

LORY-J: Location-aware Join Discovery from Data Lakes

Lu Xing
Purdue University
xingl@purdue.edu

Walid G. Aref
Purdue University
aref@purdue.edu

Mir Mahathir Mohammad
University of Utah
mahathir.mohammad@utah.edu

El Kindi Rezig
University of Utah
elkindi.rezig@utah.edu

Abstract

Data lakes enable flexible, ingestion-first data management, but their lack of global schemas makes it difficult for users to discover datasets across heterogeneous tables. While prior systems focus on relational join discovery, they largely ignore geo-spatial semantics, despite geographic attributes being pervasive in tabular data. As a result, users cannot express or discover joins based on fundamental spatial relationships, e.g., proximity, containment, or intersection, to discover datasets that are co-located in space. We demonstrate LORY-J, a system for location-aware join discovery in tabular data lakes. LORY-J treats spatial attributes as first-class signals and enables the discovery of hybrid join paths that combine equi- and semantic-joins with spatial joins. Through an interactive, map-driven interface, users specify spatial constraints to discover a dataset. The demo showcases how LORY-J supports end-to-end dataset construction by discovering, visualizing, and materializing join paths driven by both relational and spatial intent—without requiring prior knowledge of table schemas or join keys. A companion video is available at [1].

1 Introduction

Driven by the “store now, query later” model, data lakes have become pervasive across domains. Unlike traditional databases, data lakes consist of large collections of heterogeneous tables with little to no upfront curation and no global schema—for example, primary key–foreign key relationships are not provided [7]. As a result, users often lack the knowledge required to formulate meaningful queries (e.g., how to join tables). Despite the rapid adoption of data lakes, effective data discovery within them remains in its infancy, posing a significant barrier to their practical usability.

While recent work has made progress on data discovery in data lakes [4], location data is largely treated as a second-class citizen, despite being pervasive in tabular datasets (e.g., ZIP codes, addresses, GPS coordinates). This omission limits two key capabilities: map-driven and co-located data discovery using geographic constraints, and the discovery of spatial joins based on spatial relationships, e.g., proximity, containment, or intersection, that are central to many real-world analytical tasks.

In this demo, we demonstrate LORY-J (Location-aware data discovery for joins) that builds on SEMDISC [9] to provide native support for spatial joins, elevating location to a first-class signal in join discovery. Building on this extension, we realize a core component of our spatial data discovery vision [11] by enabling location-aware join discovery that automatically identifies and composes join paths integrating equi- and semantic-joins with spatial joins.

It is important to note that extending join paths with spatial joins is non-trivial. Location information is typically attached to specific entity tables (e.g., a `Schools` table with coordinates). When an equi-join (or semantic join) links such a table to one lacking spatial attributes (e.g., `StudentEnrollment` joined on `school_name`), location semantics are implicitly propagated, enabling subsequent spatial joins (e.g., with a `CrimeStatistics` table) that would otherwise be impossible. Recognizing these transitive spatial relationships requires reasoning about how location context flows through join paths, a capability absent in existing join discovery systems. The following two examples illustrate how location-aware join discovery enables analytical tasks that are out of reach for existing join discovery systems. For simplicity, we use point coordinates to represent locations and equi-joins for relational joins, but LORY-J supports arbitrary spatial representations (e.g., rectangles, polygons, lines) and inherits semantic join discovery from SemDisc [9], where join keys need not match syntactically but refer to the same concept (e.g., `zip_code` and `postal_code`).

Example 1.1 (Location-aware Data Discovery and Integration). A data scientist, Lou, is studying traffic safety in an urban area and wants to construct a dataset that captures traffic accidents occurring near schools, enriched with contextual information from other tables in a data lake. Lou is interested in accidents within a specific region of Chicago and wants to include, for each accident, the corresponding police report details and the nearest hospital.

