Similar Region Search using LLMs on Spatial Feature Space

Al-Amin Sany[1], **Mohaiminul Islam**[1], **Tanzima Hashem**[1],
Md. Ashraful Islam[1], **Mohammed Eunus Ali**[2]
[1]Bangladesh University of Engineering and Technology (BUET),
Dhaka 1000, Bangladesh
[2]Faculty of Information Technology,
Monash University, Melbourne, Australia
{alamin.sany12, mi.mohaimin}@gmail.com, tanzimahashem@cse.buet.ac.bd,
mdashrafulpramanic@gmail.com, eunus.ali@monash.edu

Abstract

Understanding regional similarities is crucial for applications such as urban planning, tourism recommendations, business expansion, and disease prevention. While spatial data, including POI distributions, check-in activity, and building footprints, offer valuable insights, existing similarity methods—based on distance metrics, embeddings, or deep metric learning—fail to capture the contextual richness and adapt to heterogeneous spatial data. To overcome these limitations, we introduce a novel similar region search framework that ranks candidate regions based on their similarity to a query region using large language models. To further enhance performance, we fine-tune the model through self-supervised learning by introducing controlled noise into spatial data. This generates similar and dissimilar samples without relying on extensive labeled data. By transforming spatial data into natural language descriptions, our method seamlessly integrates heterogeneous datasets without requiring structural modifications, ensuring scalability across diverse urban contexts. Experiments on multiple real-world city datasets, including cross-city evaluation, demonstrate that our framework significantly outperforms state-of-the-art methods in both accuracy and ranking performance.

1 Introduction

Identifying similar regions is crucial for urban planning, tourism, business expansion, and disease prevention, with spatial data playing a key role in capturing regional characteristics. Point-of-interest (POI) distributions help city planners optimize infrastructure and zoning policies (Liu et al., 2018; Jin et al., 2024a), while check-in counts and location density enhance tourism recommendations by matching travelers with preferred destinations (Li et al., 2024a; Canturk et al., 2023; Cao et al., 2023). In business, building footprints, commercial density, and customer activity patterns aid in identifying viable locations for expansion (McKenzie and Romm, 2021). In public health, factors like residential density and healthcare accessibility help assess disease risk, enabling targeted interventions for vector-borne diseases such as dengue and malaria. As large-scale spatial data becomes more accessible, developing efficient region similarity search methods is increasingly essential for data-driven decision-making across multiple sectors.

A similar region search (SRS) problem ranks candidate regions based on their similarity to a query region. We introduce a novel solution to address the SRS problem using LLMs, preserving contextual and heterogeneous spatial information through descriptive representations that are adaptable to various urban context. The context extends beyond numeric features like POI densities or spatial arrangements. For example, a dense urban neighborhood with many small grocery stores, parks, and schools may function similarly to a suburban area with fewer but larger stores and parks, both serving as residential zones despite differing POI distributions. Likewise, a city center with clustered retail stores and a suburban shopping mall with fewer but diverse offerings both act as shopping hubs, illustrating how functional equivalencies cannot be captured through numerical metrics alone. Considering heterogeneous data in region similarity search is crucial, as different data types—such as POI distributions and building footprints—capture distinct aspects of a region's characteristics. Integrating these diverse features provides a more comprehensive similarity measure, enhancing the accuracy and relevance of region matching.

Existing region similarity methods fall into three broad categories: manual distance-based formulas (Sheng et al., 2010; Feng et al., 2019), embedding-based methods (Chan and Ren, 2023; Zhou et al., 2023; Li et al., 2023), and deep metric

Findings of the Association for Computational Linguistics: EACL 2026, pages 3885–3898

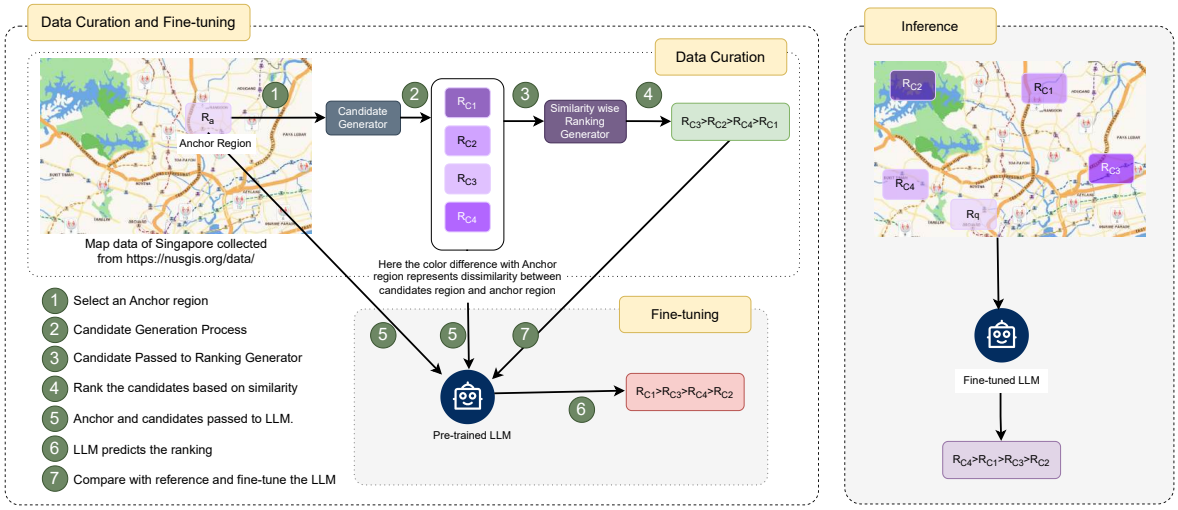

Figure 1: A LLM-based region similarity framework: Left - (1-4) data preparation pipeline, (5-8) fine-tuning of LLM model for the region similarity task; Right - inference pipeline for region similarity ranking.

learning approaches (Li et al., 2024b; Sun et al., 2024; Liu et al., 2018; Jin et al., 2024a). Distance-based methods rely on predefined metrics like Euclidean distance or cosine similarity, which can quantify spatial proximity but fail to capture functional equivalence or contextual relationships between regions. Embedding-based approaches encode spatial data into low-dimensional vectors, but they require large labeled datasets and reduce rich spatial characteristics into abstract numerical representations, making them difficult to capture context. Deep metric learning techniques, such as contrastive learning, train models to differentiate between similar and dissimilar regions, yet they struggle to capture contextual richness due to their reliance on numerical embeddings. Moreover, these methods are not inherently designed to integrate heterogeneous data sources, such as POI distributions, check-ins, and building footprints, which further limits their adaptability across various urban environments. In addition, they require a separate, computationally expensive ranking algorithm, further increasing complexity. While deep learning-based approaches have achieved promising results in various cases, they typically require model training, parameter tuning, and retraining when incorporating new constraints or updated data. Their limited generalization across tasks has led to growing interest in LLM-based methods, which offer strong adaptability and the ability to reason over complex, long-context inputs. LLMs' capacity to understand relationships and integrate multimodal information makes them intriguing candidates for addressing spatial reasoning challenges (Jiang et al., 2024; Manvi et al., 2023).

