or line to measure from (e.g. the closest point or the centroid). Most of the methods we survey in Table 3 support spatial pattern matching using metric relations, including the work of Folkers *et al.* and Papadias *et al.*, and more recently Fang *et al.*, Chen *et al.*, and Minervino *et al.*.

Topological relations

Topological relations characterize the relative position of spatial entities with invariance under transformations like translation, scaling, and rotation. Topological relations are typically binary (e.g., a road intersects another road, or it does not). Most approaches support the nine topological relations defined by the Region Connection Calculus (RCC) and the 9-intersection model, which are standardized in GeoSPARQL [Battle and Kolas, 2011]. Formally, a topological relation query Q_τ can be specified using a topological operator $\tau \in \{$intersect, overlap, meet, disjoint, equal, inside, contains, covers, covered by$\}$ as

$$Q_\tau(D) = \{x, y \in D \mid \langle x, y\rangle \in \tau\},$$

Relation	Domain	Example Use Case	Implementation Examples
Metric	Housing Search	"Which homes for sale are within 10km of a school and greater than 5km from a highway?"	[Chen *et al.*, 2019]
	Itinerary Planning	"Find a (library, museum, restaurant) or a (museum, library, restaurant) where the distance from one place to the next is less than 400m."	[Fang *et al.*, 2019]
Topological	Road Network Alignment	"Given a sketch drawing of street intersections and highways, which cities are a match?"	[Schwering *et al.*, 2014]
	Multi-scale Data Fusion	"Which other datasets at different scales are a spatial match to this one?"	[Huang *et al.*, 2023]
Directional	Landmark Search	"Which restaurant has an ocean to the East, a parking lot to the North, and a golf course to the South?"	[O'Sullivan *et al.*, 2023b]
Order	Urban Planning	"Which theme parks have visitors encounter a ticket booth, then a snack stand, then a ride, then a gift shop, then an exit, in clockwise order?"	[Schwering *et al.*, 2014]
Hybrid	Scene Recognition	"Given an aerial image with detected objects tagged, where is this place?"	[Minervino *et al.*, 2023] [Deng *et al.*, 2017]
	Similar Region Search	"Which places have a similar POI arrangement to this example city district?"	[Chen *et al.*, 2022] [Liu *et al.*, 2018]

Table 1: Situated spatial relations.

where x and y represent a pair of search candidates in database D. Most of the surveyed methods that support matching over topological relations do so over point, line, and region entities.

Directional relations

Directional relations characterize how entities are positioned with respect to one another based on a global coordinate system or local reference point. For example, a sign, statue, and fountain may be arranged with the sign to the Northwest, the statue directly East of the sign, and the fountain Southeast of both. Directional relations can be measured using cardinal directions, local directions like left and right, or angular ranges. Formally, a directional relation query Q_δ can be specified using a directional operator δ as

$$Q_\delta(D) = \{x, y \in D \mid \langle x, y\rangle \in \delta\},$$

where δ represents a cardinal direction, like North or Southeast, or an angular range like $[0°, 90°)$, and x and y represent a pair of search candidates in database D. Directional relations can be **cardinally invariant**, meaning they do not rely on a global coordinate system (i.e. a sketch map query can be drawn without knowing which direction is North). Most recent work in directional spatial pattern matching supports either point entities or all entity types.

Order relations

Order relations encode how three or more entities appear with respect to a central reference point (i.e. centroid). Formally, an order relation query Q_ω can be specified using a order operator $\omega \in \{\text{counterclockwise, clockwise}\}$ as

$$Q_\omega(D) = \{x, y, z \in D \mid \langle x, y\rangle, \langle y, z\rangle, \langle z, x\rangle \in \omega\},$$

where x, y, and z represent three search candidates in database D. This definition can be trivially extended beyond three search candidates by adding additional order constraints to the right hand side. Order relations are less commonly used than the other types of spatial relations, with only one of the methods we survey supporting them [Schwering *et al.*, 2014].

Hybrid relations

Hybrid relations combine two or more of the primitive spatial relations [Carniel, 2023]. For example, a lake may be *East* of an ocean and *contain* an island. Formally, a hybrid relation query Q_η can be specified using spatial operator $\eta \in \mathbb{P}(\{\mu, \tau, \delta, \omega\})$ as

$$Q_\eta(D) = \{x, y \in D \mid x, y \in \bigcap_{i=1}^{|\eta|} Q_{\eta_i}\},$$

where $\mid \eta \mid \geq 2$ and x and y represent a pair of search candidates in database D. Following the earlier example, the query for a lake East of an ocean and containing an island would take the form $Q_{\{\delta,\tau\}}(D) = \{x, y \in D \mid x, y \in Q_\delta \bigcap Q_\tau\}$, where Q_δ represents the directional query for a lake East of an ocean, Q_τ represents the topological query for a lake containing an island. The majority of the methods we survey support hybrid queries, usually involving metric or directional relations, with few involving topological or order relations.