Lou begins by specifying a spatial region of interest using a bounding box on a map (Figure 1 ①). Based on this spatial constraint, we want the data discovery system to filter the data lake to identify accident records whose locations fall within the selected region (Figure 1 ②), and retrieve the relevant tables from the data lake (Figure 1 ③). LORY-J then discovers an equi-join between the `Accidents` and `PoliceReports` tables based on a shared incident identifier, followed by a spatial join between accident locations and nearby hospitals using a distance-based predicate (e.g., the closest hospital within a given radius). Finally, the resulting dataset is obtained by materializing a *hybrid join path* that combines both relational and spatial joins (Figure 1 ④). This example illustrates the need for location-aware join discovery in data lakes, where datasets can be constructed by discovering join paths driven jointly by attribute similarity and by spatial relationships.

Example 1.2 (Location-aware Data Augmentation). A data scientist, Mary, is analyzing real estate listings and wants to enrich an existing table of housing listings with additional contextual information from a data lake. Each listing includes the geographic location of a property, and Mary is interested in augmenting the

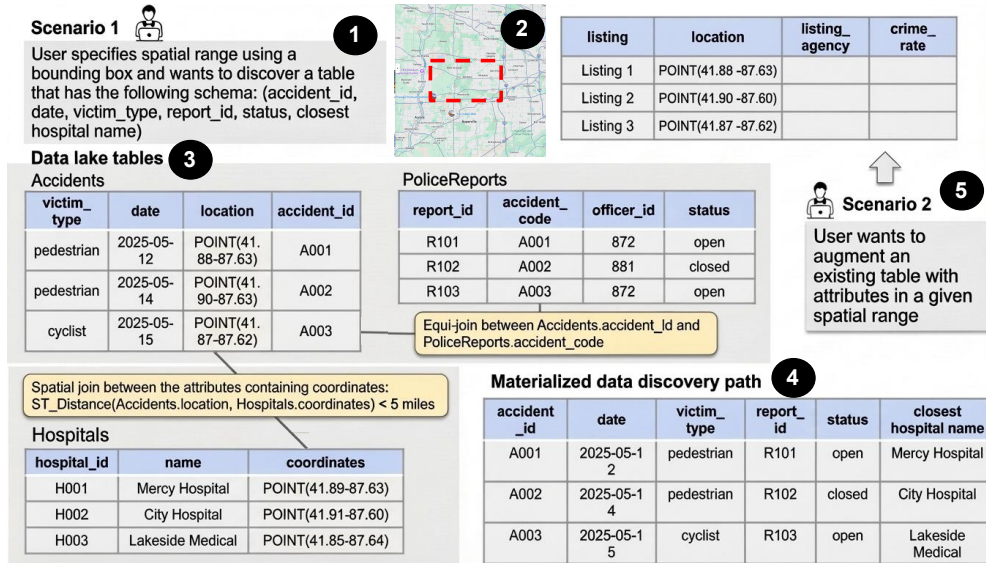


Figure 1: Example location-aware data discovery scenarios. LORY-J discovers hybrid join paths that combine equi-joins and spatial joins to construct new datasets (Scenario 1) or augment existing tables within a spatial range (Scenario 2).

listings with information about the responsible real estate agency (listing_agency) as well as local safety indicators (crime_rate).

Mary starts from a base Listings table and specifies a geographic area of interest using a spatial constraint, e.g., a bounding box over a neighborhood (Figure 1 ⑤). Based on this spatial constraint, LORY-J restricts the listings to those located within the selected region and identifies relevant tables from the data lake. Then, LORY-J discovers an equi-join between the Listings table and another table containing agency information, based on a shared non-spatial attribute (e.g., agency identifier or name). Next, LORY-J identifies a spatial join between the filtered listings and a table containing crime statistics, based on a spatial containment or intersection between listings locations and crime-rate regions.

By composing these equi- and spatial joins, LORY-J produces an augmented dataset enriching each listing with agency metadata and localized crime information, a result unattainable through purely relational join discovery.

