GER-LLM: Efficient and Effective Geospatial Entity Resolution with Large Language Model

Haojia Zhu, Zhicheng Li, Jiahui Jin*
School of Computer Science and Engineering, Southeast University
zhuhaojia@seu.edu.cn, lizhicheng@seu.edu.cn, jjin@seu.edu.cn

Abstract

Geospatial Entity Resolution (GER) plays a central role in integrating spatial data from diverse sources. However, existing methods are limited by their reliance on large amounts of training data and their inability to incorporate commonsense knowledge. While recent advances in Large Language Models (LLMs) offer strong semantic reasoning and zero-shot capabilities, directly applying them to GER remains inadequate due to their limited spatial understanding and high inference cost. In this work, we present **GER-LLM**, a framework that integrates LLMs into the GER pipeline. To address the challenge of spatial understanding, we design a spatially informed blocking strategy based on adaptive quadtree partitioning and Area of Interest (AOI) detection, preserving both spatial proximity and functional relationships. To mitigate inference overhead, we introduce a group prompting mechanism with graph-based conflict resolution, enabling joint evaluation of diverse candidate pairs and enforcing global consistency across alignment decisions. Extensive experiments on real-world datasets demonstrate the effectiveness of our approach, yielding significant improvements over state-of-the-art methods. The data and code is available in https://github.com/luck-seu/GER-LLM.

1 Introduction

A high-quality, comprehensive geospatial database is the cornerstone of modern Location-Based Services (LBSs) such as turn-by-turn navigation, ride-hailing, and personalised place recommendation. These services rely on Points of Interest (POIs)—geospatial entities described by spatial attributes (latitude and longitude) and textual attributes (name, address, category). In practice, POI records are scattered across multiple providers (e.g., OpenStreetMap[1], Yelp[2], municipal open data), each covering only a partial and often noisy view of the real world. Consolidating these heterogeneous records into a single, clean dataset is therefore indispensable.

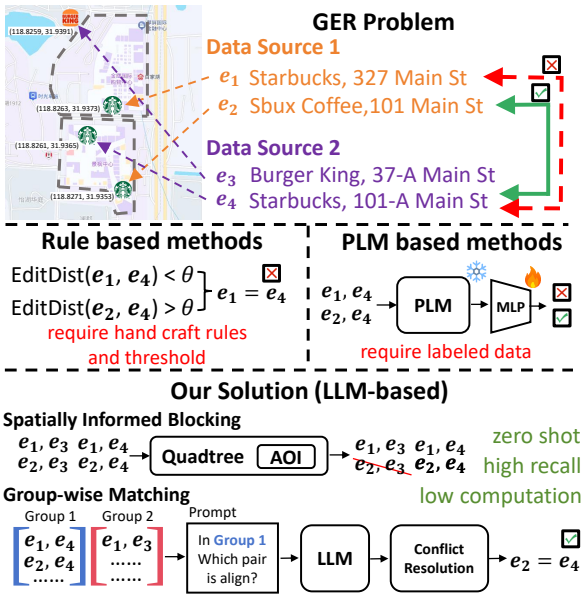

Figure 1: An example of GER problem and comparison of previous GER and GER-LLM.

Geospatial Entity Resolution (GER) is one of the core components of data integration. It aims to find which spatial entities belong to the same physical entity (Sehgal et al., 2006; Isaj et al., 2022). Figure 1 shows an example of GER, where four POI records taken from different providers describe POIs located only meters apart. Correctly merging e_2 ("Sbux Coffee, 101 Main St"), e_4 ("Starbucks, 101-A Main St"), while keeping e_1 ("Starbucks, 327 Main St") separate requires the model to resolve brand abbreviations ("Sbux") and to understand that "101" and "101-A" denote the same storefront. Such judgments require the model to possess commonsense knowledge. However,

*Corresponding author. Email: jjin@seu.edu.cn

[1]https://www.openstreetmap.org/
[2]https://www.yelp.com/

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 23261–23277
November 4-9, 2025

early approaches relied on token-level similarity and hand-crafted rules or thresholds (Isaj et al., 2022; Deng et al., 2019; Shivaprabhu et al., 2017), which limit their capacity to understand entity semantics or handle subtle attribute variations. Recent advances have turned to Pre-trained Language Models (PLMs) (Devlin et al., 2019; Liu et al., 2019), framing GER as a sequence-pair classification task (Balsebre et al., 2022, 2023). While these models offer improved semantic information, they remain blind to spatial context and require large amounts of labeled GER data for effective fine-tuning—resources that are often scarce in practice.

Recent Large Language Models (LLMs) have showcased remarkable abilities in many fields. Trained on trillions of tokens, they implicitly encode much of the background knowledge that GER depends on, enabling them to make informed predictions even in the absence of task-specific labeled data. A growing body of work has therefore begun leveraging LLMs for general entity-resolution tasks (Narayan et al., 2022; Peeters et al., 2025; Steiner et al., 2024; Li et al., 2024; Fan et al., 2024; Wang et al., 2025). However, directly applying these LLM-based ER methods to GER remains inadequate for two key reasons: (1) **Shortage in spatial information understanding.** Existing LLM-based ER methods typically overlook the additional information provided by spatial relationships. They mostly include pipelines that treat each record as text and ask the model to decide whether two descriptions match. Yet spatial cues are often numerical (lat/lon) rather than linguistic. Current LLMs excel at narrative context but are poor at metre-level distance calculation or coordinate comparison (Yang et al., 2025). (2) **Limited scalability due to pairwise comparison overhead.** GER involves exhaustively comparing all possible entity pairs to identify potential matches, resulting in an inherent $O(N^2)$ candidate space where N is the number of entities. While feasible for small datasets, this approach becomes impractical at the city scale, where millions of POIs translate into billions of candidate pairs. Even when candidate filtering or blocking techniques are applied, the number of pairs remains extremely large (Papadakis et al., 2016). Since each LLM invocation incurs significant computational time and cost, applying the model to all pairs would lead to prohibitive runtime and resource demands.

To address these challenges, we propose **GER-LLM**, a novel framework that integrates spatial-aware pre-processing with LLM-based matching. Our approach consists of two key components: (1) A spatially informed blocking strategy that simultaneously reduces the candidate space while leveraging both spatial proximity and functional relationships (e.g., shared Areas of Interest). (2) A group-wise matching pipeline that jointly evaluates diverse candidate pairs to reduce LLM inference cost. To ensure consistent global alignment, we further introduce a graph-based conflict resolution mechanism. Together, these components enable accurate and scalable entity resolution with minimal supervision.

Contributions:

- We present the first framework to integrate LLMs into Geospatial Entity Resolution (GER).
- We develop GER-LLM, a framework that combines spatially informed blocking and group-wise LLM matching.
- We conduct extensive experiments on GER-LLM and discuss the value and limitations of LLMs in entity alignment tasks.

2 Related Work

2.1 Geospatial Entity Resolution

Early GER approaches predominantly relied on comparing coordinate distances and textual similarities using hand-crafted rules or thresholds (Isaj et al., 2022; Deng et al., 2019; Shivaprabhu et al., 2017; Morana et al., 2014). These methods, however, often struggled with capturing nuanced attribute semantics and effectively handling subtle variations, thus limiting their capacity for deeper understanding.

The advent of PLMs (Devlin et al., 2019; Liu et al., 2019) signified a notable progression, recasting GER as a sequence-pair classification task. This approach yielded improved semantic understanding via context-aware embeddings from fine-tuning (Balsebre et al., 2022, 2023). Despite these strides, the considerable demand for labeled data in fine-tuning PLMs and their inherent limitations in deeply integrating spatial context continue to spur methodological innovation.