3.3 Spatial Pattern Matching Problem

Formally, spatial pattern matching is the process of matching a query Q consisting of q spatial entities and m relation constraints to corresponding entities in a database D consisting of n entities spanning r regions. Queries can be specified in many forms, including by identifying an example pattern in the database, creating a pictorial representation, drawing a sketch map, or constructing a graph, matrix, or tensor of desired entities and relations. Depending on the encoding method, the search process can take on many possible forms. If entities are stored in a graph, spatial pattern matching can be done using Depth First Search (DFS), Bipartite Matching (BPM), subgraph matching (SGM), or by solving a Constraint Satisfaction Problem (CSP). If entities are stored in a matrix or tensor, search can be done using branch and bound algorithms or recursive grid search. In some cases, methods can handle uncertain constraints or perform matching within some error tolerance (we term these **fuzzy methods**).

Notation	Explanation
D	Database of spatial entities and relations
n	Number of entities in database D
r	Number of regions to search in database D
n'	A pruned subset of all possible entities n
Q	Query pattern of spatial entities and relations
q	Number of entities in query Q
m	Number of relations in query Q
ζ	A sampling threshold in $[0, 1]$
μ	Metric spatial operator
δ	Directional spatial operator
τ	Topological spatial operator
ω	Order spatial operator
η	Hybrid spatial operator

Table 2: Summary of Notation.

4 Surveyed Approaches

In this section we present the major spatial pattern matching methods organized by their query and encoding formats. For each query format we describe the common encoding and search methods that use it, highlighting the benefits and drawbacks of each. In Table 3 we summarize each method in greater detail, including the entity and relation types it supports, whether it uses Machine Learning (ML), and its theoretical worst case time complexity. We derive the time complexities by doing asymptotic analysis of the algorithms presented by each paper. In some cases, authors provide the theoretical worst case in their papers. In those cases we validate their derivations and standardize the variables to match our notation, which is defined in Table 2.

4.1 Example-based Query Approaches

Spatial pattern matching queries can be specified using an example pattern that the user identifies from the database. The methods that use example-based queries allow the user to select example entities from a visual interface, implicitly encoding them in a graph with the relation constraints inferred from the distance and/or directional relationships between the example entities selected. To perform the search, these methods use depth first search or bipartite matching to identify patterns with similar metric or directional configurations to the query pattern [Chen *et al.*, 2022; Luo *et al.*, 2017; Zhang *et al.*, 2022]. Example-based input format is well suited for the similar region search use case of spatial pattern matching, where the user wants to find a region similar to an existing region. However, it is not well suited to other use cases, like location recall and landmark search, where the user recalls a place but cannot remember its address and would like to find it by its spatial configuration. In that case it would be difficult to specify an example region to use as the query, since there may be no similar regions to identify it by.

4.2 Graph-based Query Approaches

Graphs are a natural way to encode spatial queries, as evidenced by GeoSPARQL [Battle and Kolas, 2011], which standardizes the core spatial relationships we discuss in this survey. Moreover, some spatial pattern matching methods take query input directly in graph form [Chen *et al.*, 2019; Minervino *et al.*, 2023; Duckham *et al.*, 2023; Liu *et al.*, 2003]. These papers focus on computational efficiency, glossing over the step where a user's search intention is realized as a query graph. Methods like ESPM and QQESPM provide efficient solutions for graph-based metric and topological spatial pattern matching on small query graphs (up to 6 entities) and large databases (upwards of 10 million entities) [Chen *et al.*, 2019; Minervino *et al.*, 2023]. Their efficiency comes from a pruning step, where inverted linear quadtrees are used to eliminate unpromising candidates early. However, if the pruned set of entities n' is not much smaller than the full set n, the worst case performance is exponential. Further, to use these methods in practice, users must specify queries by defining a graph, which may be inconvenient or unnatural for some use cases.

4.3 Sketch-based Query Approaches

Although less common, it is possible to specify a spatial pattern matching query using a sketch drawing. While this input format aligns well with some of the use cases, like searching for a neighborhood based on a sketch of its road network layout, it requires a drawing mechanism for the user to sketch their search intention. Additional steps are also needed to parse the sketch to interpret both the spatial entities themselves, and the relationships between them that are implied by the drawing. Schwering *et al.* proposes SketchMapia, a system designed to support sketch map input for a limited set of primitive spatial entities (street segments and city block regions) for road network alignment [Schwering *et al.*, 2014]. Their framing of the problem limits the number of valid topological, directional, and order relations that can be supported, leading to a computationally efficient system requiring only quadratic processing time to resolve queries. However, their method does not easily generalize to the broader spatial pattern matching problem.