Related Work. Data discovery from data lakes has garnered significant attention in recent years [3, 6, 8, 9]. A major line of work focuses on discovering equi-joins across large collections of heterogeneous tables [5, 7, 12]. These systems identify joinable tables based on value overlap or schema similarity, and are complementary to LORY-J. While LORY-J builds on equi-join discovery, LORY-J supports spatial reasoning to provide more expressive and flexible data integration. Nexus [6] addresses the discovery of correlations in spatio-temporal datasets, but primarily relies on equi-joins to compose join paths and does not support spatial join primitives, e.g., k -nearest neighbor (kNN) search, containment, or intersection. Thus, it cannot discover join paths driven by spatial semantics.

2 System Overview

Figure 2 provides an overview of the LORY-J architecture, organized into four stages spanning offline preprocessing and online query answering. Offline, LORY-J identifies columns encoding spatial information (explicitly as coordinates or implicitly as addresses/region

names), geocodes textual locations via a geocoding API¹, and partitions tables by spatial locality into a quadtree [13] (Figure 2 ①). Using this index, LORY-J detects equi-joinability bottom-up via SEMDISC [9]’s Jaccard-based sketch detector and spatial joinability via geometric predicates (intersection, containment, distance) (Figure 2 ②), then composes these into hybrid join paths indexed by spatial extent (Figure 2 ③). At query time (Figure 2 ④), the user specifies target attributes and optional spatial constraints; LORY-J retrieves spatially consistent join paths, prunes candidates, and materializes the highest-ranked paths satisfying both relational and spatial requirements.

2.1 Offline Data Lake Preprocessing

LORY-J performs a set of offline preprocessing steps to prepare the data lake for efficient, location-aware join discovery at query time. These steps extract spatial signals, organize data by geographic locality, and precompute joinability information, allowing online queries to be answered with low latency.

2.1.1 Location Column Detection and Geocoding. First, LORY-J scans the data lake to identify columns that encode location information, either explicitly (e.g., latitude–longitude pairs) or implicitly (e.g., addresses, ZIP codes, or place names). Large Language Models have shown good results in semantic type detection [9]. So, to detect these columns, LORY-J leverages a foundation model (e.g., GPT [2]) to classify columns based on their semantic content. For each table, LORY-J provides the model with the table header along with a small sample of rows, and issues the following prompt template:

Task: Given the table below, which attributes contain location information?

Table Header: [attribute_1, attribute_2, . . . , attribute_n]

Sample Rows (5): [row_1, row_2, row_3, row_4, row_5]

The model’s response is used to mark candidate location columns. For columns that contain textual location descriptors, LORY-J applies geocoding to resolve them to geographic representations. The geocoding component is modular and supports any provider (e.g.,

¹<https://developers.google.com/maps/documentation/geocoding/overview>

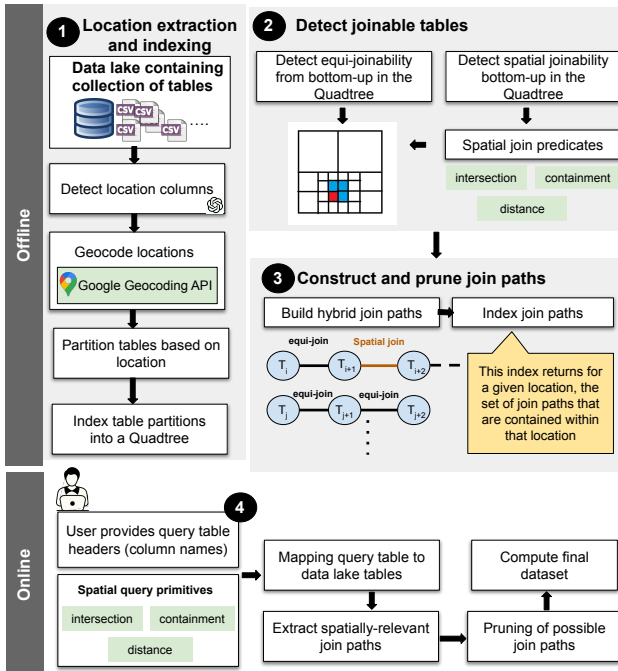


Figure 2: Architecture of LORY-J. LORY-J extracts spatial attributes ①, detects equi- and spatial joins via a quadtree ②, constructs and prunes hybrid join paths ③, and answers queries by materializing relevant datasets ④.