2.2 Entity Resolution with LLMs

LLMs, via In-Context Learning (ICL), enable zero-shot or few-shot entity matching, significantly reducing reliance on extensive training data. For

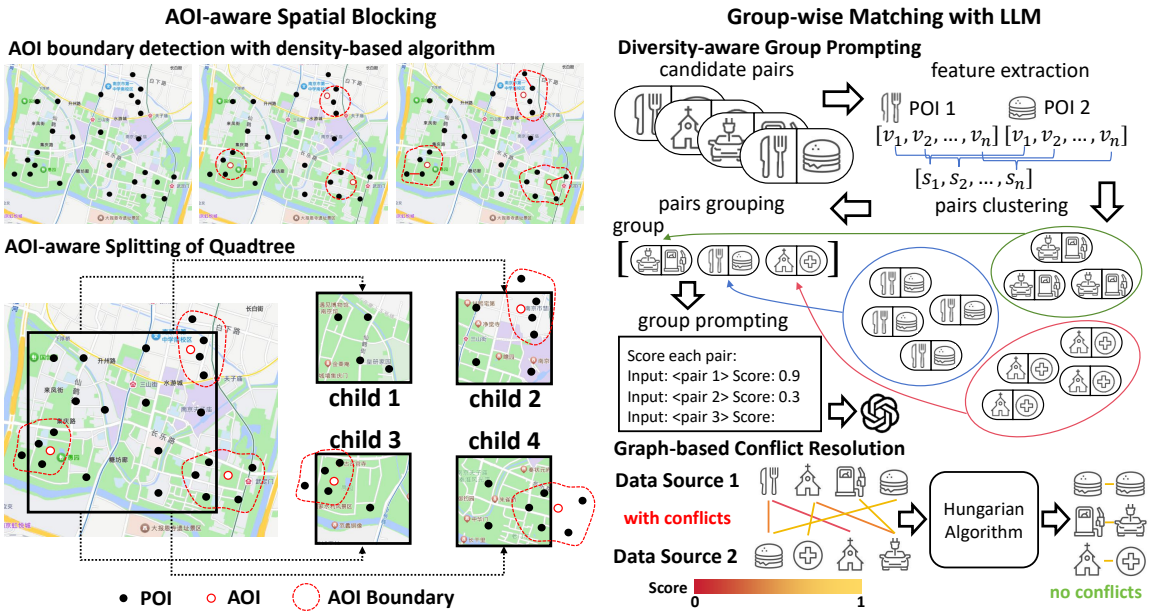

Figure 2: The overview framework of our proposed , which consists of two main components namely: (1) AOI-aware Spatial Blocking and (2) Group-wise Matching with LLM.

general ER, LLMs often surpass PLMs, as demonstrated by work like Peeters et al. (2025), particularly in low-data settings. Research also addresses ICL operational aspects such as cost (Fan et al., 2024) and advanced matching paradigms (Wang et al., 2025), alongside broader explorations of LLMs in ER (Narayan et al., 2022; Steiner et al., 2024; Li et al., 2024).

However, directly applying these general LLM-based ER methods to GER is not straightforward. Current approaches often overlook potential conflicts from independent LLM evaluations and typically lack specific designs tailored to GER's unique spatial intricacies. While Pyo and Chiang (2024) applied LLMs to geospatial data generation for PLM fine-tuning, enhancing LLMs directly for GER tasks remains largely unaddressed.

3 Preliminaries

We begin by defining the basic unit of GER—the spatial entity. A **spatial entity** refers to an object in the urban space, which in most cases can be considered a Point of Interest (POI). Each entity is associated with a set of attributes, including both spatial attributes (e.g., latitude and longitude) and textual or categorical attributes (e.g., name, address, category). Formally, we represent a spatial entity e as a key–value dictionary $\{(k_i : v_i)\}_{1 \le i \le n}$ where $(k_i : v_i)$ represents a pair of attribute and value.

We make use of the concept of **Area of Interest (AOI)** to incorporate spatial semantics in our method. An AOI refers to a localized region in urban space that groups together spatial entities with shared functional or semantic context. Examples include shopping malls or university campuses.

We now formalize the **Geospatial Entity Resolution (GER)** task. Given two sets of spatial entities $S_1 = \{e_i \mid 1 \le i \le |S_1|\}$ and $S_2 = \{e_j \mid 1 \le j \le |S_2|\}$ from different data sources, the goal of GER is to identify entity pairs (e_i, e_j) such that $e_i \in S_1$, $e_j \in S_2$, and the two refer to the same real-world object.

4 Overview

As shown in Figure 2, the GER-LLM framework includes two main components namely: (1) AOI-aware Spatial Blocking and (2) Group-wise Matching with LLM.

We first design an **AOI-aware spatial blocking strategy** to integrate spatial semantic. Unlike traditional GER methods that rely solely on distance thresholds, our approach leverages both spatial proximity and functional relationships—such as shared AOIs—to group entities likely to be related. Specifically, we extract AOIs from spatial entities using a classification model and apply a density-based algorithm to detect flexible AOI boundaries. We then propose an adaptive quadtree algorithm that assigns boundary entities to multiple blocks based on shared AOIs, ensuring that matches are preserved even across partition edges. This strategy not only narrows down the candidate scope for each entity but also integrates both spatial adjacency information within the urban space.

To further address the computation overhead, we introduce a **group-wise matching pipeline** that reduces the need for exhaustive pairwise comparisons. Rather than evaluating each candidate pair individually, we organize pairs into diverse groups and prompt the LLM to jointly assess alignment scores. To maximize the effectiveness of group prompting, we first cluster the candidate pairs and then construct groups by sampling pairs from different clusters to ensure diversity—an approach shown to enhance entity resolution performance (Fan et al., 2024). Finally, since independent group assessments may produce conflicting alignments, we develop a graph-based conflict resolution mechanism to enforce global consistency across the matching results.

5 AOI-aware Spatial Blocking

In this section, we introduce our spatially informed blocking strategy for GER. We hope our blocking method maintains spatial proximity and functional relationships throughout the blocking process. To achieve this, we propose a quadtree-based method enhanced with AOI.

5.1 Quadtree with AOI-Aware Splitting

The quadtree is a hierarchical spatial index that recursively partitions a 2D space into four quadrants, efficiently organizing spatial data by assigning objects to leaf nodes based on their location (Fu et al., 2021; Tsuzuki et al., 2019; Li et al., 2018). In our GER setting, spatial entities from both data sources are jointly partitioned by the same quadtree structure. Each leaf node corresponds to a spatial block containing entities from both sources that fall within the same spatial region, while internal nodes represent coarser subdivisions of the urban space.

Split Strategy: The core difference between our method and conventional spatial blocking lies in its customized splitting strategy. Traditional methods use fixed geometric boundaries to split space, which can arbitrarily separate spatially or semantically related entities across partition lines. This often leads to fragmented candidate sets and lower recall. To address this, we propose a soft splitting strategy that incorporates AOI information. During splitting, it not only assigns spatial entities based on their spatial coordinates but also propagates semantically related spatial entities—those sharing the same AOIs—into the same or adjacent blocks. This ensures that potential matches are not excluded simply due to rigid spatial boundaries.

Algorithm 1 AOI-Aware Split Strategy

Require: node q, AOI-POI relations $\mathcal{I}_{A_2P}, \mathcal{I}_{P_2A}$
Ensure: four child nodes $child[1..4]$

initialize $child[1..4] \leftarrow$ empty child nodes
for $e \in q.internal\_set$ **do**
    assign e to corresponding $child[i]$ by spatial position
end for
for $e \in q.external\_set$ **do**
    $related\_pois \leftarrow \mathcal{I}_{A_2P}[\mathcal{I}_{P_2A}[e]]$
    for each i such that $child[i]$ contains a $poi \in related\_pois$ **do**
        add e to $child[i].external\_set$
    end for
end for
for $i \in 1..4$ **do**
    for $e \in child[i].internal\_set$ **do**
        $related\_pois \leftarrow \mathcal{I}_{A_2P}[\mathcal{I}_{P_2A}[e]]$
        add missing $related\_pois$ to $child[i].external\_set$
    end for
end for
return $child[1..4]$

Specifically, Algorithm 1 outlines the AOI-aware split process, which recursively divides a quadtree node into four child nodes, assigning spatial entities based on both spatial position and shared AOIs to preserve semantic locality. Internal spatial entities are allocated by coordinates, while external spatial entities are propagated across children if they share AOIs with any internal points.