4.4 Pictorial Query Approaches

The majority of the spatial pattern matching methods we survey employ a pictorial query interface, in which a user constructs a query by positioning point, line, and/or region entities onto a canvas. This query format is simpler to process than a sketch map drawing, since the interface captures the entities and their placement during query construction. Depending on the system, relations between the entities can be inferred through the implicit distance, topological, or directional configuration of the entities, or they can be explicitly encoded by the user through drop down menus or typing. Similar to sketch map input, pictorial queries require a custom interface to enable the user to capture their search intention, and that interface is typically tailored to one particular use case. Examples include the GESTALT interface used by COMPASS$_{OO}$ and COMPASS$_{CI}$ to capture implicit directional landmark search queries [O'Sullivan *et al.*, 2023b] and the Spacekey interface used by Multi-Pair and Multi-Star Join to capture explicit metric queries [Fang *et al.*, 2019; Fang *et al.*, 2018]. Pictorial spatial pattern matching methods have many encoding types since a pictorial query can naturally be converted into a graph by turning the entities into nodes and the relations into edges, or a matrix by mapping

Method Type			**Entity & Relations**					**Other Features**						
Query	**Encoding**	**Search**	**Entity***	**Metric**	**Topological**	**Directional**	**Order**	**Fuzzy**	**Approx.**	**Card. Inv.**	**ML-based**	**Name**	**Citation**	**Complexity**
Example	Graph	DFS	*P*	✓				✓	✓			SEQ	[Luo *et al.*, 2017]	$\mathcal{O}(mn^2)^\dagger$
			P	✓		✓		✓		✓		ESS_{HSP}	[Zhang *et al.*, 2022]	$\mathcal{O}(mn^2)^\dagger$
			P	✓		✓		✓	✓	✓		ESS_{LORA}	[Zhang *et al.*, 2022]	$\mathcal{O}(\zeta mn^2)^\dagger$
		BPM	*P*	✓		✓		✓		✓		EPM	[Chen *et al.*, 2022]	$\mathcal{O}(q^2n^2)$
Graph	Graph	SGM	*P*	✓				✓				ESPM	[Chen *et al.*, 2019]	$\mathcal{O}(n'^q)$
			P,L,R	✓	✓			✓				QQESPM	[Minervino *et al.*, 2023]	$\mathcal{O}(n'^q)$
		CSP	*P*			✓		✓				QSR-UE	[Duckham *et al.*, 2023]	$\mathcal{O}(2^n)^\dagger$
			P			✓				✓		StarVars	[Lee *et al.*, 2013]	$\mathcal{O}(m^n)^\dagger$
	Tree	Greedy	*P*	✓		✓		✓	✓	✓		CSQ	[Deng *et al.*, 2017]	$\mathcal{O}(nm)^\dagger$
Sketch	Graph	CSP	*P,L,R*		✓	✓	✓	✓	✓	✓		SketchMapia	[Schwering *et al.*, 2014]	$\mathcal{O}(n^2m^2)^\dagger$
Pictorial	Graph	SGM	*P*	✓		✓						checkSsl	[Soffer and Samet, 1998]	$\mathcal{O}(n^q)$
			P	✓		✓						McheckSsl	[Soffer and Samet, 1998]	$\mathcal{O}(2^n)$
			P	✓		✓				✓		PQIS	[Folkers *et al.*, 2000]	$\mathcal{O}(n2^n)$
			P	✓				✓				M-Pair Join	[Fang *et al.*, 2019]	$\mathcal{O}(m\zeta n^2 + n^q)$
			P	✓				✓				M-Star Join	[Fang *et al.*, 2019]	$\mathcal{O}(n^4 + mn^2 + n^q)$
		CSP	*P,L,R*	✓	✓	✓		✓				MSJ	[Papadias *et al.*, 1998]	$\mathcal{O}(n^q)^\dagger$
			P,L,R	✓	✓	✓		✓				MSJ_{WR}	[Papadias *et al.*, 1998]	$\mathcal{O}(n^m)^\dagger$
			P,L,R	✓	✓	✓		✓				MSJ_{JWR}	[Papadias *et al.*, 1998]	$\mathcal{O}(n^m)^\dagger$
	Matrix	Grid Search	*P*			✓		✓				COMPASS_{OO}	[O'Sullivan *et al.*, 2023b]	$\mathcal{O}(rq + rn^2)$
			P			✓		✓		✓		COMPASS_{CI}	[O'Sullivan *et al.*, 2023b]	$\mathcal{O}(rq^2 + rqn^2)$
	Tensor	Branch/Bound	*P*	✓		✓		✓	✓	✓	✓	E-SRS	[Liu *et al.*, 2018]	$\mathcal{O}(r \log r)$