Google Geocoding API). Depending on the granularity of the spatial attribute, locations can be resolved to point coordinates (e.g., for street addresses) or boundary polygons (e.g., for counties or ZIP code regions), enabling more precise spatial predicates such as containment and intersection.

2.1.2 Spatial Partitioning and Indexing. Once spatial representations are extracted, LORY-J partitions tables based on spatial proximity. Tuples are logically assigned to quadtree cells according to their geographic coordinates; tables themselves are not physically split. The quadtree is expanded until each leaf cell contains fewer than τ tuples (configurable; $\tau = 500$ in our demo). This spatial partitioning serves two purposes: (i) it enables efficient pruning of irrelevant data during join discovery, and (ii) it provides a shared spatial hierarchy over which both relational and spatial joinability can be detected bottom-up.

2.1.3 Detecting Relational and Spatial Joinability. Using the spatial index, LORY-J detects joinable tables by jointly reasoning about non-spatial and spatial attributes. Relational (equi-)joinability is identified by analyzing value overlap between non-spatial columns within corresponding spatial partitions. In parallel, spatial joinability is detected by evaluating geometric relationships between spatial attributes, e.g., intersection, containment, or distance-based proximity. Performing these checks bottom-up in the quadtree allows LORY-J to localize joinability and detect join paths at multiple spatial granularities. For example, joins can first be identified among tuples within a fine-grained region (e.g., Boston) and then naturally lifted to coarser regions (e.g., Massachusetts), enabling the construction of join paths that span larger spatial regions.

2.1.4 Hybrid Join Path Construction and Indexing. Based on the detected joinability relationships, LORY-J constructs candidate join paths that combine equi-joins and spatial joins into hybrid join paths. To support efficient query-time retrieval, LORY-J indexes the constructed join paths by their spatial extent. Given a geographic region, this index enables LORY-J to quickly retrieve only those join paths whose spatial footprint lies within the region of interest.

Overall, offline preprocessing shifts the computationally intensive tasks of location extraction, joinability detection, and path construction away from query time, enabling interactive and scalable location-aware join discovery.

2.2 Online Query Answering

At query time, LORY-J leverages the offline indexes and precomputed join paths to support interactive, location-aware join discovery. Online query answering focuses on mapping user intent to data lake tables, retrieving spatially relevant join paths, and efficiently pruning candidate solutions.

2.2.1 Query Specification and Table Mapping. Users initiate a data discovery query by providing a query table header (column names) along with optional spatial constraints, e.g., intersection, containment, or distance-based predicates. LORY-J maps query attributes to semantically compatible columns in the data lake using value overlap and schema similarity, while restricting the search to tables that overlap the specified region.

2.2.2 Join Path Retrieval and Materialization. Based on the query-to-table mapping and spatial constraints, LORY-J retrieves candidate hybrid join paths from the spatially indexed join paths, considering only paths that intersect or lie within the query region. These paths are further pruned using query-specific constraints (e.g., attribute compatibility and estimated result size), and the highest-ranked join paths are materialized to produce the final dataset.

3 Demonstration Plan

We demonstrate LORY-J on four real-world data lakes: (1) **DrugCentral (DC)** [14], an online pharmacological database; (2) two data lakes collected from **Data.gov** [10]: **The U.S. Fish and Wildlife Service (FWS)** that provides ecological and geographic datasets, and the **Centers for Disease Control and Prevention (CDC)** that contains public health records on disease incidence, mortality, and drug usage. VLDB attendees will explore both intra-lake join paths (e.g., linking CDC county-level disease rates to nearby vaccination sites to analyze accessibility gaps) and cross-lake scenarios (e.g., linking CDC asthma hospitalization rates to FWS contaminated habitat sites to study environmental health correlations).

Demonstration Outline. The demonstration follows an end-to-end workflow that illustrates how LORY-J supports interactive, location-aware join discovery in tabular data lakes. Figure 3 highlights the main stages of the demo.