In most cases, the complexity of our method is no more than $O((|S_1|+|S_2|)\log D)$ where D is the diagonal length of the square region covering all POIs. A more detailed complexity analysis, which depends on the spatial density and AOI coverage, is provided in the Appendix A.1.

5.2 AOI Inference via Classification and Boundary Detection

Our blocking strategy assumes the availability of AOI-POI relationships. However, such labels are often missing in real-world geospatial data. Existing approaches commonly approximate AOIs using fixed-radius circles around these points (Balsebre et al., 2023), but such heuristics fail to reflect the actual irregular shapes of AOIs and can introduce significant overlap or redundancy.

AOI inference via classification model: To ad-

dress the absence of AOI labels, we train a binary classification model that predicts whether a given entity is an AOI based on its attributes—particularly textual features such as name or category. Since AOIs often have distinguishable naming patterns (e.g., "Campus", "Industrial Park"), we use a lightweight text encoder followed by a classifier to make predictions. For training, we construct a labeled dataset by selecting entities of known types (e.g., campuses, malls) as positive examples and randomly sampling other entities as negatives.

Boundary detection via density-based detection algorithm: Once AOIs are identified, we infer their spatial boundaries using a density-based detection algorithm. By analyzing the local distribution of surrounding spatial entities, the algorithm adaptively identifies the high-density region associated with each AOI. This allows us to infer more realistic boundaries and more accurately determine which entities fall within a given AOI.

Specifically, we classify each entity as a core, reachable, or outlier. An entity is marked as a core if it has at least M neighbors within a given radius ζ; otherwise, it is either reachable—if it's close to at least one core—or an outlier. We first examine each AOI and retain only those that qualify as core entities; others are treated as individual POIs. For the remaining POIs, we identify nearby core AOIs within the same radius ζ and establish bidirectional links between them. This allows POIs to associate with the most relevant AOIs based on local spatial density. A more detailed description of the algorithm, including pseudocode and algorithm details, is provided in Appendix A.2.

6 Group-wise Matching with LLM

After spatial blocking, each block yields a large number of candidate entity pairs via Cartesian product. To address this, we introduce a group-wise matching strategy that organizes candidate pairs into semantically diverse batches and prompts the LLM to assess them jointly. Moreover, to enhance reasoning quality, each group is constructed to maximize internal diversity, enabling the LLM to leverage richer comparative context when distinguishing true matches from non-matches (Section 6.1). Additionally, since individual group assessments may produce conflicting decisions across the dataset, we resolve inconsistencies through a graph-based matching algorithm (Section 6.2).

6.1 Diversity-aware Group Prompting

The core idea behind our group prompting strategy is to maximize the diversity within each prompt group. To achieve this, we aim to avoid grouping similar candidate pairs together. We first cluster similar pairs and then construct each prompt group by sampling from different clusters. Our group prompting pipeline consists of four steps: feature extraction, pairs clustering, pairs group, and group prompting.

Feature Extraction: We transform each candidate entity pair (a, b) into a feature vector r by computing similarity scores across all aligned attributes. For each attribute—such as name, address, or category—we calculate a corresponding similarity score. For textual attributes, we apply standard metrics such as token-based similarity (e.g., Jaccard) and normalized edit distance (e.g., Levenshtein). For spatial attributes, such as latitude and longitude, we compute normalized geographical distance to capture spatial closeness. These computed similarity scores are then concatenated in a predefined order to form the final feature vector $r = [s_1, s_2, \ldots, s_n]$. This concise vector aims to capture essential relational knowledge from multi-attribute matching signals for subsequent model processing.

Pairs Clustering: Next, we cluster candidate entity pairs based on Euclidean distances between their n-dimensional feature vectors. We employ the HDBSCAN algorithm (Campello et al., 2013), valued for its density-based identification of arbitrarily shaped clusters and effective noise point designation. This could yields a primary set of clusters $\mathcal{K}$, alongside some identified noise points. To ensure comprehensive clustering, these identified noise points are then assigned to existing clusters using a K-Nearest Neighbors (KNN) approach (Cover and Hart, 1967). Each noise point adopts the majority cluster label of its k nearest clustered neighbors. This two-stage process ensures every candidate pair is assigned to a cluster.

Pairs Grouping: We then construct prompt groups G of size g by selecting candidate entity pairs from the previously formed clusters $\mathcal{K}$ to maximize intra-group diversity. This process has two stages: if at least g distinct clusters with available pairs remain, we ensure diversity by randomly selecting one pair from each of g different clusters. Otherwise, we employ a round-robin selection from the remaining available clusters until the group

reaches size g. This method consistently yields prompt groups with varied entity pairs, thereby optimizing the contextual information for the LLM.

Group Prompting: Once the diverse prompt groups G are constructed, we leverage an LLM for the matching task. Each group is systematically formatted into a single, consolidated prompt, the specifics of which are detailed in Appendix A.6. This consolidated prompt is then submitted to the LLM. The LLM, in turn, jointly evaluates all candidate entity pairs, assigning to each pair p a prediction $y_p \in \{0, 1\}$ and an associated confidence score c_p.

6.2 Graph-based Conflict Resolution

Independent LLM evaluations of distinct prompt groups can yield globally inconsistent alignments. For instance, an entity e_1 from one data source might be matched by the LLM to both entities e_2 and e_3 from another, violating the expected one-to-one correspondence often assumed in GER, especially when dealing with well-maintained source datasets. Such ambiguities, arising from the LLM's isolated assessment of groups without a global view, directly reduce the precision of the final resolution.

To resolve these conflicts and enforce global consistency, we adopt a graph-based strategy. As illustrated in Figure 2, we construct a bipartite graph where nodes represent candidate entities, partitioned into two disjoint sets according to their original data sources. An edge is added between two entities from different partitions if the LLM predicts them as a match, with the edge weight corresponding to the LLM's confidence score c_p. We then apply the Hungarian algorithm (Kuhn, 1955) to identify the maximum weight matching in the graph, which yields a globally consistent set of aligned entity pairs—effectively reconciling potentially conflicting predictions made across different prompt groups.

7 Experiments

In this section, we present an experimental evaluation of GER-LLM, using three real-world datasets. Our evaluation aims to answer the following research questions:

- **RQ1**: How does GER-LLM compare to state-of-the-art methods in GER performance?
- **RQ2**: How efficient is GER-LLM compared to other LLM-based ER methods?
- **RQ3**: How does each component contribute to GER-LLM?

7.1 Experimental Setup

Datasets. For our experiments, we create three datasets by collecting geospatial entities from five real-world LBSs, accessed via their respective APIs. These datasets are constructed from pairings of these services and cover three distinct cities. Within each dataset, we manually annotated the matching entity pairs and labeled AOIs. Detailed statistics and sources of the data set are available in Appendix A.3.

Baselines. We compare our model with a representative set of baseline methods. These include approaches for general ER (**GraphER** (Li et al., 2020), **CollaborEM** (Ge et al., 2021)), dedicated frameworks for GER (**SkyEx** (Isaj et al., 2022), **GeoER** (Balsebre et al., 2022), **GTMiner** (Balsebre et al., 2023)), and recent LLM-based ER methods (**BATCHER** (Fan et al., 2024), **COMEM** (Wang et al., 2025)). We follow the descriptions in these works to preprocess the data into the corresponding format and run the baselines with the best parameter settings to ensure fairness. More details can be found in the Appendix A.4.

Metrics and Implementation. To evaluate overall performance, we employ three widely recognized metrics: Precision (P), Recall (R), and F1 score (F1). Following related studies (Neuhof et al., 2024), we assess blocking performance using three key metrics: Pair Completeness (PC), Pair Quality (PQ), and Reduction Ratio (RR).

Pair Completeness corresponds to recall, measuring the proportion of detectable matched pairs in C with respect to those in $S_1 \times S_2$:

$$PC(C, S_1, S_2) = \frac{|\mathcal{D}(C)|}{|\mathcal{D}(S_1 \times S_2)|}, \quad (1)$$

where $\mathcal{D}(x)$ denotes the set of matched pairs in set x.