*Entity types are represented by *P*, *L*, and *R* for Point, Line, and Region, respectively
† Cases where we derive the worst-case complexity since it is not provided in the original paper

Table 3: Spatial pattern matching method comparison.

locations into discrete bins. Tensor, matrix, or graph encodings of pictorial queries are good candidates for AI integration since models like neural networks already operate on those kinds of input. One method for similar region search, E-SRS, encodes pictorial queries with implicit metric and directional relations in tensors and performs branch and bound search using a Convolutional Neural Network (CNN) to rank the similarity between the query tensor and candidate region tensors [Liu *et al.*, 2018]. Since the matching is done using a CNN, the complexity grows with the spatial footprint covered, not with the number of database objects, making the method efficient for small dense regions. [2] E-SRS is a prime example of how ML can be used to approximately solve spatial pattern matching problems.

5 Discussion

We identify several challenges inhibiting the widespread adoption of spatial pattern matching in generalized search systems (Table 4) and discuss for each the associated research opportunities and situations where AI may be applicable.

[2]Their experiments test regions 0.5 to 2 times the size of the query region

5.1 C1: Noisy and Incomplete Geospatial Data

Challenge. Datasets including geographic entities and relations are often noisy and incomplete since they are typically crowd-sourced, leading to incorrectly placed markers, missing entities and unknown or mismatched relationships.

Open Problem: Inferring missing spatial relations. Many spatial pattern matching methods work with graph-based input. But geospatial datasets are often noisy, meaning these graphs have missing or incorrect entities and relations. A large body of work has been dedicated to addressing noise in generic knowledge graphs, but little attention has been given to geographic knowledge graphs. Successful generic knowledge graph completion methods typically use rule-based, translational, or deep learning methods to infer missing links based on statistical properties of the surrounding graph [Meilicke *et al.*, 2025; Liu *et al.*, 2021; Betz *et al.*, 2024; Wang *et al.*, 2014]. R-GCN, for example, uses graph convolution to reconstruct a 3-way tensor representation of a graph [Schlichtkrull *et al.*, 2018]. However, such methods do not perform well at inferring geographic relations, since they do not account for the inherent geospatial nature of the graph. Some work has begun to address this problem, like Geo-KGE [Hu *et al.*, 2024], which learns over both relation information and geo-coordinates but,

Challenge	Open Problem	Work in the Area	Limitation
Noisy geospatial knowledge stores	Benchmark knowledge graph completion methods on geospatial knowledge graphs	N/A	N/A
	Develop geo-aware knowledge graph completion	[Hu *et al.*, 2024; Balsebre *et al.*, 2023]	Closed world or entity prediction only
Computational complexity	Benchmark spatial pattern matching methods	N/A	N/A
	Develop approximate spatial pattern matching methods	[Liu *et al.*, 2018; Zhang *et al.*, 2022; Deng *et al.*, 2017; Luo *et al.*, 2017; Schwering *et al.*, 2014]	Limited relation types
	Optimize spatial pattern matching methods for large queries	[Cao *et al.*, 2024]	Generic subgraph matching only
Heterogeneous entity and relation types supported	Develop efficient spatial pattern matching for line and region queries	N/A	N/A
	Develop ensemble approaches to route queries to the best method	N/A	N/A
Niche query input format	Develop natural language to spatial query methods	[Schneider *et al.*, 2025a; Schneider, 2025]	Point entities and directional relations only

Table 4: Spatial pattern matching challenges and corresponding opportunities.

due to its closed-world problem framing, it does not generalize to situations where the entities of interest are not already seen at training time, which is common in geographic datasets [Balsebre *et al.*, 2023]. Adapting existing BERT-based, graph convolutional, and transformer-based knowledge graph completion methods to geographic knowledge graphs in an open-world setting is a prominent challenge, and one that the AI community is well-positioned to solve.

5.2 C2: Computational Complexity

Challenge. Relationships between spatial entities are numerous, often quadratic or worse in the number of entities, making them computationally difficult to search.