To overcome the limitations of existing approaches and establish the potential of LLMs in geo-spatial domain, we introduce a novel similar region search framework that leverages LLMs' understanding of graph-structured spatial data to infer relationships even when direct spatial connections are absent (see Figure 1). Our framework ranks candidate regions based on their similarity to a query region by encoding spatial attributes—such as POI distributions, check-in activities, and building footprints—into natural language descriptions, preserving contextual information and leveraging pretrained LLMs' commonsense reasoning for deeper spatial understanding. To further improve performance, we employ self-supervised fine-tuning by injecting controlled noise into spatial data, generating similar and dissimilar samples without labeled data. By representing spatial data in natural language, our approach seamlessly integrates heterogeneous datasets and scales across diverse urban contexts. Unlike traditional methods, our LLM-based framework directly produces region similarity rankings, eliminating the need for separate and computationally expensive ranking algorithms.

Our key contributions are summarized as follows: (i) By encoding region data as natural language descriptions through prompting, we leverage commonsense knowledge embedded in pretrained LLMs, enabling a more context-aware understanding and ranking of regional similarities. (ii) We

introduce controlled noise into spatial data to generate similar and dissimilar region pairs, then fine-tune the LLM through self-supervised learning, refining its understanding without extensive labeled data. (iii) Representing region features using natural language allows our method to adapt seamlessly across heterogeneous datasets. (iv) Experiments on mul- tiple real-world city datasets demonstrate that our LLM-based framework[1] significantly outperforms state-of-the-art methods.

2 Related Works

Similar Region Search Existing methods for region similarity learning can be broadly categorized into manual distance-based formulas, region embeddings, and deep metric learning approaches.

Manual distance-based methods rely on predefined metrics, such as Euclidean distance or cosine similarity, to compare spatial attributes of regions. Sheng et al.(Sheng et al., 2010) proposed a method that measures region similarity based on the spatial distances between objects and predefined reference points within a region. This approach calculates similarity by assessing the proximity of POIs to these reference points and applying cosine similarity to quantify how closely two regions resemble each other. Similarly, Feng et al.(Feng et al., 2019) manually selected region attributes to define similarity, relying on domain expertise to identify the most relevant features for specific tasks. While structured, these methods are highly task-specific, lack generalizability, and fail to capture contextual relationships between regions.

Region embedding approaches encode spatial characteristics (e.g., POI distributions, land use) into low-dimensional vectors (Chan and Ren, 2023; Zhou et al., 2023; Li et al., 2023), facilitating similarity computations in latent space. Graph-based embeddings have been applied to tasks like crime prediction (Wu et al., 2022; Liu et al., 2023), economic forecasting (Hui et al., 2020; Sun et al., 2023), and trajectory modeling for next POI recommendation (Cao et al., 2023). However, these approaches rely on labeled data and reduce rich spatial characteristics to abstract numerical representations, making them difficult to interpret. Additionally, they struggle to capture the full contextual richness, limiting their effectiveness in region similarity tasks.

Deep metric learning approaches aim to learn a distance function that can measure the similarity between regions based on their features. Liu et al. (Liu et al., 2018) proposed a CNN-based model for region similarity, representing regions as images and extracting spatial features from POI distributions. Using a triplet loss function, the model learns to minimize the distance between similar regions while maximizing the distance from dissimilar ones. Building on this work, Jin et al. (Jin et al., 2019) extended the model by incorporating spatial knowledge graphs to capture semantic relationships between POIs. This approach goes beyond simple spatial proximity, considering how POIs are related in terms of functionality. Zhao et al. (Jin et al., 2024a) improved the accuracy of learning of region similarity by introducing graph-based contrastive learning. Zhang et al. (Zhang et al., 2022) advanced this further with multi-view contrastive learning, where an intra-view module refines region embeddings and an inter-view module transfers knowledge across views.

A notable advancement is CARE by Jin et al. (Jin et al., 2024b), which addresses spatial imbalance in POI densities through spatial normalization, making region similarity learning more robust. It also leverages contrastive learning with triplet loss to generate training samples and improve performance in data-sparse scenarios. However, despite these advancements, deep metric learning approaches remain limited by their reliance on numerical representations of region features, making them less adaptable to diverse urban contexts and varying similarity criteria.

Our LLM-based approach dynamically assesses similarity using context-specific features and handles data sparseness with heterogeneous spatial inputs, representing a fundamental shift in region similarity learning.

LLMs for Spatial Analysis and Prediction Although LLMs have not been directly applied to region similarity search before our work, recent studies demonstrate their potential for spatial analysis. UrbanLLM (Jiang et al., 2024) decomposes urban queries into sub-tasks (e.g., travel time estimation, parking availability prediction), while GeoGPT (Zhang et al., 2023) and TrafficGPT (Zhang et al., 2024) extend LLM functionalities to geospatial and traffic data analytics. However, these models rely on structured inputs and predefined task hierarchies, limiting their ability to handle heterogeneous spatial data. UrbanGPT (Li et al., 2024c) and GeoLLM (Manvi et al., 2023) introduce instruction

[1]Code is available at: https://github.com/LLM4SRS.

tuning and fine-tuning techniques for specific urban queries, such as population density estimation. However, they do not address the broader challenge of region similarity ranking. Following (Li et al., 2024a), which used prompt engineering for next POI recommendation, we adapted the prompt engineering approach for searching for similar regions.

Unlike these studies, our work is the first to apply LLMs directly to region similarity search by transforming spatial features into rich natural language descriptions. Thus, our approach could maintain the contextual richness and utilize commonsense knowledge inherent in LLMs, without the restrictions of rigid data structures and predefined metrics. By framing the region similarity computation as a ranking problem, our framework effectively uses LLMs to measure and compare regions with nuanced spatial characteristics, setting new directions in large-scale spatial data analytics.

3 Problem Definition

We model an area as a 2D grid, dividing it into n non-overlapping regions $\mathcal{R} = \{r_j\}_{j=1}^{n}$, following (Liu et al., 2018; Jin et al., 2024b). Each region $r_j \in \mathcal{R}$ is characterized by its Points of Interest (POIs) $\mathcal{P}_j = \{P_{1,j}, P_{2,j}, \ldots, P_{k,j}\}$ and building footprints $\mathcal{B}_j = \{B_{1,j}, B_{2,j}, \ldots, B_{m,j}\}$, detailed below.

Let $P_{i,j} = \{POICounts_{i,j}, TotalSize_{i,j}, CheckIns_{i,j}\}$ represent the POI information for category c_i (e.g., "restaurant", "park") in region r_j, where $POICounts_{i,j}$, $TotalSize_{i,j}$, $CheckIns_{i,j}$ denote the total POI count, POI size, and number of check-ins at POIs, respectively.

Again, $B_{i,j} = \{BuildingCounts_{i,j}\}$ corresponds to a specific building type b_i (e.g., "commercial", "industrial", "residential") in region r_j, where $\{BuildingCounts_{i,j}\}$ denotes the total number of buildings.

By combining the POI data $\mathcal{P}_i$ and building footprint data $\mathcal{B}_i$, a region r_i is represented as: $r_j = \{\mathcal{P}_j, \mathcal{B}_j\}$

We aim to assess the similarity between a set of candidate regions and an anchor region[2] based on their regional characteristics. The similar region search (SRS) problem is formulated as follows.