Pair Quality corresponds to precision, measuring the proportion of pairs in C that are true matches:

$$PQ(C) = \frac{|\mathcal{D}(C)|}{|C|}. \quad (2)$$

Reduction Ratio measures the reduction in the number of candidate pairs in C with respect to the brute-force approach:

$$RR(C, S_1, S_2) = 1 - \frac{|C|}{|S_1 \times S_2|}. \quad (3)$$

Table 1: Effectiveness comparison of different models across three datasets. The best results for each metric are highlighted in bold, the second-best are underlined.

Model	NJ			HZ			PIT		
	P	R	F1	P	R	F1	P	R	F1
GraphER	0.5698	0.5698	0.5698	0.6137	0.6458	0.6293	0.5563	0.6146	0.5840
CollaborEM	**0.9604**	0.5253	0.6791	0.9263	0.6391	0.7564	0.9053	0.3660	0.5212
SkyEx	0.6431	0.5830	0.6116	0.6395	0.5846	0.6108	0.6034	0.5362	0.5678
GeoER	0.8146	0.9389	0.8723	0.8345	0.8962	0.8643	0.8740	0.8127	0.8422
GTMiner	0.9333	0.8077	0.8660	0.9076	0.8233	0.8634	0.8831	0.8281	0.8547
PairMatching	0.6454	0.9343	0.7634	0.7464	**0.9728**	0.8447	0.7528	**0.9628**	0.8450
BATCHER	0.7265	**0.9410**	0.8200	0.8268	0.9523	0.8851	0.7704	0.9484	0.8502
COMEM	0.8746	0.8425	0.8582	**0.9468**	0.8830	0.9138	0.9036	0.8624	0.8825
GER-LLM	0.8665	0.9132	**0.8892**	0.9407	0.9418	**0.9412**	**0.9064**	0.9313	**0.9187**

For these metrics, higher values consistently signify more effective performance. Detailed specifics of our implementation are provided in Appendix A.5.

7.2 RQ1&RQ2: Overall Performance

We first evaluate the GER performance of GER-LLM compared to the baselines.

7.2.1 Effectiveness

The effectiveness results of all methods across the three datasets are shown in Table 1. These results reveal GER-LLM's strong performance, with an average F1 score of 0.9164 and top F1 scores on all datasets. Although some baselines may achieve higher precision or recall individually, **GER-LLM** consistently outperforms them in terms of F1 score, highlighting its stronger balance between precision and recall, and its overall effectiveness in GER tasks. GER-LLM out performs the general ER methods, GER methods, and LLM-based ER methods.

GER-LLM vs. general ER methods. General entity resolution (ER) methods often fail to account for the spatial characteristics critical to GER. By explicitly incorporating spatial awareness into the LLM matching process, our GER-specific framework effectively addresses this gap in spatial context understanding.

GER-LLM vs. GER methods. GER-LLM provides stronger semantic understanding than earlier rule-based approaches and surpasses recent PLM-based methods by more effectively integrating LLMs with spatial context. While PLM-based models often rely on task-specific training data, GER-LLM achieves competitive performance without requiring labeled GER datasets, highlighting its adaptability and practical scalability.

Table 2: Efficiency comparison with contemporary LLM-based ER methods. Best runtimes are in bold, second-best are underlined. ("s" denotes seconds; "m" denotes minutes)

Model	NJ	HZ	PIT
PairMatching	157m 23s	161m 42s	176m 42s
BATCHER	7m 35s	8m 12s	11m 20s
COMEM	43m 40s	54m 18s	60m 9s
GER-LLM	**5m 28s**	**6m 10s**	**8m 35s**

Table 3: Comparison of LLM interaction counts with contemporary LLM-based ER methods, highlighting the best entries in bold and second-best entries with an underline.

Model	NJ	HZ	PIT
PairMatching	6008	6423	6623
BATCHER	319	324	371
COMEM	1804	1966	2185
GER-LLM	**229**	**245**	**284**

GER-LLM vs. LLM-based ER methods. Direct applications of LLMs to GER—such as our PairMatching baseline with independent pairwise evaluation—struggle to interpret spatial nuances and suffer from the scalability challenges of $O(N^2)$ comparisons. Even more advanced LLM-based ER methods often lack deep integration of spatial context or fail to provide robust conflict resolution. In contrast, GER-LLM addresses these limitations through a geospatially tailored architecture. This architecture combines specialized blocking, which itself demonstrates strong benchmark performance (Table 10), with diversity-aware group-wise matching and global consistency enforcement.

7.2.2 Efficiency

We validate the runtime efficiency of our framework against contemporary LLM-based ER methods. For a fair and reliable comparison, all evaluated methods use parallelized LLM interactions. The reported runtimes and LLM interaction counts, obtained using the DeepSeek-V3 model and averaged over five independent executions, are shown in Table 2 and Table 3.

The results demonstrate GER-LLM's superior runtime. It substantially outperforms PairMatching, which is constrained by its slow pairwise evaluations, and COMEM, characterized by a costly two-stage filtering process. Furthermore, GER-LLM achieves an approximate 26% runtime reduction over the more optimized BATCHER. This efficiency stems from its GER-specific design, combining effective blocking to reduce candidate entities with group-wise matching to lessen LLM calls.

7.2.3 Scalability

To evaluate scalability at larger scales, we expand the number of POIs and measure runtime versus data growth. Table 4 shows near-linear growth from $1.0\times$ to $2.0\times$, indicating our efficiency gains persist at large scale.

Table 4: Runtime vs. dataset expansion (minutes:seconds).

Expand / Time	NJ	HZ	PIT
1.0×	5m28s	6m10s	8m35s
1.25×	6m59s	7m53s	10m58s
1.5×	8m31s	9m37s	13m22s
1.75×	10m6s	11m24s	15m52s
2.0×	11m43s	13m13s	18m24s

7.3 RQ3: Ablation Studies

To evaluate the distinct contributions of GER-LLM's core components, we conduct a series of ablation studies.

7.3.1 Ablation on Blocking Component

We perform an ablation study on our spatially informed blocking method to examine the impact of its constituent elements. Variants include: **Quadtree**, which uses a conventional quadtree, omitting our adaptive assignment of boundary entities via shared AOIs. **AOI-Only**, which removes quadtree partitioning, generating candidate pairs directly from entities within the same AOI. **Circular AOIs**, which replaces our density-based AOI boundary detection with a simpler circular buffer method. Figure 3 shows these results. Removing any designed component degrades the blocking performance of our method, confirming each element's contribution.

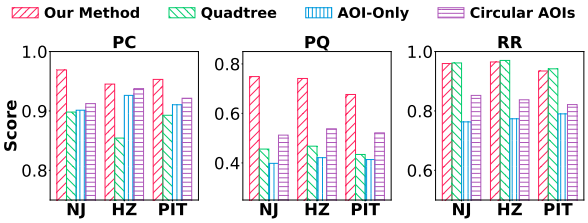

Figure 3: Ablation results for the blocking component.

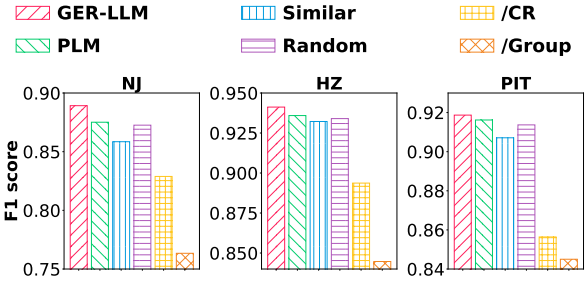

Figure 4: Ablation results for the matching component.

7.3.2 Ablation on Matching Component

To assess the effectiveness of GER-LLM's individual group-wise matching components, ablation studies on three datasets evaluate the following variants: **PLM** employs PLMs for entity pair feature extraction. For grouping, **Similar** and **Random** substitute diverse grouping with similarity-based and random strategies, respectively. **/CR** omits the Conflict Resolution module, and **/Group** eliminates group-wise matching strategies, reverting to pairwise LLM interaction. Results (Figure 4) demonstrate each module's contribution, with the Conflict Resolution module notably yielding the largest improvements.