Open problem: Benchmarking. State of the art spatial pattern matching approaches typically use pruning techniques, binning, localization, and approximate ML-based methods to search through candidate entities efficiently. However, the accuracy and efficiency of these approaches can depend on the size of the localized region or pruned entity set, which may vary significantly based on the query, dataset, and method being used. Presently, empirical testing is done using ad-hoc datasets which vary from paper to paper. Without a common benchmark or leader-board, spatial pattern matching methods cannot be rigorously evaluated with respect to one another and the trade-off between accuracy and efficiency of different techniques cannot be fully understood. To elicit worst-case behavior, a spatial pattern matching benchmark should include complex datasets and large query patterns covering the major types of spatial entities and relations. It should also vary the number of distinct entities in the queries and databases, since this can influence the size of the pruned candidate set for some methods, which impacts runtime.

Open problem: Supporting large query patterns. Most ad-hoc datasets used for empirical testing by the approaches we surveyed include query patterns of no more than six entities. Spatial pattern matching over larger query patterns remains an open problem, with most existing methods having a complexity that is exponential in the size of the query. Depending on the search method being used, there are a few possible avenues to enable larger query patterns. Some generic subgraph matching algorithms optimize the order in which query nodes are matched to reduce redundant comparisons of node labels, speeding up the processing time significantly [Cao *et al.*, 2024]. A similar approach could be adapted to SGM-based spatial pattern matching approaches to improve scalability in the query size. Alternatively, approximate approaches like E-SRS use a CNN to determine similarity between the query and candidate regions, a process that scales to include more query entities without incurring the exponential increase in runtime that most exact methods encounter. However, there is a potential cost in accuracy due to the approximate nature of the method. Supporting large query patterns with exact methods remains an open problem.

Open problem: Approximate spatial pattern matching. There are relatively few approximate methods for spatial pattern matching compared to the numerous exact methods [Schneider *et al.*, 2024b]. In the methods we survey, approximation is typically achieved through simplifying the data being searched, or through simplifying the search mechanism. Schwering *et al.* simplifies the data, by localizing the scope of entities and relations considered, only supporting the road network alignment application of spatial pattern matching. The intuition behind their method could be applied to the broader spatial pattern matching landscape by reducing the number of database relations considered during search, perhaps by dynamically choosing the number of neighbors to store for each entity based on the density of the spatial region (i.e. an urban area would require more densely connected nodes than a rural area). To incorporate other factors, a method could be developed to learn which areas need to be stored with greater detail based on an expected query load, similar to learned indexes [Kraska *et al.*, 2018; Gupta *et al.*, 2022]. Taking the alternate approach and reducing the complexity of the search procedure instead of the data, ESS$_{LORA}$, approximates spatial pattern matching over point entities by first grouping database entities with similar attributes together to find rough matches to the query pattern, then refining those results [Zhang *et al.*, 2022]. A similar tactic may be applicable to line and region entities, if an appropriate aggregation technique could be devised to preserve the spatial properties needed for accurate search. Other road network alignment methods may also prove inspiring. SP-GEM trains a GNN to classify a collection of line entities into broad categories, including grid, irregular grid, and circular patterns to aid in road network matching [Zheng *et al.*, 2025]. In its present state, SP-GEM does not address the spatial pattern matching problem, but the idea is promising, especially for spatial pattern queries that include line entities. Adapting such a method to approximately solve spatial pattern matching may prove fruitful. With proper benchmarks in place, new approximate solutions can be evaluated to determine if their efficiency gains outweigh the loss in accuracy they incur.

5.3 C3: Heterogeneous Entity/Relation Types

Challenge. Different spatial entity and relation types present a challenge to reason about concurrently, making it difficult for any one method to support all of them without having a high computational complexity.

Open problem: Efficiently handling non-point queries. The vast majority of methods we survey are designed for point entities, which can be stored with just a single coordinate, making them easy to evaluate. As a result, many of the fastest spatial pattern matching methods only support metric relations over point data, or scope the problem to a subset of entity and relation types, with limited broader applicability. While substantial progress has been made at efficiently solving some forms of spatial pattern matching, few methods are capable of handling most or all entity and relation types, and those that handle multiple tend to be inefficient, with processing time growing exponentially with the number of database or query entities. More work is needed to develop methods that efficiently perform spatial pattern matching over line and region data and can handle matching on topological and order relations in addition to metric and directional ones.