Similar region search (SRS). Given an anchor region r_a and a set of candidate regions $\mathcal{G} = \{r'_1, r'_2, \ldots, r'_g\}$, the SRS problem computes a ranking function that orders the candidate regions based on their similarity to r_a. A higher rank indicates greater similarity to the anchor region.

Our LLM-based solution first transforms raw region attributes $\{\mathcal{P}_j, \mathcal{B}_j\}$ of r_j into a natural language description $\mathcal{D}_j$ and generate region prompts. A fine-tuned LLM then learns the similarity ranking function by leveraging spatial, contextual, and functional information from prompts, directly outputting the ranked list.

Table 1 lists the notations used in the paper.

Table 1: Notation Summary

Symbol	Description
$\mathcal{C}$	Set of POI categories, $\mathcal{C} = \{c_i\}_{i=1}^{k}$
$\mathcal{F}$	Set of building footprint types, $\mathcal{F} = \{b_i\}_{i=1}^{m}$
p	A POI: $< pos_{lat}, pos_{lon}, c_i, size >$
bf	A building footprint: $< pos_{lat}, pos_{lon}, b_i >$
$\mathcal{R}$	Area consisting of n non-overlapping regions
r_j	The j^{th} region of $\mathcal{R}$
$P_{i,j}$	Accumulated POI info of category c_i in region r_j
$\mathcal{P}_j$	Accumulated POI info of region r_j
$B_{i,j}$	Accumulated building footprint info of type b_i in region r_j
$\mathcal{B}_j$	Accumulated building info of region r_j
$\mathcal{G}$	Set of candidate regions
r'_g	The g^{th} candidate region
$\mathcal{S}(r_a, r_b)$	Similarity between regions r_a and r_b
$\mathcal{D}_j$	Natural language description of region r_j

4 Methodology

Our method includes three components: generating similar and dissimilar candidate regions of the anchor regions, generating region prompt, and supervised fine-tuning of pre-trained LLM. First, we divide an area into grids, and consider each grid as an anchor region. For each anchor region, we generate the candidate regions by introducing noise into the raw features of the anchor region in a controllable fashion (Section 4.1)). Secondly, we create the prompt by merging the anchor region description along with the candidate region descriptions and their similarity rankings based on the injected noise to the raw features (Section 4.2). Finally, we fine-tune the LLM with supervised learning using the region prompts (Section 4.3).

4.1 Similar and Dissimilar Candidate Region Generation

Since we do not have explicit ground truth data of similar and dissimilar regions with respect to anchor region, to train our model, we generate similar and dissimilar candidate regions by introducing noise and shift (noise-tailored parameter) to the raw features, eg, POI count, size, checkin counts and building footprints of the anchor regions. Our candidate region generation includes two components:

[2]We use 'anchor region' and 'query region' interchangeably

generating positive, r_+ (similar to anchor, r_a) and generating negative, r_- (dissimilar to anchor, r_a) candidate regions.

Positive (Similar) Candidate Regions: We generate positive candidate regions by adding small noise like adding POI or building footprint at random places, and removing them from a random place, and shifting anchor region features using some methods like repositioning an existing POI or building footprint, adjusting the size and checkin counts of the POI to keep these attributes consistent with the noise. Similarly, we use these methods to add small perturbations to other features such as POI size, check-in count, and building footprints. Figure 2 (bottom) shows an example of generating positive candidate region.

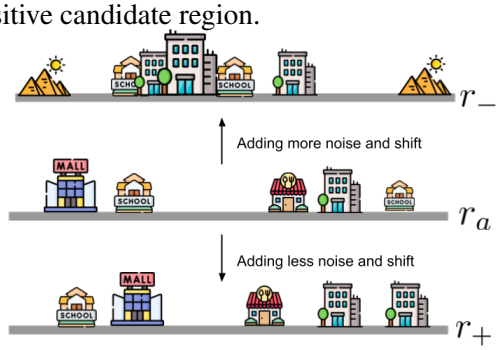

Figure 2: Positive candidate (bottom) r_+ (less noise & shit) and hard-negative candidate (top) r_- (more noise & shift) generation

Negative (Dissimilar) Candidate Regions: We generate the negative candidate regions by randomly sampling regions over the map. However, purely random sampling can easily involve very dissimilar regions with the anchor, making the task of distinguishing similar and dissimilar candidates trivial. To address this issue, we introduce a balanced approach by considering both soft-negative regions, as in Figure 3, and hard-negative regions, including slightly more noise and shift to the anchor features than to the positive candidates, as illustrated in Figure 2 (top), with a certain balance. This ensures that it challenges our model sufficiently, being less sensitive to noise while promoting its ability for nuanced similarities.

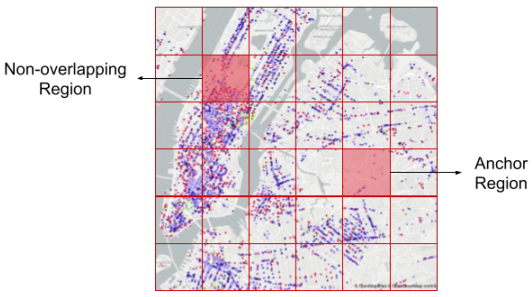

Figure 3: Soft-negative candidate generation

4.2 Region Prompt Generation

Effective prompt design is essential for fine-tuning, ensuring the LLM produces the desired output. At first, we generate the description of a region r_j as $\mathcal{D}_j$ in natural language from raw region attributes (Figure 7a in Appendix A). This description provides a structured representation of geographic and infrastructural data, suitable for LLM processing. Each region r_i is defined by two primary components: Points of Interest (POIs) and building footprints. POI data includes metrics such as *POICounts*, *TotalSize*, and *CheckIns* across categories $(c_1, c_2, \ldots, c_k) \subset \mathcal{C}$. Building footprints, categorized by types $(b_1, b_2, \ldots, b_m) \subset \mathcal{F}$, provide structural information through the *BuildingCounts* attribute. This structured approach enables LLMs to process regional characteristics, supporting tasks like region similarity assessment, functional zone analysis, and urban activity pattern recognition.

Using region descriptions, we generate prompts for ranking similar regions. The process involves selecting an anchor region and identifying candidate regions based on spatial characteristics such as POI distribution, check-in patterns, and building footprints. These descriptions are then analyzed and ranked by similarity to the anchor region. The structured prompt, encapsulating input parameters and the expected output format, is provided in Figure 7b (Appendix A).

4.3 Supervised Fine-tuning of LLM

Our framework's final step fine-tunes the Llama-3-8B model to learn a similarity ranking function for spatial region analysis, leveraging pretrained LLMs to capture contextual and functional similarities beyond rigid numerical embeddings. While the LLM has strong language understanding, fine-tuning aligns it with spatial reasoning tasks. Specifically, we refine the model to:

- **Rank candidate regions** based on their similarity to an anchor region.
- **Understand spatial relationships** by integrating POI distributions, check-in activity, and building footprints.
- **Enhance domain-specific accuracy** by adapting the model to region similarity tasks, ensuring it distinguishes between functionally equivalent and dissimilar regions.

We apply 4-bit quantization (Jacob et al., 2017) and LoRA (Hu et al., 2021), to fit the model into a small-sized GPU. The model is trained on a custom dataset generated using the methods in Sections 4.1 and 4.2. We incorporate techniques like flash-attention (Dao et al., 2022), PEFT (Xu et al., 2023) and Unsloth (Daniel Han and team, 2023) for scalability and accuracy. These adaptations allow our model to process spatial data efficiently for urban planning, geographic research, and location-based services.