Step 1: Data Lake Ingestion and Offline Indexing. The demo begins by uploading a data lake consisting of heterogeneous tables (Figure 3 ①). Upon ingestion, LORY-J automatically performs offline preprocessing, including detecting location columns, geocoding spatial attributes, partitioning tables by geographic locality, and indexing table partitions and join paths.

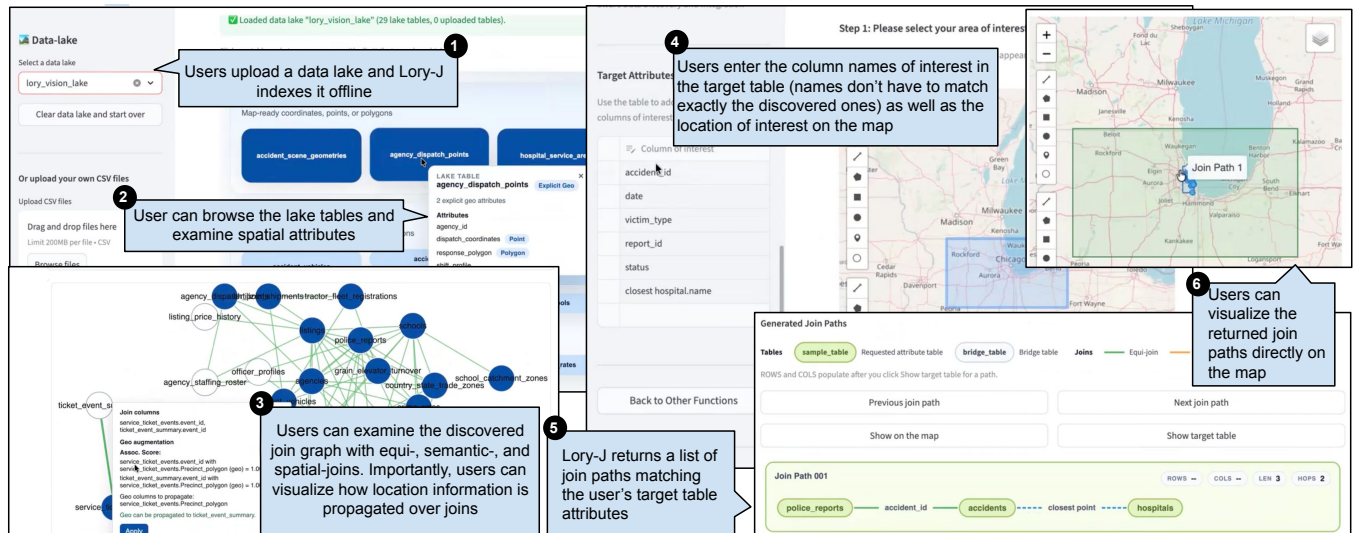


Figure 3: User interface of LORY-J (multiple screens). Users upload a data lake, specify target attributes and a spatial range, explore spatially valid join paths, and visualize the geographic footprint and results of selected join paths.

Step 2: Browsing Tables and Spatial Attributes. Users can browse the ingested tables and examine their spatial attributes (Figure 3 ②). For each table, LORY-J displays detected spatial columns along with their geometry types (e.g., Point, Polygon), allowing users to understand the spatial content of the data lake before querying.

Step 3: Exploring the Join Graph. Users can examine the discovered join graph, which includes equi-joins, semantic joins, and spatial joins (Figure 3 ③). Importantly, the interface visualizes how location information is propagated across joins, showing which tables acquire spatial semantics through their connections to spatially-grounded tables.

Step 4: Query Specification with Spatial Constraints. Users specify a data discovery query by entering target column names and defining a spatial region of interest on the map (Figure 3 ④). Column names need not match the data lake schemas exactly, as LORY-J resolves them via semantic similarity. The spatial constraint guides join discovery and prunes irrelevant portions of the data lake.

Step 5: Exploring Candidate Join Paths. Given the query, LORY-J retrieves and presents a ranked list of candidate hybrid join paths satisfying the spatial constraint (Figure 3 ⑤). Each path displays the involved tables, join types (equi-join or spatial), path length, and number of hops. Users can navigate paths and select those of interest for further exploration.