Open problem: Flexibility across use cases. Most spatial pattern matching algorithms are designed for particular systems, which are tailored to a small set of use cases, such as similar region search [Liu *et al.*, 2018], landmark search [O'Sullivan *et al.*, 2023b], or road network alignment [Schwering *et al.*, 2014]. The lack of support across all entity and relation types prevents any one method from generalizing beyond a narrow range of spatial pattern matching applications. To integrate spatial pattern matching into a general search pipeline, a flexible method is needed that can resolve a wide variety of queries efficiently enough to enable real time search. Even if individual methods can be made more efficient, an ensemble approach may be needed, where different queries are offloaded to different methods based on the entities and constraints in the query. While such an approach has yet to be explored in the spatial pattern matching literature, this is a common paradigm in meta-learning [Hospedales *et al.*, 2022; Gharoun *et al.*, 2024]. Applying it here, one might devise a classification model that learns which method is most appropriate for a given query based on previous examples.

5.4 C4: Query Input Format

Challenge. To perform spatial pattern matching, a query pattern must be specified with desired entities and relations, which requires some mechanism for construction.

Open problem: Handling multi-modal queries. Most systems that perform spatial pattern matching take input queries in the form of a drawn sketch map or pictorial query constructed on a canvas. While these options offer exceptional expressivity to capture the user's search intention, they require a custom interface and specific processing of the query before spatial pattern matching can be performed. On the other hand, some methods directly take a graph of query objects with nodes and edge labels as input, or construct a graph from a user selected set of example entities. While these approaches simplify the input process, they do not handle multiple modalities which could allow the user to capture the search intention in any convenient manner. Moreover, each query input format we identify in this survey has one limitation or another that makes it difficult to incorporate into a general search pipeline. As a promising way forward, recent work has explored issuing directional spatial pattern matching queries via natural language, using large language models to generate pictorial queries automatically, given a text description of the desired spatial pattern [Schneider *et al.*, 2025a]. However, this work only addresses point entities and directional relations, and is limited to text input. Expanding text-to-pictorial processing to the broader set of possible entity and relation types and supporting image and other query modalities is an open area of research that could benefit from recent developments in large language models and vision language models. If pictorial queries can be generated for arbitrary queries, they can then be encoded in a graph, tensor, or matrix to enable exact or approximate spatial pattern matching using any of the methods we surveyed.

6 Conclusion

Spatial pattern matching is a powerful form of spatial search that allows users to capture precise spatial search intentions. State of the art spatial pattern matching algorithms are capable of answering questions that most search engines cannot currently handle, but their widespread use in general search scenarios is inhibited by several challenges that we identify and discuss in this paper. We survey the prominent exact and approximate spatial pattern matching methods and discuss their benefits and drawbacks. Based on these insights, we highlight the opportunities where AI can be used to address the challenges associated with spatial pattern matching to enable spatial search more broadly in the future.

References

[Balsebre *et al.*, 2023] Pasquale Balsebre, Dezhong Yao, Gao Cong, Weiming Huang, and Zhen Hai. Mining geospatial relationships from text. *Proc. ACM Manag. Data*, 1(1), May 2023.

[Battle and Kolas, 2011] Robert Battle and Dave Kolas. Geosparql: enabling a geospatial semantic web. *Semantic Web Journal*, 3(4):355–370, 2011.

[Betz *et al.*, 2024] Patrick Betz, Luis Galárraga, Simon Ott, Christian Meilicke, Fabian Suchanek, and Heiner Stuckenschmidt. Pyclause - simple and efficient rule handling for knowledge graphs. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24, 2024.

[Cao *et al.*, 2015] Xin Cao, Gao Cong, Tao Guo, Christian S. Jensen, and Beng Chin Ooi. Efficient processing of spatial group keyword queries. *ACM Transactions on Database Systems*, 40(2), jun 2015.

[Cao *et al.*, 2024] Hongtai Cao, Qihao Wang, Xiaodong Li, Matin Najafi, Kevin C. Chang, and Reynold Cheng. Large Subgraph Matching: A Comprehensive and Efficient Approach for Heterogeneous Graphs. In *40th IEEE International Conference on Data Engineering, ICDE 2024*, Utrecht, Netherlands, May 2024.

[Carniel, 2020] Anderson Chaves Carniel. Spatial information retrieval in digital ecosystems: A comprehensive survey. In *Proceedings of the 12th International Conference on Management of Digital EcoSystems*, pages 10–17, 2020.

[Carniel, 2023] Anderson Chaves Carniel. Defining and designing spatial queries: the role of spatial relationships. *Geo-spatial Information Science*, pages 1–25, 2023.

[Chen *et al.*, 2019] Hongmei Chen, Yixiang Fang, Ying Zhang, Wenjie Zhang, and Lizhen Wang. Espm: Efficient spatial pattern matching. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1227–1233, 2019.