5 Experiments and Results

5.1 Experimental Setup

We conducted detailed experiments on two public data sets: Tokyo and Singapore. The POI dataset, collected over 29 months, comprises data from Tokyo and Singapore and is sourced from Gowala[3]. The building footprint data are collected from OpenStreetMap (OSM)[4]. The attributes of the POI datasets are POI category (e.g., Entertainment, Community, etc.), size, location, and check-ins. The attribute of a building footprint data is its type (e.g., Commercial, Residential, etc.). We sampled 1000 non-overlapping regions from both Singapore and Tokyo dataset that contain more than 100 POIs as regions. In addition to Singapore and Tokyo, we further evaluate our model LLM4SRS on a New York City (NYC) dataset; detailed results are reported in Figures 10 and 11 (in Appendix B.2). All experiments were conducted using a cross-city evaluation protocol, where the model was fine-tuned on Singapore and evaluated on held-out regions from Singapore, Tokyo, and NYC without any city-specific adaptation. Table 2 shows the statistics of the datasets.

Table 2: Dataset Statistics

City	Cat.	POIs	Area (km^2)	Check-ins	Types	Bldgs.
TKY	8	16,508	397.72	236,288	12	968,013
SG	8	18,982	562.30	178,352	12	154,097
NYC	8	14,648	310.18	166,246	12	343,178

Test data generation. To evaluate the effectiveness of our model, we randomly sample 20% of all the regions as the test query regions, $\mathcal{R}_q = \{r_q\}_{q=1}^{Q}$, where $Q = |\mathcal{R}_q|$. We create the candidate regions for each test query region r_q using the similar approach discussed in Section 4.1 and select the most similar candidate region (r_+) as ground truth. Finally, we make the test region description prompts as described in Section 4.2. We use these test prompts of each query region r_q to produce the similarity rankings of its candidates. We varied the noise rate to generate the candidates of each r_q from 0.1 to 0.5 in experiments.

Baselines. We compare our model LLM4SRS with the following baselines:

- **SVSM (Sheng et al., 2010)**: This method uses five predefined reference points (the four corners and the center of a region) and computes the average distance of each type of spatial entity to each reference point as the reference distance feature. The region similarity between two regions is calculated by computing the cosine similarity between two reference distance feature vectors.

- **Triplet (Liu et al., 2018)**: This method encodes the attribute feature of a spatial entity into a one-hot vector with respect to its type. It treats each region as a pixel, and the relative locations of spatial entities inside a region are learned by a triplet network (CNN) to extract region features. The region similarity is computed by the Euclidean distance between the feature vectors of two regions.

- **CARE (Jin et al., 2024b)**: This method presents the Context-Aware REgion similarity learning framework, a method that computes similarities among regions depending on their spatial and application specific contexts. The existence of spatial imbalance in urban POI data is addressed with a new spatial normalization technique that estimates the importance of a region with respect to its neighborhood. CARE adopts a self-supervised contrastive learning approach where regions are compared based on normalized features that express their distinguished functional roles using triplet loss. The similarity between regions is finally determined by the Euclidean distance between learned embeddings.

- **Sentence Transformer (Li et al., 2020)**: This method utilizes the retrieval part of the Retrieval-Augmented Generation (RAG) pipeline. A sentence transformer model (all-MiniLM-L6-v2) is used to compute the embeddings in the retrieval part. Each region represented as a descriptive input, split into anchor and candidate sections. The model

[3]https://snap.stanford.edu/data/loc-Gowalla.html
[4]https://download.geofabrik.de/

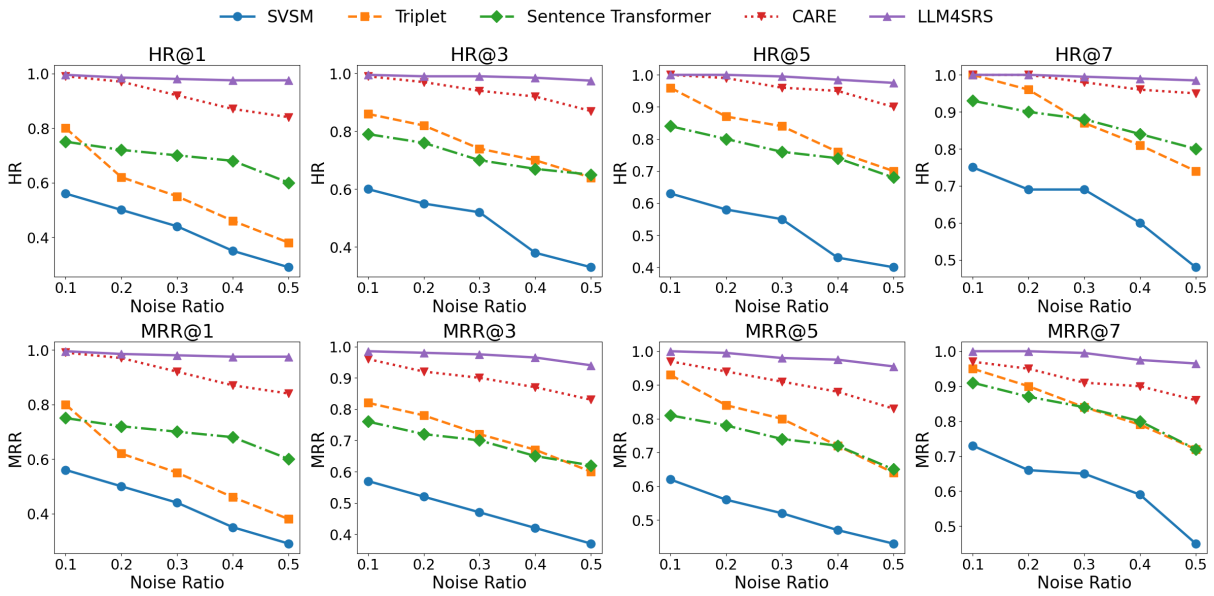

Figure 4: SG@10 candidate regions: Performance comparison with 10 candidates of LLM4SRS with SVSM, Triplet, Sentence Transformer, and CARE across noise ratios (η) and evaluation metrics (HR@k, MRR@k) for the *Singapore* datasets.

generates embeddings for the anchor and candidate regions and then computes cosine similarity between these embeddings. Candidate regions are ranked based on their similarity to the anchor region. This approach captures contextual nuances of spatial data through embeddings and leverages similarity scores to identify functionally or structurally similar regions.

Evaluation Metrics. We evaluate our region similarity measure using two standard metrics. The first, Hits@k, is defined as

$$\text{Hits@k} = \begin{cases} 1, & \text{if } r_+ \in \text{top-}k(r_q) \\ 0, & \text{otherwise} \end{cases}$$

where r_+ is the ground truth similar region and r_q is the query region. Higher Hits@k values indicate that r_+ is more frequently retrieved at top positions, signifying improved ranking performance.