In addition, we compare HDBSCAN with k-means, DBSCAN, and agglomerative clustering for pair grouping. As shown in Table 5, HDBSCAN yields the best average F1.

Table 5: Clustering methods for grouping (F1).

Method	NJ	HZ	PIT
k-means	0.8760	0.9331	0.9091
DBSCAN	0.8682	0.9306	0.9156
Agglomerative	0.8744	0.9318	0.9117
HDBSCAN (ours)	**0.8892**	**0.9412**	**0.9187**

Table 6: F1 scores of GER-LLM with different underlying LLMs across three datasets. Best F1 scores are highlighted in bold, the second-best are underlined. ("gemini-2.5-flash" in the table refers to gemini-2.5-flash-preview-04-17.)

Model	NJ	HZ	PIT
DeepSeek-V3	0.8892	<u>0.9412</u>	<u>0.9187</u>
DeepSeek-R1	0.8922	0.9340	0.9004
gemini-2.0-flash	0.7534	0.9312	0.9149
gemini-2.5-flash	<u>0.8930</u>	0.9355	0.9163
gpt-4.1-mini	0.8341	0.9280	0.9115
o4-mini	**0.9155**	**0.9442**	**0.9437**
Qwen3-14B	0.7552	0.8383	0.7979
DeepSeek-R1	0.7583	0.9079	0.8654
Qwen3-32B	0.7926	0.8954	0.8676
QwQ-32B	0.8231	0.8889	0.8947

7.3.3 Performance with Different LLMs

Besides these component-specific ablation studies, we evaluate GER-LLM's adaptability across diverse LLMs (Table 6), encompassing both reasoning-focused and efficiency-optimized types. While o4-mini consistently yields the highest F1 scores, our framework demonstrates robust and effective performance across all tested LLMs. Notably, models such as DeepSeek-V3 and gemini-2.5-flash also deliver strong results, underscoring GER-LLM's broad applicability and consistent high performance. In addition, we include several budget-friendly open-source 14B–32B models; the best 32B model (QwQ-32B) remains within 7 F1 points of the 671B reference on HZ and PIT, and even 14B models deliver competitive results, showing GER-LLM is effective without relying on proprietary premium LLMs.

7.4 Case Study

To intuitively study the superiority of GER-LLM, we illustrate two cases from the test set.

Case of AOI-Aware Splitting. Two records for the same café—p_1: "The Bean Scene, Unit K5, CityPlaza Mall" and p_2: "Bean Scene Coffee, Kiosk 5A, CityPlaza"-lie inside the shopping-mall AOI "CityPlaza Mall" but have coordinates that differ by a few metres. Standard quadtree blocking assigns them to adjacent tiles, so the pair is never compared, lowering recall. In contrast, our AOI-aware split strategy first recognises that both points belongs to the same AOI, then keeps them in the same block after quadtree partitioning.

Graph-Based Conflict Resolution. Table 11 (Appendix A.8) lists representative pairs from the PIT dataset that the LLM, when evaluated in group-wise mode, incorrectly labelled as mutual matches, creating one-to-many conflicts. Our graph-based conflict resolution strategy resolves the conflicts and measurably improves overall precision.

7.5 Cross-city Generalization of the AOI Classifier

We examine whether one AOI classifier trained in a city can be reused in another city of the *same language* without retraining. Table 7 reports cross-city classification results, and Table 8 shows the end-to-end GER F1 impact. A model trained on NJ and applied to HZ only drops 8% in AOI classification F1, and the final GER F1 decreases by merely $\approx 2.6\%$; the reverse transfer (HZ$\rightarrow$NJ) exhibits similarly strong robustness. These results suggest a single pretrained AOI model can be reused across cities, reducing annotation and deployment costs.

Table 7: Cross-city AOI classification results.

Train	Test	Precision	Recall	F1
NJ	HZ	0.9103	0.8621	0.8855
HZ	NJ	0.8755	0.8904	0.8829

Table 8: Cross-city reuse and GER F1 impact.

Train	Test	Precision	Recall	F1	Drop
NJ	NJ	0.8665	0.9132	0.8892	Baseline
HZ	NJ	0.8581	0.8715	0.8647	-2.76%
HZ	HZ	0.9407	0.9418	0.9412	Baseline
NJ	HZ	0.9325	0.9023	0.9172	-2.55%

8 Conclusion

In this work, we revisited the GER problem through the lens of recent advances in LLMs. We showed that LLMs—despite their strong zero-shot capabilities—cannot be directly applied due to their inefficiency and lack of spatial grounding. To bridge this gap, we proposed **GER-LLM**, a novel framework that integrates spatially informed blocking with group-wise LLM-based matching and global consistency resolution. Our experiments demonstrate that this approach enables accurate and scalable entity resolution across real-world geospatial datasets, offering new insights into the potential and challenges of deploying LLMs for spatial data integration.

9 Limitations

While GER-LLM demonstrates strong performance across multiple real-world datasets, there are several limitations worth noting. First, due to the time and resource constraints of manual annotation, we did not conduct city-scale human evaluation beyond the current benchmark datasets. Expanding the test set with broader geographic coverage and finer-grained labels could offer further insights into the model's generalizability. Second, although our method performs well with powerful LLMs, we did not explore its effectiveness when combined with smaller or more resource-efficient models. Evaluating the trade-off between performance and computational cost in low-resource settings is a promising direction for future work.

10 Ethical Impact

All datasets in this study are constructed from publicly available, open-licensed POI records (e.g., commercial LBS providers and open platforms). We do *not* use individual mobility trajectories, device identifiers, or any data that could directly track persons. Therefore, the typical privacy risks associated with trajectory data do not arise in our setting, and our use complies with community ethics guidelines and licenses.

Acknowledgments

This work is supported by the National Key R&D Program for the 14th-Five-Year Plan of China (2023YFC3804104 in 2023YFC3804100).

References

Pasquale Balsebre, Dezhong Yao, Gao Cong, and Zhen Hai. 2022. Geospatial entity resolution. In *Proceedings of the ACM Web Conference 2022*, WWW '22, New York, NY, USA. Association for Computing Machinery.

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

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.

Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Yue Deng, An Luo, Jiping Liu, and Yong Wang. 2019. Point of interest matching between different geospatial datasets. *ISPRS Int. J. Geo Inf.*, 8(10):435.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

Meihao Fan, Xiaoyue Han, Ju Fan, Chengliang Chai, Nan Tang, Guoliang Li, and Xiaoyong Du. 2024. Cost-effective in-context learning for entity resolution: A design space exploration. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 3696–3709. IEEE.

Cheng Fu, Haosheng Huang, and Robert Weibel. 2021. Adaptive simplification of GPS trajectories with geographic context - a quadtree-based approach. *Int. J. Geogr. Inf. Sci.*, 35(4):661–688.

Congcong Ge, Pengfei Wang, Lu Chen, Xiaoze Liu, Baihua Zheng, and Yunjun Gao. 2021. Collaborem: A self-supervised entity matching framework using multi-features collaboration. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12139–12152.

Suela Isaj, Torben Bach Pedersen, and Esteban Zimányi. 2022. Multi-source spatial entity linkage. *IEEE Trans. Knowl. Data Eng.*, 34(3):1344–1358.

Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. 2011. Density-based clustering. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(3):231–240.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Bing Li, Wei Wang, Yifang Sun, Linhan Zhang, Muhammad Asif Ali, and Yi Wang. 2020. Grapher: Token-centric entity resolution with graph convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8172–8179.

Huahang Li, Shuangyin Li, Fei Hao, Chen Jason Zhang, Yuanfeng Song, and Lei Chen. 2024. Booster: leveraging large language models for enhancing entity resolution. In *Companion Proceedings of the ACM Web Conference 2024*, pages 1043–1046.