Step 6: Visualizing Join Paths on the Map. For a selected join path, LORY-J renders its geographic footprint directly on the map (Figure 3 ⑥), showing the spatial distribution of data points from the base tables. This allows users to validate spatial relevance and inspect coverage before materializing the final dataset.

Demonstration Engagement. Beyond a guided walkthrough, the demo allows participants to explore how the data lake is processed and indexed offline. Attendees can examine how location extraction and spatial indexing are performed, and observe how enabling

spatial joins introduces new edges in the join graph that are absent in non-spatial join discovery. This highlights how spatial semantics expand the space of discoverable join paths and enable richer dataset construction in data lakes.

References

- [1] [n. d.]. LORY-J Demo video. https://drive.google.com/drive/u/0/folders/1pWN_LBVw4oajcn9F2cd5OT5gHfc1ZQ65
- [2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florenca Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. arXiv:2303.08774 (2023).
- [3] Martin Pekár Christensen, Aristotelis Leventidis, Matteo Lissandrini, Laura Di Rocco, Renée J Miller, and Katja Hose. 2025. Fantastic Tables and Where to Find Them: Table Search in Semantic Data Lakes. (2025).
- [4] Yuhao Deng, Chengliang Chai, Lei Cao, Qin Yuan, Siyuan Chen, Yanrui Yu, Zhaoze Sun, Junyi Wang, Jiajun Li, Ziqi Cao, et al. 2024. LakeBench: A Benchmark for Discovering Joinable and Unionable Tables in Data Lakes. *Proceedings of the VLDB Endowment* 17, 8 (2024), 1925–1938.
- [5] Raul Castro Fernandez, Ziawasch Abedjan, Famiem Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A data discovery system. In 2018 IEEE 34th International Conference on Data Engineering (ICDE).
- [6] Yue Gong, Sainyam Galhotra, and Raul Castro Fernandez. 2024. Nexus: Correlation Discovery over Collections of Spatio-Temporal Tabular Data. *Proceedings of the ACM on Management of Data* 2, 3 (2024), 1–28.
- [7] Yue Gong, Zhiru Zhu, Sainyam Galhotra, and Raul Castro Fernandez. 2023. Ver: View discovery in the wild. In 2023 IEEE 39th International Conference on Data Engineering (ICDE). IEEE, ICDE, ICDE, 503–516.
- [8] Aamod Khatiwada, Grace Fan, Roe Shruga, Zixuan Chen, Wolfgang Gatterbauer, Renée J Miller, and Mirek Riedewald. 2023. Santos: Relationship-based semantic table union search. *Proceedings of the ACM on Management of Data* 1, 1 (2023).
- [9] Mir Mahathir Mohammad and El Kindi Rezig. 2026. Qualitative Join Discovery in Data Lakes using Examples. In *Proceedings of the 2026 International Conference on Management of Data (SIGMOD)*. ACM. <https://mirmahathir.com/papers/semdisc.pdf>
- [10] The Home of the U.S. Government’s Open Data. 2026. <https://data.gov>.
- [11] El Kindi Rezig and Walid G. Aref. 2025. LORY: Location-aware Data Discovery from Data Lakes. In *Proceedings of the 33rd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM.
- [12] El Kindi Rezig, Anshul Bhandari, Anna Fariha, Benjamin Price, Allan Vanterpool, Vijay Gadepally, and Michael Stonebraker. 2021. DICE: data discovery by example. *Proceedings of the VLDB Endowment* 14, 12 (2021), 2819–2822.
- [13] Hanan Samet. 1990. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA.
- [14] Oleg Ursu, Jayme Holmes, Cristian G Bologna, Jeremy J Yang, Stephen L Mathias, Vasileios Stathias, Dac-Trung Nguyen, Stephan Schürer, and Tudor Oprea. 2019. DrugCentral 2018: an update. *Nucleic acids research* 47, D1 (2019), D963–D970.