The second metric is MRR (Mean Reciprocal Rank), defined as:

$$\text{MRR} = \frac{1}{Q} \sum_{q=1}^{Q} \frac{1}{rank_q}$$

where $rank_q$ is the position of the ground truth similar region r_+ in the ranked list generated by a similar region search model; by averaging the inverse ranks at which r_+ appears across all queries, a higher MRR means r_+ typically ranks closer to the top, indicating a more effective system.

Implementation Details. For fine-tuning, we employ a linear learning rate schedule with a learning rate of 2e-4, combined with a warm-up of five steps and a weight decay of 0.01. We use a batch size of 2 per device along with gradient accumulation steps of 4 to simulate a larger effective batch size without exceeding memory constraints. Our experiments are conducted on NVIDIA T4 GPUs, each equipped with 16 GB of RAM.

5.2 Experimental Results

5.2.1 Comparison of LLM4SRS with SVSM, Triplet, Sentence Transformer, and CARE.

In this section, we present the experimental results of our proposed model LLM4SRS compared to the baseline methods SVSM, Triplet, Sentence Transformer, and CARE on the task of similar region search.

We report the hit ratio (HR), and mean reciprocal rank (MRR) at $k = \{1, 3, 5, 7\}$ for both 10 and 20 candidates under varying noise ratios $\eta \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. The results are summarized in Figures 4, 5 and Figures 8, 9 (in Appendix B.1), where each graph shows the performance trends for the respective noise ratios and evaluation metrics.

Performance Across Noise Ratios. The results

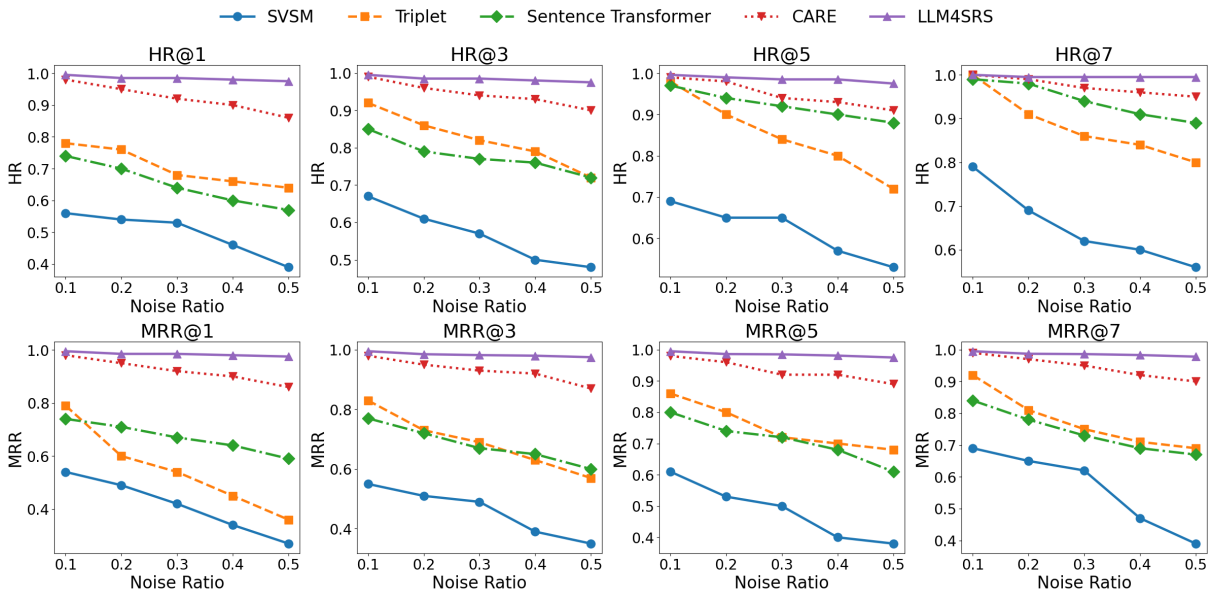

Figure 5: TKY@10 candidate regions: Performance comparison with 10 candidates of LLM4SRS with SVSM, Triplet, Sentence Transformer, and CARE across noise ratios (η) and evaluation metrics (HR@k, MRR@k) for the *Tokyo* datasets.

show that LLM4SRS consistently outperforms the baseline methods across all noise ratios. Specifically, for $\eta = 0.1$, the accuracy improvement of LLM4SRS over CARE, the closest baseline, is up to **12%** on the *Singapore* dataset. For $\eta = 0.3$, the relative gain in HR@5 reaches **15%** for the *Tokyo* dataset, demonstrating the robustness of LLM4SRS in handling moderate levels of noise. At higher noise levels (e.g., $\eta = 0.5$), the performance of all methods decline, but LLM4SRS maintain a margin of **8–10%** improvement in accuracy compared to Sentence Transformer and **10–13%** over SVSM on both datasets. This indicates the superiority of our model in noisy environments.

Performance Across Different k. LLM4SRS significantly improves HR and MRR metrics for all values of k. For example, on the *Singapore* dataset with 10 candidates, LLM4SRS demonstrates an increase of **20%** in MRR@3 compared to Triplet. Similarly, on the *Tokyo* dataset with 20 candidates, LLM4SRS achieves **18%** higher HR@7 than CARE.

For smaller values of k (e.g., $k = 1$), LLM4SRS shows its ability to rank the correct candidate as the top result, outperforming Sentence Transformer by **15%** on average. As k increases, the improvements in HR and MRR remains consistent, highlighting the scalability of LLM4SRS.

Effect of Candidate Size. We analyze the impact of candidate size by comparing results for 10 and 20 candidates. Across both settings, LLM4SRS demonstrates stable and robust performance. With 10 candidates, the model achieves higher HR and MRR values, particularly under lower noise ratios and for larger k, reflecting the reduced ranking difficulty in smaller candidate sets. When the candidate size increases to 20, performance slightly declines as expected, yet LLM4SRS consistently maintains a clear margin over baseline methods. Figures 4, 5 and Figures 8, 9 (in Appendix B.1) show these trends, reinforcing the observed effects of candidate size on performance.

5.2.2 Ablation Study.

In this section, we evaluated the impact of different features used in our model, namely POI attributes (POI), check-in data (CK), and building footprint (BF), on the model's performance across various k values (1, 3, 5, 7) and noise ratios (0.1 to 0.5). Figure 6 shows that LLM4SRS achieves the highest performance when all features (POI, CK, BF) are used, with HR and MRR values peaking at 1.0 in certain cases. Removing CK (POI, BF) leads to a slight decline, while excluding BF (POI, CK) results in a more significant drop, particularly at higher noise ratios, underscoring the importance of BF. The lowest performance was observed when using only the POI feature, highlighting the necessity of integrating multiple regional features for enhanced accuracy.