Zhuoran Li, Guiling Wang, Jinlong Meng, and Yao Xu. 2018. The parallel and precision adaptive method of marine lane extraction based on quadtree. In *Collaborative Computing: Networking, Applications and Worksharing - 14th EAI International Conference, CollaborateCom 2018, Shanghai, China, December 1-3, 2018, Proceedings*, volume 268 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 170–188. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Anthony Morana, Thomas Morel, Bilal Berjawi, and Fabien Duchateau. 2014. Geobench: a geospatial integration tool for building a spatial entity matching benchmark. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 533–536.

Avanika Narayan, Ines Chami, Laurel J. Orr, and Christopher Ré. 2022. Can foundation models wrangle your data? *Proc. VLDB Endow.*, 16(4):738–746.

Franziska Neuhof, Marco Fisichella, George Papadakis, Konstantinos Nikoletos, Nikolaus Augsten, Wolfgang Nejdl, and Manolis Koubarakis. 2024. Open benchmark for filtering techniques in entity resolution. *VLDB J.*, 33(5):1671–1696.

George Papadakis, George Papastefanatos, Themis Palpanas, and Manolis Koubarakis. 2016. Scaling entity resolution to large, heterogeneous data with enhanced meta-blocking. In *EDBT*, pages 221–232.

Ralph Peeters, Aaron Steiner, and Christian Bizer. 2025. Entity matching using large language models. In *Proceedings 28th International Conference on Extending Database Technology, EDBT 2025, Barcelona, Spain, March 25-28, 2025*, pages 529–541.

Jiyoon Pyo and Yao-Yi Chiang. 2024. Leveraging large language models for generating labeled mineral site record linkage data. In *Proceedings of the 7th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, pages 86–98.

C Carl Robusto. 1957. The cosine-haversine formula. *The American Mathematical Monthly*, 64(1):38–40.

Vivek Sehgal, Lise Getoor, and Peter D Viechnicki. 2006. Entity resolution in geospatial data integration. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 83–90.

Setu Shah, Vamsi Meduri, and Mohamed Sarwat. 2021. GEM: an efficient entity matching framework for geospatial data. In *SIGSPATIAL '21: 29th International Conference on Advances in Geographic Information Systems, Virtual Event / Beijing, China, November 2-5, 2021*, pages 346–349. ACM.

Vivek R. Shivaprabhu, Booma Sowkarthiga Balasubramani, and Isabel F. Cruz. 2017. Ontology-based instance matching for geospatial urban data integration. In *Proceedings of the 3rd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics, Redondo Beach, CA, USA, November 7, 2017*, pages 8:1–8:8. ACM.

Aaron Steiner, Ralph Peeters, and Christian Bizer. 2024. Fine-tuning large language models for entity matching. *arXiv preprint arXiv:2409.08185*.

Satori Tsuzuki, Daichi Yanagisawa, and Katsuhiro Nishinari. 2019. Auto-generation of centerline graphs from geometrically complex roadmaps of real-world traffic systems using hierarchical quadtrees for cellular automata simulations. *Inf. Sci.*, 504:161–177.

Tianshu Wang, Xiaoyang Chen, Hongyu Lin, Xuanang Chen, Xianpei Han, Le Sun, Hao Wang, and Zhenyu Zeng. 2025. Match, compare, or select? an investigation of large language models for entity matching. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 96–109.

Anran Yang, Cheng Fu, Qingren Jia, Weihua Dong, Mengyu Ma, Hao Chen, Fei Yang, and Hui Wu. 2025. Evaluating and enhancing spatial cognition abilities of large language models. *International Journal of Geographical Information Science*, pages 1–36.

A Appendix

A.1 Time Complexity Analysis of Our Blocking Method

Let S_1 and S_2 represent two datasets to be linked, with c being the minimum distance between any two points in these datasets. D_1 and D_2 represent the dimensions of the initial region containing all points, and δ is the average coverage diameter of the AOI, approximated as a circle. It is clear that the physical diagonal length of the quadtree node $L_{physical}$ at the i-th level (where $i \geq 0$) is given by

$$L_{physical,i} = \sqrt{\frac{D_1^2 + D_2^2}{4^i}}. \tag{4}$$

In our method, to accommodate more potential matching entity pairs, the quadtree logically expands during its splitting process using the AOIs. As a result, the actual maximum size of each node is increased by a distance of δ in both the length and width directions. Based on this, the logical diagonal length of each quadtree node $L_{logical}$ at level i (where $i \geq 1$) is calculated using the following formula:

$$L_{logical,i} = \sqrt{\frac{D_1^2 + D_2^2}{4^i} + \frac{(D_1 + D_2)\delta}{2^{i-1}} + 2\delta^2}. \tag{5}$$

Since c represents the minimum distance between any two points from the two datasets, the distance $dist(p_1, p_2)$ between any two points p_1 and p_2 within a quadtree node at any level i ($i \geq 1$) will always satisfy the following inequality:

$$c \leq dist(p_1, p_2) \leq L_{logical,i}. \tag{6}$$

Thus, we can conclude that c satisfies the inequality:

$$c \leq L_{logical,i}. \tag{7}$$

By squaring both sides of Ineq. (7), shifting terms, and applying other operations, and setting $x = 2^i$, the inequality is obtained:

$$(c^2 - 2\delta^2) \cdot x^2 - 2(D_1 + D_2)\delta \cdot x - (D_1^2 + D_2^2) \leq 0. \tag{8}$$

We can analyze the solution to Ineq. (8) by considering different cases:

(1) When $c^2 - 2\delta^2 \leq 0$, the inequality always holds. This case indicates that the current datasets has a relatively dense distribution of POIs, with most of them falling within the semantic coverage of the AOIs. In this case, manually set thresholds — such as the maximum diagonal length of the node or the POI density within the node — determine when the construction process should stop. For instance, with the maximum diagonal length threshold, the diagonal length of the nodes decrease almost exponentially by a factor of 2 as the quadtree splits starting from level 0. A constant number of splits is sufficient for the node's diagonal to reach the set threshold, halting further splitting of the quadtree. As a result, in such case, the overall time complexity for constructing the quadtree is

$$O((|S_1| + |S_2|) \log_2 \sqrt{D_1^2 + D_2^2}). \tag{9}$$

(2) When $c^2 - 2\delta^2 > 0$, the specific solution to the inequality needs to be found. This case suggests that the distribution of POIs in the current datasets is quite sparse, and in most cases, it falls outside the semantic coverage of the AOIs. In such case, the datasets itself determines the termination of the construction process. Solving the inequality above gives:

$$i \leq \log_2(w) + 1, \tag{10}$$

where $i \geq 0$ and $w = \frac{\Delta}{c^2 - 2\delta^2}$, with

$$\Delta = \delta(D_1 + D_2) + \sqrt{\begin{matrix}(D_1^2 + D_2^2)(c^2 - \delta^2) \\ + 2D_1 D_2 \delta^2\end{matrix}}. \tag{11}$$

The total time complexity for constructing the structure of our method, considering all POIs in the two datasets, is:

$$O((|S_1| + |S_2|)(\log_2(w) + 1)). \tag{12}$$

A.2 Density-based AOI Boundary Refinement

We adopt the idea of density-based (Ester et al., 1996), (Kriegel et al., 2011) and design a density-based boundary detection algorithm. Specifically, for each entity in the geospatial database, we first define three types of entity (*i.e.,* core entity, reachable entity, and outlier entity) as follows.

Core Entity: An entity e is considered a core entity when it has more than M neighbors within ζ. Specifically, the indicated function $f_c(e)$ is calculated as follows.

$$f_c(e) = |N_\zeta(e)| \geq M \tag{13}$$

$$N_\zeta(e) = \{e' \mid \text{distance}(e, e') \leq \zeta\} \tag{14}$$

Here, $N_\zeta(e)$ represents the entities within the ζ-neighborhood of e, and M denotes the number of neighbors required for e to become a core entity.