[Chen *et al.*, 2022] Yue Chen, Kaiyu Feng, Gao Cong, and Han Mao Kiah. Example-based spatial pattern matching. *Proceedings of the VLDB Endowment*, 15(11):2572–2584, 2022.

[Deng *et al.*, 2017] Ke Deng, Huanliang Sun, Yu Ge, Xiaofang Zhou, Christian Søndergaard Jensen, et al. Clue-based spatio-textual query. *Proceedings of the VLDB Endowment*, 10(5):529–540, 2017.

[Duckham *et al.*, 2023] Matt Duckham, Jelena Gabela, Allison Kealy, Ross Kyprianou, Jonathan Legg, Bill Moran, Shakila Khan Rumi, Flora D Salim, Yaguang Tao, and Maria Vasardani. Qualitative spatial reasoning with uncertain evidence using markov logic networks. *International Journal of Geographical Information Science*, 37(9):2067–2100, 2023.

[Dylla *et al.*, 2017] Frank Dylla, Jae Hee Lee, Till Mossakowski, Thomas Schneider, André Van Delden, Jasper Van De Ven, and Diedrich Wolter. A survey of qualitative spatial and temporal calculi: algebraic and computational properties. *ACM Computing Surveys (CSUR)*, 50(1):1–39, 2017.

[Fang *et al.*, 2018] Yixiang Fang, Reynold Cheng, Jikun Wang, Lukito Budiman, Gao Cong, and Nikos Mamoulis. Spacekey: exploring patterns in spatial databases. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 1577–1580. IEEE, 2018.

[Fang *et al.*, 2019] Yixiang Fang, Yun Li, Reynold Cheng, Nikos Mamoulis, and Gao Cong. Evaluating pattern matching queries for spatial databases. *The VLDB Journal*, 28:649–673, 2019.

[Folkers *et al.*, 2000] Andre Folkers, Hanan Samet, and Aya Soffer. Processing pictorial queries with multiple instances using isomorphic subgraphs. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 4, pages 51–54. IEEE, 2000.

[Gharoun *et al.*, 2024] Hassan Gharoun, Fereshteh Momenifar, Fang Chen, and Amir H. Gandomi. Meta-learning approaches for few-shot learning: A survey of recent advances. *ACM Comput. Surv.*, 56(12), July 2024.

[Gupta *et al.*, 2022] Nilesh Gupta, Patrick Chen, Hsiang-Fu Yu, Cho-Jui Hsieh, and Inderjit Dhillon. Elias: End-to-end learning to index and search in large output spaces. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 19798–19809. Curran Associates, Inc., 2022.

[Hospedales *et al.*, 2022] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169, 2022.

[Hu *et al.*, 2024] Lei Hu, Wenwen Li, and Yunqiang Zhu. Geometric feature enhanced knowledge graph embedding and spatial reasoning. In *Proceedings of the 7th SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, GeoAI '24, page 50–53, New York, NY, USA, 2024. ACM.

[Huang *et al.*, 2023] Zhekun Huang, Haizhong Qian, Xiao Wang, Defu Lin, Junwei Wang, and Limin Xie. Graph neural network-based identification of ditch matching patterns across multi-scale geospatial data. *Geocarto International*, 38(1):2294900, 2023.

[Kraska *et al.*, 2018] Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, page 489–504, New York, NY, USA, 2018. ACM.

[Lee *et al.*, 2013] Jae Hee Lee, Jochen Renz, and Diedrich Wolter. Starvars: effective reasoning about relative directions. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, page 976–982. AAAI Press, 2013.

[Liu *et al.*, 2003] Xuan Liu, Shashi Shekhar, and Sanjay Chawla. Object-based directional query processing in spatial databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):295–304, February 2003.

[Liu *et al.*, 2018] Yiding Liu, Kaiqi Zhao, and Gao Cong. Efficient similar region search with deep metric learning. In *Proceedings of the 24th SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1850–1859. ACM, 2018.

[Liu *et al.*, 2021] Xiyang Liu, Huobin Tan, Qinghong Chen, and Guangyan Lin. Ragat: Relation aware graph attention network for knowledge graph completion. *IEEE Access*, 9:20840–20849, 2021.

[Luo *et al.*, 2017] Siqiang Luo, Jiafeng Hu, Reynold Cheng, Jing Yan, and Ben Kao. Seq: Example-based query for spatial objects. In *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*, pages 2179–2182, 2017.

[Mahmood and Aref, 2017] Ahmed Mahmood and Walid G. Aref. Query processing techniques for big spatial-keyword data. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, page 1777–1782, New York, NY, USA, 2017. ACM.