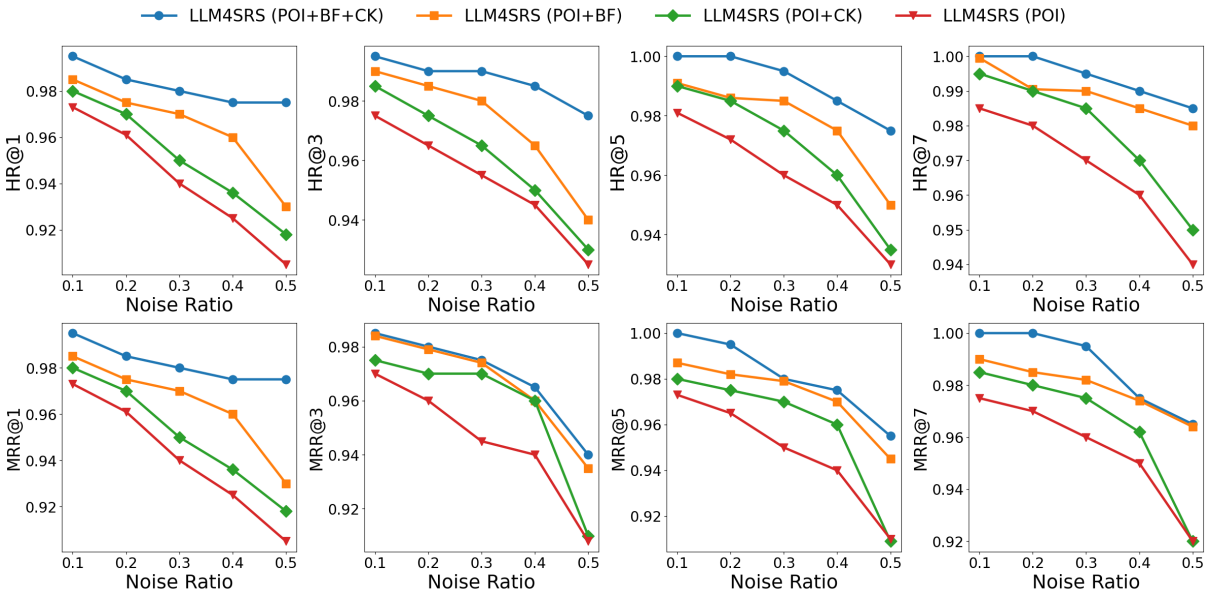

Figure 6: Performance comparison for different feature combinations.

6 Conclusion & Future Works

We introduced a novel framework for region similarity search that leverages large language models (LLMs) to provide a more context-aware and adaptable approach. Unlike traditional methods that rely on rigid distance metrics or numerical embeddings, our approach transforms spatial features into natural language descriptions, allowing for a richer understanding of regional characteristics. Our controlled noise injection technique enables self-supervised learning, reducing dependence on extensive labeled data while improving robustness.

7 Limitations

Evaluation Scope. In this work, region similarity is defined using known perturbations of high-dimensional numerical spatial features, and the evaluation is carried out under a controlled, self-supervised protocol. Although this makes benchmarking dependable and repeatable, it lacks external human or application-specific similarity labels. Without visualization or domain knowledge, annotators cannot directly interpret the aggregated numerical statistics of POI distributions, check-in volumes, and building footprints that represent regions in our setting. Human judgments would therefore rely on subjective heuristics in the absence of well-defined visual or application-level ground truth. Thus, we concentrate on data-driven evaluation in line with previous work on region similarity learning. Future research focus on creating can application-grounded or human-in-the-loop evaluation protocols, which could be aided by expert knowledge or visualization.

Inference Cost. For similarity ranking, our method uses large language models, which are typically more computationally expensive than lightweight embedding-based techniques like SVSM, Triplet networks, or CARE. This trade-off is deliberate because our objective is not to maximize runtime efficiency but rather to investigate whether LLMs can extract contextual and functional similarity from heterogeneous spatial features. In actuality, inference is carried out over small candidate sets and benefits from quantized models and parameter-efficient fine-tuning, which makes the latency tolerable for offline analysis. Therefore, we position LLM4SRS as a complementary approach appropriate for high-value decision-making scenarios and exploratory analysis, where contextual reasoning and accuracy are more important than real-time performance. Future research can focus on increasing efficiency through smaller models, distillation, or hybrid retrieval–reranking pipelines.

Model and Modal Constraints. While our proposed framework demonstrates promising potential for integrating spatial reasoning with LLMs, several other limitations remain. First, our experiments are based on a single backbone model, LLaMA-3, which may limit generalizability across

architectures with different reasoning or spatial understanding capabilities. Evaluating other LLMs such as GPT-4, Gemma, or Mistral could provide deeper insights into model-specific strengths and adaptability. Second, although natural language representations enable contextual flexibility, they introduce additional variability depending on prompt phrasing, potentially affecting consistency and reproducibility. Finally, the current framework primarily focuses on text-based spatial descriptions and does not yet incorporate visual or graph-based modalities (e.g., satellite imagery, road networks), which could further enhance the richness of spatial similarity reasoning.

Acknowledgements

Tanzima Hashem is supported by the Basic Research Grant from Bangladesh University of Engineering and Technology (BUET).

AI Use Statement

The authors used AI-based tools exclusively for editing and improving the clarity of the manuscript. These tools were not involved in the development of the system, experimental design, data analysis, or conclusions.

References

Deniz Canturk, Pinar Karagoz, Sang-Wook Kim, and Ismail Hakki Toroslu. 2023. Trust-aware location recommendation in location-based social networks: A graph-based approach. *Expert Systems with Applications*, 213:119048.

Gang Cao, Shengmin Cui, and Inwhee Joe. 2023. Improving the spatial–temporal aware attention network with dynamic trajectory graph learning for next point-of-interest recommendation. *Information Processing & Management*, 60(3):103335.

Weiliang Chan and Qianqian Ren. 2023. Region-wise attentive multi-view representation learning for urban region embedding. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3763–3767.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Preprint*, arXiv:2205.14135.

Kaiyu Feng, Gao Cong, Christian S Jensen, and Tao Guo. 2019. Finding attribute-aware similar region for data analysis. *Proceedings of the VLDB Endowment*, 12(11):1414–1426.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Bo Hui, Da Yan, Wei-Shinn Ku, and Wenlu Wang. 2020. Predicting economic growth by region embedding: A multigraph convolutional network approach. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 555–564.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2017. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *Preprint*, arXiv:1712.05877.

Yue Jiang, Qin Chao, Yile Chen, Xiucheng Li, Shuai Liu, and Gao Cong. 2024. Urbanllm: Autonomous urban activity planning and management with large language models. *arXiv preprint arXiv:2406.12360*.

Jiahui Jin, Yifan Song, Dong Kan, Binjie Zhang, Yan Lyu, Jinghui Zhang, and Hongru Lu. 2024a. Learning context-aware region similarity with effective spatial normalization over point-of-interest data. *Information Processing & Management*, 61(3):103673.

Jiahui Jin, Yifan Song, Dong Kan, Binjie Zhang, Yan Lyu, Jinghui Zhang, and Hongru Lu. 2024b. Learning context-aware region similarity with effective spatial normalization over point-of-interest data. *Information Processing & Management*, 61(3):103673.

Xiongnan Jin, Byungkook Oh, Sanghak Lee, Dongho Lee, Kyong-Ho Lee, and Liang Chen. 2019. Learning region similarity over spatial knowledge graphs with hierarchical types and semantic relations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 669–678.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Peibo Li, Maarten de Rijke, Hao Xue, Shuang Ao, Yang Song, and Flora D Salim. 2024a. Large language models for next point-of-interest recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1463–1472.

Yi Li, Weiming Huang, Gao Cong, Hao Wang, and Zheng Wang. 2023. Urban region representation learning with openstreetmap building footprints. In *Proceedings of the 29th ACM SIGKDD Conference*

on Knowledge Discovery and Data Mining, pages 1363–1373.