Reachable Entity A reachable entity is a non-core entity that can reach core entities within its ζ-neighborhood. The formal definition of its indicator function $f_r(e)$ is as follows.

$$f_r(e) = |N_{c,\zeta}(e)| \geq 1 \tag{15}$$

$$N_{c,\zeta}(e) = \{e' \mid \text{distance}(e, e') \leq \zeta \text{ and } f_c(e')\} \tag{16}$$

Here, $N_{c,\zeta}(e)$ denotes the core entities within the ζ-neighborhood of e.

Outlier Entity An outlier entity is an entity that is neither a core entity nor a reachable entity.

Then, we eliminate isolated AOIs and detect the boundaries of the remaining ones according to Algorithm 2. First, for each AOI in S_A, we check if it is a core entity. If it is, we add it to S_{repair_A}; otherwise, we refine it as a non-core POI and add it to S_P (Lines 4-12). Next, for each POI in S_P, we find its ζ-neighborhood and compute the intersection with S_{repair_A}. If the POI is reachable from any core AOI, we establish the relationship index between the POI and its reachable AOIs (Lines 13-21). Finally, the algorithm outputs the refined set of POIs and AOIs, S_P and S_{repair_A}, along with the corresponding relationship indexes between AOIs and POIs, specifically $\mathcal{I}_{A_2P}$ and $\mathcal{I}_{P_2A}$ (Line 22).

A.3 Dataset Statistics and Sources

Table 9: Statistics of the datasets used in our experiments. The column *#Positive* shows the number of positive samples.

Source	City	$\lvert S_1\rvert$	$\lvert S_2\rvert$	$\lvert S_{AOI}\rvert$	#Positive
DP-MT	NJ	12176	828	180	411
GD-DP	HZ	1982	2959	107	808
OSM-FSQ	PIT	2383	2474	181	1237

This section provides details on the datasets used in our experiments, including their sources and key statistics. The data originates from five Location-Based Services (LBSs): Dianping[3] (DP), Meituan[4] (MT), Gaode[5] (GD), OpenStreetMap[6] (OSM), and Foursquare[7] (FSQ). These services were used to construct datasets for three cities: Nanjing (NJ), Hangzhou (HZ), and Pittsburgh (PIT).

Table 9 summarizes these datasets. In this table, the "Source" column indicates the specific LBS pairing (e.g., DP-MT for the dataset from Dianping and Meituan), while the "City" column uses the respective city abbreviations defined above. The table further presents the number of entities from each LBS in the pair ($|S_1|$ and $|S_2|$), the count of identified AOIs ($|S_{AOI}|$), and the number of manually verified positive matches (*#Positive*).

[3]https://www.dianping.com/
[4]https://www.meituan.com/
[5]https://www.amap.com/
[6]https://www.openstreetmap.org/
[7]https://foursquare.com/

Algorithm 2 Density-based Boundary Detection

Require: set of spatial entities S, set of AOIs extracted from spatial entities S_A; density parameters ζ and M
Ensure: refined set of POIs and AOIs S_P, S_{repair_A}; relationship indexes between AOIs and POIs $\mathcal{I}_{A_2P}, \mathcal{I}_{P_2A}$

1: $S_P \leftarrow S - S_A$
2: $\mathcal{I}_{A_2P}, \mathcal{I}_{P_2A} \leftarrow \{\}$
3: $S_{repair_A} \leftarrow \emptyset$
4: **for** $aoi \in S_A$ **do**
5: $N_\zeta(aoi) \leftarrow$ find the ζ-neighborhood of aoi in S
6: // Check if the aoi is a core entity.
7: **if** $|N_\zeta(aoi)| \geq M$ **then**
8: add aoi to S_{repair_A}
9: **else**
10: add aoi to S_P
11: **end if**
12: **end for**
13: **for** $poi \in S_P$ **do**
14: $N_\zeta(poi) \leftarrow$ find the ζ-neighborhood of poi in S
15: // Find the aois reachable from the current poi.
16: $share_A \leftarrow N_\zeta(poi) \cap S_{repair_A}$
17: **for** $aoi \in share_A$ **do**
18: add $aoi \rightarrow poi$ to $\mathcal{I}_{A_2P}$
19: add $poi \rightarrow aoi$ to $\mathcal{I}_{P_2A}$
20: **end for**
21: **end for**
22: **return** $S_P, S_{repair_A}, \mathcal{I}_{A_2P}, \mathcal{I}_{P_2A}$

A.4 Baselines Descriptions

We compare our model with representative baseline models, including approaches for general ER, dedicated frameworks for GER and recent LLM-based ER methods.

I. Approaches for general ER:

- **GraphER** (Li et al., 2020) performs token-centric ER by using an Entity Record Graph Convolutional Network (ER-GCN) to generate in-

formed token embeddings and aggregating token-level features for matching.

- **CollaborEM** (Ge et al., 2021) enables self-supervised ER by first automatically generating training labels and then collaboratively learning matching signals from combined graph and sentence tuple features.

II. Dedicated frameworks for GER:

- **SkyEx** (Isaj et al., 2022) achieves GER by ranking candidate pairs using Pareto optimality on multi-attribute similarities and classifying them via an optimal skyline-level cut-off.
- **GeoER** (Balsebre et al., 2022) performs GER by combining Transformer-based language models for textual analysis with specialized embeddings for spatial distance and neighborhood context.
- **GTMiner** (Balsebre et al., 2023) mines geospatial relationships to construct knowledge graphs by jointly modeling textual and geospatial data through a pre-trained language model, a geospatial encoder, and a Geo-Textual interaction mechanism.

III. Recent LLM-based ER methods:

- **BATCHER** (Fan et al., 2024) enables cost-effective LLM-based ER via a batch prompting framework that explores question batching and demonstration selection, notably introducing a covering-based strategy for demonstrations.
- **COMEM** (Wang et al., 2025) executes entity matching by first filtering candidates with a medium-sized LLM using local strategies (matching/comparing), then employs a powerful LLM with a global selection strategy for precise identification.

To specifically assess the performance of our proposed spatially informed blocking method, we benchmark it against the following established blocking techniques within the GER domain:

- **GeoPrune** (Shah et al., 2021) is an efficient and lightweight blocking technique that uses the geohash encoding mechanism.
- **GeoER** (Balsebre et al., 2022) utilises a rule-based blocking method, which filters candidate entity pairs by combining their name similarity and geographical distance.
- **QuadFlex** (Isaj et al., 2022) is an unsupervised quadtree-based method for spatial entity blocking.
- **GTMiner** (Balsebre et al., 2023) extracts AOIs from spatial entities to select a small number of candidate pairs.

A.5 Implementation Details

AOI Inference via Classification and Boundary Detection. For the AOI classification model, the text encoder $\mathcal{F}$ is implemented using pre-trained BERT models: bert-base-uncased[8] for English data and bert-base-chinese[9] for Chinese data. The subsequent classifier $\mathcal{C}$ is a multi-layer perceptron (MLP). We utilize the [CLS] token output from BERT as the input feature vector for the MLP. Key training parameters include a maximum sequence length of 128, a dropout rate of 0.1, a learning rate of 3e-5, and a batch size of 32.

To train the model, we first split our overall dataset into training, validation, and test sets with a 5:2:3 ratio, respectively. We then employ a weighted binary cross-entropy loss function. In this function, the positive class (AOIs) is assigned a higher weight ($w_p > w_n$) than the negative class. This approach more heavily penalizes false negatives, which is crucial for improving the recall rate for AOI identification. Real-world datasets often exhibit a significant class imbalance where non-AOI entities (POIs) greatly outnumber AOIs; training directly on such imbalanced data can adversely affect classifier performance. To mitigate this, we construct a balanced training dataset by randomly sampling (without replacement) ρ negative examples for each positive AOI example. The optimal values for ρ (selected from the set $\{1, 3, 5, 7, 10\}$), along with the weights w_p and w_n, are determined based on achieving the best performance on the validation set.

For the density-based boundary detection algorithm, we set M to 5 and determined the value of ζ by plotting the k-distance graph (Ester et al., 1996).

Quadtree with AOI-Aware Splitting. In this module, we tune hyperparameters by grid search. Specifically, the diagonal d is selected from $\{1, 20, 40, 60, 80, 100\}$, and the density m is selected from $\{0.01, 0.03, 0.05, 0.07, 0.09\}$.