[Mahmood and Aref, 2019] Ahmed R. Mahmood and Walid G. Aref. *Querying Spatial-Keyword Data*, pages 7–23. Springer International Publishing, Cham, 2019.

[Meilicke *et al.*, 2025] Christian Meilicke, Melisachew Wudage Chekol, Patrick Betz, Manuel Fink, and Heiner Stuckenschmidt. Correction: Anytime bottom-up rule learning for large-scale knowledge graph completion. *The VLDB Journal*, 34(4), 2025.

[Minervino *et al.*, 2023] Carlos Minervino, Claudio Campelo, Maxwell Oliveira, and Salatiel Silva. Qqespm: A quantitative and qualitative spatial pattern matching algorithm. *arXiv preprint arXiv:2312.08992*, 2023.

[O'Sullivan *et al.*, 2023a] Kent O'Sullivan, Nicole R. Schneider, Aleeza Rasheed, and Hanan Samet. Gestalt: Geospatially enhanced search with terrain augmented location targeting. In *Proceedings of the 2nd SIGSPATIAL International Workshop on Searching and Mining Large Collections of Geospatial Data*, GeoSearch'23. ACM, 2023.

[O'Sullivan *et al.*, 2023b] Kent O'Sullivan, Nicole R. Schneider, and Hanan Samet. Compass: Cardinal orientation manipulation and pattern-aware spatial search. In *Proceedings of the 2nd SIGSPATIAL International Workshop on Searching and Mining Large Collections of Geospatial Data*, GeoSearch'23. ACM, 2023.

[Papadias *et al.*, 1998] Dimitrios Papadias, Nikos Mamoulis, and B Delis. Algorithms for querying by spatial structure. In *Proceedings of Very Large Data Bases Conference (VLDB), New York*, 1998.

[Purves *et al.*, 2018] Ross S. Purves, Paul Clough, Christopher B. Jones, Mark H. Hall, and Vanessa Murdock. Geographic information retrieval: Progress and challenges in spatial search of text. *Foundations and Trends in Information Retrieval*, 12(2-3):164–318, 02 2018.

[Schlichtkrull *et al.*, 2018] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

[Schneider *et al.*, 2024a] Nicole R. Schneider, Kent O'Sullivan, and Hanan Samet. Graph-based spatial pattern matching: A theoretical comparison. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '24, page 505–508, New York, NY, USA, 2024. ACM.

[Schneider *et al.*, 2024b] Nicole R. Schneider, Kent O'Sullivan, and Hanan Samet. The Future of Graph-based Spatial Pattern Matching (Vision Paper). In *40th International Conference on Data Engineering, ICDE – SEAGraph Workshop*, pages 360–364, Utrecht, Netherlands, May 2024. IEEE.

[Schneider *et al.*, 2025a] Nicole R Schneider, Avik Das, Kent O'Sullivan, and Hanan Samet. Search by spatial query: Text to pictorial queries. In *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '25, page 804–807, New York, NY, USA, 2025. ACM.

[Schneider *et al.*, 2025b] Nicole R Schneider, Kent O'Sullivan, and Hanan Samet. Graph-enhanced large language models for spatial search. *Proceedings of the VLDB Endowment. ISSN*, 2150:8097, 2025.

[Schneider, 2025] Nicole R Schneider. Spatial pattern search through language models. In *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '25, page 1258–1261, New York, NY, USA, 2025. ACM.

[Schwering *et al.*, 2014] Angela Schwering, Jia Wang, Malumbo Chipofya, Sahib Jan, Rui Li, and Klaus Broelemann. Sketchmapia: Qualitative representations for the alignment of sketch and metric maps. *Spatial cognition & computation*, 14(3):220–254, 2014.

[Soffer and Samet, 1998] Aya Soffer and Hanan Samet. Pictorial query specification for browsing through spatially referenced image databases. *Journal of Visual Languages & Computing*, 9(6):567–596, 1998.

[Wang *et al.*, 2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1112–1119. AAAI Press, 2014.

[Zhang *et al.*, 2022] Hanyuan Zhang, Siqiang Luo, Jieming Shi, Jing Nathan Yan, and Weiwei Sun. Example-based spatial search at scale. In *38th International Conference on Data Engineering (ICDE)*, pages 539–551. IEEE, 2022.

[Zheng *et al.*, 2025] Chenghao Zheng, Yunfei Qiu, Jian Yang, Bianying Zhang, Zeyuan Li, Zhangxiang Lin, Xianglin Zhang, Yang Hou, and Li Fang. Sp-gem: Spatial pattern-aware graph embedding for matching multi-source road networks. *ISPRS International Journal of Geo-Information*, 14(7), 2025.