Zechen Li, Weiming Huang, Kai Zhao, Min Yang, Yongshun Gong, and Meng Chen. 2024b. Urban region embedding via multi-view contrastive prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8724–8732.

Zhonghang Li, Lianghao Xia, Jiabin Tang, Yong Xu, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. 2024c. Urbangpt: Spatio-temporal large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5351–5362.

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

Yu Liu, Jingtao Ding, Yanjie Fu, and Yong Li. 2023. Urbankg: An urban knowledge graph system. *ACM Transactions on Intelligent Systems and Technology*, 14(4):1–25.

Rohin Manvi, Samar Khanna, Gengchen Mai, Marshall Burke, David Lobell, and Stefano Ermon. 2023. Geollm: Extracting geospatial knowledge from large language models. *arXiv preprint arXiv:2310.06213*.

Grant McKenzie and Daniel Romm. 2021. Measuring urban regional similarity through mobility signatures. *Computers, Environment and Urban Systems*, 89:101684.

Chang Sheng, Yu Zheng, Wynne Hsu, Mong Li Lee, and Xing Xie. 2010. Answering top-k similar region queries. In *Database Systems for Advanced Applications: 15th International Conference, DASFAA 2010, Tsukuba, Japan, April 1-4, 2010, Proceedings, Part I 15*, pages 186–201. Springer.

Fengze Sun, Jianzhong Qi, Yanchuan Chang, Xiaoliang Fan, Shanika Karunasekera, and Egemen Tanin. 2024. Urban region representation learning with attentive fusion. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 4409–4421. IEEE.

Xigang Sun, Jingya Zhou, Ling Liu, and Wenqi Wei. 2023. Explicit time embedding based cascade attention network for information popularity prediction. *Information Processing & Management*, 60(3):103278.

Shangbin Wu, Xu Yan, Xiaoliang Fan, Shirui Pan, Shichao Zhu, Chuanpan Zheng, Ming Cheng, and Cheng Wang. 2022. Multi-graph fusion networks for urban region embedding. *arXiv preprint arXiv:2201.09760*.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *Preprint*, arXiv:2312.12148.

Liang Zhang, Cheng Long, and Gao Cong. 2022. Region embedding with intra and inter-view contrastive learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9031–9036.

Siyao Zhang, Daocheng Fu, Wenzhe Liang, Zhao Zhang, Bin Yu, Pinlong Cai, and Baozhen Yao. 2024. Trafficgpt: Viewing, processing and interacting with traffic foundation models. *Transport Policy*, 150:95–105.

Yifan Zhang, Cheng Wei, Shangyou Wu, Zhengting He, and Wenhao Yu. 2023. Geogpt: understanding and processing geospatial tasks through an autonomous gpt. *arXiv preprint arXiv:2307.07930*.

Silin Zhou, Dan He, Lisi Chen, Shuo Shang, and Peng Han. 2023. Heterogeneous region embedding with prompt learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4981–4989.

A Prompt

Region r_j Description, $\mathcal{D}_j$

Region r_j has the following POIs:

$c_1 \Rightarrow$ **count**: $POICounts_{1,j}$, **total size**: $TotalSize_{1,j}$, **total check-in count**: $CheckIns_{1,j}$

$c_2 \Rightarrow$ **count**: $POICounts_{2,j}$, **total size**: $TotalSize_{2,j}$, **total check-in count**: $CheckIns_{2,j}$

⋮

$c_k \Rightarrow$ **count**: $POICounts_{k,j}$, **total size**: $TotalSize_{k,j}$, **total check-in count**: $CheckIns_{k,j}$

Region r_j has the following building footprints:

$b_1 \Rightarrow$ **count**: $BuildingCounts_{1,j}$

$b_2 \Rightarrow$ **count**: $BuildingCounts_{2,j}$

⋮

$b_m \Rightarrow$ **count**: $BuildingCounts_{m,j}$

(a) Structure of region description using raw data, like, POI count, size, check-ins, building footprints

Region Prompt

<instruction> Analyze the input data to identify candidate regions that are spatially similar to the anchor region. Your analysis should intelligently account for the spatial characteristics of each region, including POI count, distribution, category size, and building footprint information. Based on this, rank the candidate regions according to their similarity to the anchor in descending order of similarity, i.e., the top similar candidate regions will be ranked first.

<input> **[anchor, $\mathcal{D}_a$] : [candidates, $\mathcal{D}_g$'s]**

<expected output format> A list containing the ranked IDs of candidate regions most similar to the anchor, e.g.,

`[<region_id>, <region_id>, <region_id>, ...]`

<output>The actual similarity ranking of candidate regions to the anchor is: `[rankings]`

(b) Generation of prompt using region description

Figure 7: Prompt construction pipeline.

B Additional Results

B.1 Additional Results on SG and TKY

Figures 8 and 9 additionally shows the performance of LLM4SRS and baseline methods when the candidate set size is increased to 20 for both Singapore and Tokyo. Compared to the 10-candidate setting in the main results, all methods exhibit a natural performance degradation due to the increased ranking difficulty.

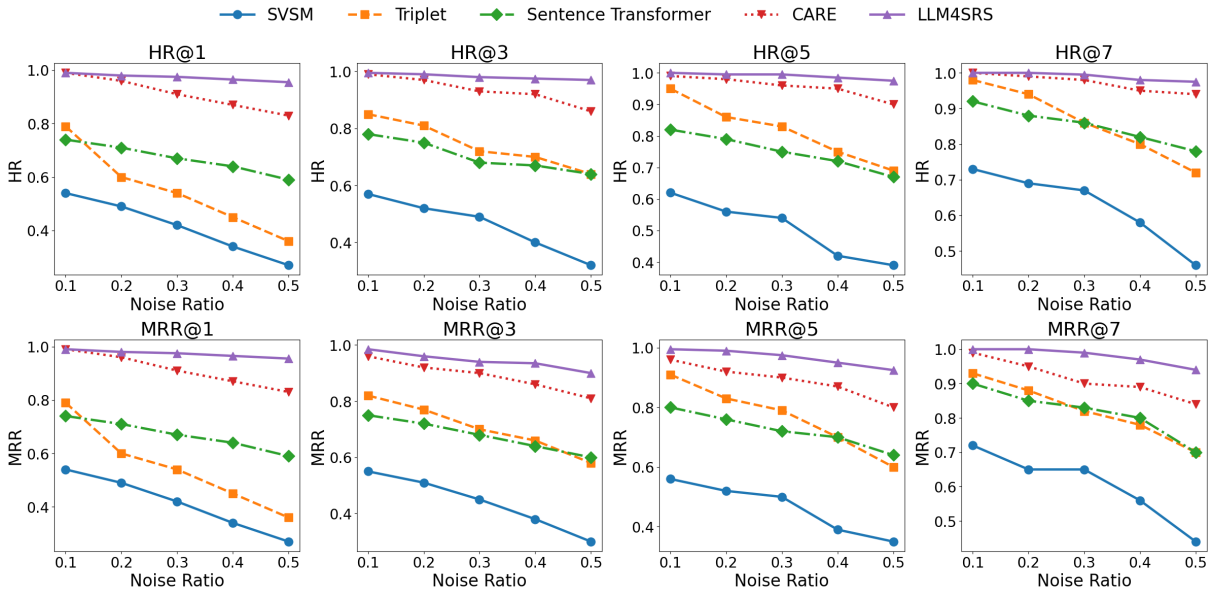

Figure 8: SG@20 candidate regions: Performance comparison with 20 candidates of LLM4SRS with SVSM, Triplet, Sentence Transformer, and CARE across noise ratios (η) and evaluation metrics (HR@k, MRR@k) for the *Singapore* datasets.