Group-wise Matching with LLM. For textual attribute similarity, we utilized Levenshtein sim-

[8] https://huggingface.co/bert-base-uncased
[9] https://huggingface.co/bert-base-chinese

ilarity. Spatial proximity was quantified using the Haversine formula (Robusto, 1957) to compute distances between entity coordinates, subsequently normalized to the [0,1] range. The feature vector dimension n was 4 for the Nanjing and Hangzhou datasets (comprising similarities for name, category, address, and spatial distance) and 3 for the Pittsburgh dataset (name, address, and spatial distance). In the HDBSCAN algorithm, we set min_cluster_size to 12 and min_samples to 5. For the assignment of any remaining noise points, the KNN algorithm used $k = 5$. The prompt group size g for selecting candidate entity pairs was set to 32.

We conducted experiments across a range of contemporary LLMs, including various versions of DeepSeek[10] (V2.5, V3, R1), Gemini[11] (2.0-flash, 2.5-flash-preview-04-17), and ChatGPT[12] (4.1-mini, o4-mini). These models represent diverse offerings and capabilities. DeepSeek-V3 was chosen for reporting primary experimental results due to its observed balance of speed and effectiveness; it also formed the basis for our LLM-based baselines. For LLM interactions, a temperature of 0.7 was used. The maximum number of output tokens was set to 2048 for models designated as non-reasoning types and 16384 for those designated as reasoning types. Specific examples of the prompts used are detailed in Appendix A.6. We instructed the LLMs to return outputs in a predefined format, enabling straightforward parsing of match decisions and confidence scores.

The Hungarian algorithm, employed for conflict resolution, was implemented using the SciPy[13] library.

We implemented our model with Pytorch 1.8, and conducted experiments on a machine equipped with a NVIDIA RTX 3090 GPU, a 64 GB RAM and a 3.50 GHz CPU.

A.6 Specific Example of the Prompt

To illustrate our group-wise matching approach, a concrete example of the prompt designed for a group size of 32 candidate pairs is detailed in the display box below. Each prompt is structured to include three core components: a clear description of the GER task, the specific information for each candidate POI pair, and strict formatting requirements for the LLM's output.

Determine if each pair of Points of Interest (POIs) refers to the same real-world entity by synthesizing textual attributes (name, category, address) and spatial distance.

Question 1:
POI A: (Name: Alumni House, Category: University Building, Address: 4765 Forbes Ave)
POI B: (Name: Henderson House, Category: Campus Residence, Address: Margaret Morrison St, at Carnegie Mellon University)
The distance between the POIs A and B is 485.06m.

.....

Question 32:
POI A: (Name: BNY Mellon Center, Category: Office Building, Address: Grant Street, 500)
POI B: (Name: Mellon Garage, Category: Parking Facility, Address: 410 Sixth Ave)
The distance between the POIs A and B is 206.66m.

INSTRUCTION:
1. Process ALL 32 question pairs above sequentially. For each question, first determine if POI A and POI B refer to the same real-world entity (1 means same, 0 means different). Second, provide your confidence score for this determination (a numerical value between 0.0 and 1.0, where 1.0 is highest confidence and 0.0 is lowest).
2. Strictly format each response line as:
Question X: <0 or 1> <confidence_score>
(Example for one line: Question 1: 0 0.95 - This means for Question 1, the POIs are considered different (0) with a confidence of 0.95)
(Example for another line: Question 2: 1 0.80 - This means for Question 2, the POIs are considered the same (1) with a confidence of 0.80)

The output should look like this for 32 questions:
Question 1: <0 or 1> <confidence_score_1>
...
Question n: <0 or 1> <confidence_score_n>

STRICT SCHEMA REQUIREMENTS:
1. Sequential continuity: Lines must progress from Question 1 to Question 32 without any breaks or missing question numbers.
2. Lexical purity: Each line MUST start with "Question X: " (where X is the question number), followed by a single space, then the 0 or 1 determination, followed by a single space, and then the confidence score. The confidence score must be a decimal number inclusively between 0.0 and 1.0. To illustrate the expected format, such a score might look like 0.7, 0.85, or 0.99, reflecting the confidence from 0.0 (no confidence) to 1.0 (full confidence); these are format examples only, not a restrictive list of allowed values.
3. Output EXACTLY 32 lines, one for each question.
4. Do NOT include any explanations, comments, slashes, or any additional characters beyond the specified format for each line.

A.7 More Experimental Results

Here we provide the benchmark results for our spatially informed blocking method against other

[10] https://www.deepseek.com/
[11] https://deepmind.google/technologies/gemini/
[12] https://openai.com/chatgpt/
[13] https://scipy.org/

Table 10: Blocking performance of our method versus baselines on three datasets using PC, PQ, and RR metrics. The best results for each metric are highlighted in bold, the second-best are underlined.

Model	Nanjing			Hangzhou			Pittsburgh		
	PC	PQ	RR	PC	PQ	RR	PC	PQ	RR
GeoPrune	0.8042	<u>0.6154</u>	<u>0.9344</u>	0.7985	<u>0.6494</u>	<u>0.9569</u>	0.8473	<u>0.6282</u>	**0.9735**
GeoER	**0.9818**	0.3731	0.5853	<u>0.9362</u>	0.2741	0.5937	<u>0.9401</u>	0.3318	0.5723
QuadFlex	0.8981	0.4059	0.8246	0.8543	0.4176	0.7801	0.8927	0.3841	0.7838
GTMiner	0.9012	0.3982	0.7631	0.9144	0.4412	0.7736	0.9104	0.4236	0.7901
Our Method	<u>0.9690</u>	**0.7488**	**0.9596**	**0.9452**	**0.7412**	**0.9649**	**0.9531**	**0.6765**	<u>0.9345</u>

methods (Table 10)

A.8 Examples of LLM Misjudgments Corrected by Graph-based Conflict Resolution

Table 11 presents characteristic examples of entity pairs from the PIT dataset that were initially misjudged as matches by the LLM during group-wise evaluation. These errors were subsequently corrected by our graph-based conflict resolution module, which enforces global consistency. The examples highlight common scenarios where the LLM, lacking a global view, might make incorrect inferences based on local similarities.

Table 11: Characteristic Examples of LLM Misjudgments Rectified by Graph-based Conflict Resolution

Record 1 (OpenStreetMap)	**Record 2 (Foursquare)**	**Reason for Initial LLM Misjudgment / Key Distinction**
Name: Murray Avenue Locksmith **Address:** Murray Avenue, 2004	**Name:** Squirrel Hill Locksmith **Address:** nan	Different business names despite both being locksmiths in a similar area (Squirrel Hill / Murray Ave). LLM might overweigh category and general location.
Name: Amberson Apartments Buildings #1 & #2 **Address:** Bayard Rd, 2	**Name:** Amberson Apartments and Towers: Building 3 **Address:** Bayard Rd	Both are "Amberson Apartments" on "Bayard Rd", but refer to distinct building numbers/groups. LLM initially overlooked specific building identifiers.
Name: Kennywood Park **Address:** nan	**Name:** Kennywood maint. shop **Address:** nan	"Kennywood maint. shop" is likely related to or within "Kennywood Park" but is a distinct, more specific entity. LLM confused a part/related service with the main entity.
Name: Southside Works Cinema **Address:** Cinema Drive, 425	**Name:** South Side Works **Address:** 445 S 27th St	The cinema is a specific venue *within* the larger "South Side Works" development area. LLM misidentified a component as the whole.
Name: Bouquet Gardens Building J **Address:** nan	**Name:** Bouquet Building D **Address:** nan	Shared "Bouquet" name and likely close proximity, but clearly distinct building letters (J vs. D). LLM focused on the common "Bouquet" and missed the specific designator.
Name: Bouquet Gardens Building G **Address:** nan	**Name:** Bouquet Gardens - Building J **Address:** 315 Oakland Ave	Both are "Bouquet Gardens" but with different building letters (G vs. J). LLM potentially confused due to the primary name match and one missing address.