However, LLM4SRS consistently maintains a clear performance margin over all baselines across noise ratios and evaluation metrics. Notably, the relative advantage of LLM4SRS becomes more pronounced at higher noise levels ($\eta \geq 0.3$), indicating that the proposed framework is more robust to feature

perturbations even under larger candidate pools. These results further validate the scalability of LLM4SRS and confirm that its superior performance is not limited to small candidate sets.

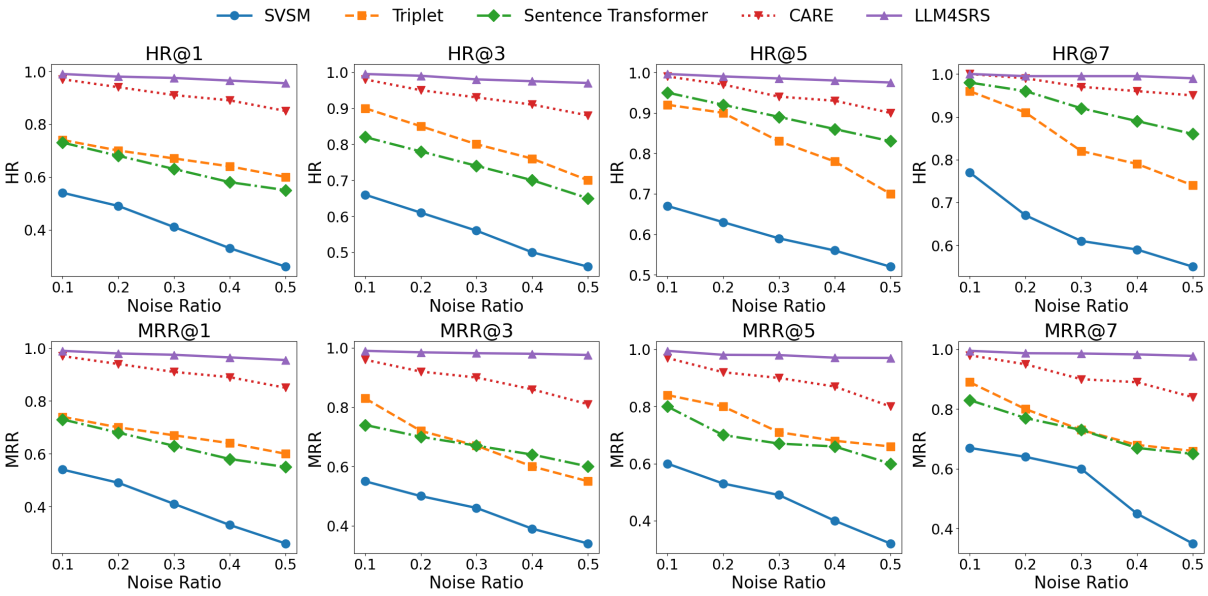

Figure 9: TKY@20 candidate regions: Performance comparison with 20 candidates of LLM4SRS with SVSM, Triplet, Sentence Transformer, and CARE across noise ratios (η) and evaluation metrics (HR@k, MRR@k) for the *Tokyo* datasets.

B.2 Additional Results on NYC

Figures 10 and 11 presents the performance of LLM4SRS on the New York City (NYC) dataset under both 10 and 20 candidate settings. Importantly, the model is evaluated on NYC without any city-specific fine-tuning, demonstrating its cross-city generalization capability. Despite differences in urban structure, POI distributions, and building characteristics between NYC and the training city (Singapore), LLM4SRS maintains stable HR@k and MRR@k performance across all noise ratios.

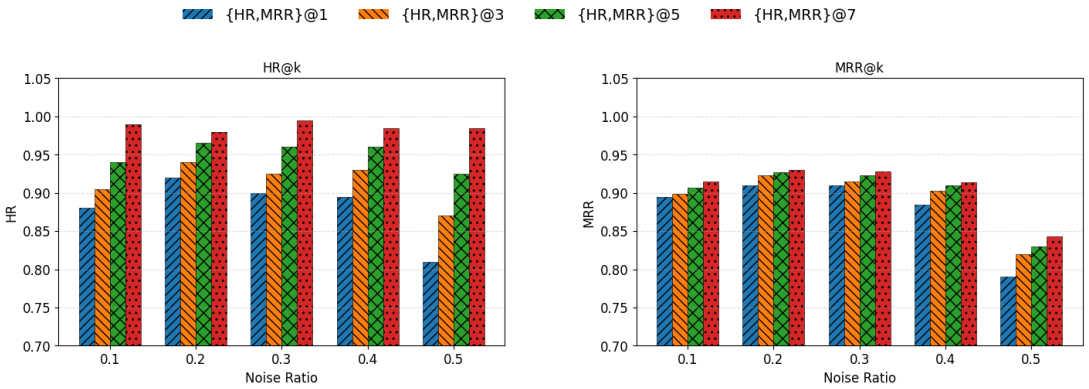

Figure 10: HR@k (left) and MRR@k (right) performance of LLM4SRS on the NYC dataset with 10 candidate regions under different noise ratios.

While higher noise levels naturally reduce ranking accuracy, the overall trends remain consistent with

those observed in Singapore and Tokyo, indicating that the model captures transferable, high-level spatial semantics rather than city-dependent patterns.

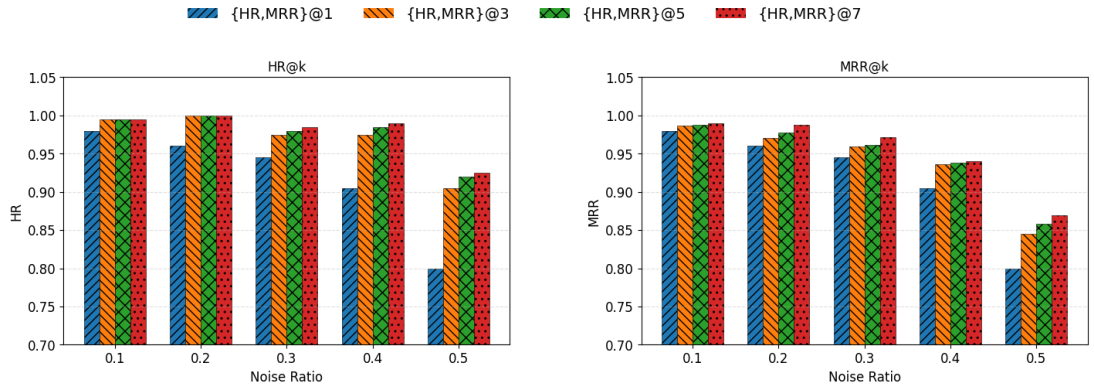

Figure 11: HR@k (left) and MRR@k (right) performance of LLM4SRS on the NYC dataset with 20 candidate regions under different noise ratios.