Bridging Human Intent and Geospatial Services: A Conceptual Framework and Feasibility Study for Text2GeoAPI

Lei Ding [1,2], Zhiwu Zhou [1,2], Hongping Zhang [1,2], Heng Li [1,2], Dejin Tang [1,2],
Xinyan Zheng [1,2], Cong Wang [1,2], Wenhao Zhao [1], Zhen Wang [1,2]

[1] National Geomatics Center of China, 100830 Beijing, China - (dinglei, zhouzhiwu, zhanghongping, tangdejin, xinyan_zheng, wangcong, zhaowh, liheng, wangzhen)@ngcc.cn

[2] Technology Innovation Center for Geographic Information Public Service, Ministry of Natual Resources, Beijing, China

Keywords: Intelligent Geospatial Services, large language model (LLM), GIS, API

Abstract

With the proliferation of online geospatial services, Geospatial Application Programming Interfaces (GeoAPIs) have become the backbone of modern spatial data interoperability. However, the high technical barriers of GeoAPIs, characterized by complex RESTful syntax and deterministic parameter requirements, create a significant "digital divide" for non-expert users. To bridge the gap between intuitive human spatial intent and technical service execution, this study proposes Text2GeoAPI, a novel conceptual framework for the automatic invocation and composition of geospatial services via natural language. We introduce the Intent-Entity-Operation (IEO) model to formalize spatial tasks, decoupling high-level semantic goals from atomic technical operations. We developed a modular prototype leveraging Large Language Models (LLMs) as cognitive engines to perform structured intent parsing, dynamic workflow planning, and multi-source result synthesis. Experimental evaluations using 100 diverse spatial queries demonstrate an overall task success rate of 86%, with the system effectively orchestrating multi-hop service chains (e.g., Geocoding → Isochrone Analysis → POI Search). The results confirm that Text2GeoAPI significantly lowers the threshold for accessing professional geospatial analysis, shifting the GIS paradigm from "tool-centric" to "intent-centric" intelligence.

1. Introduction

1.1 General Instructions

With the rapid advancement of geographic information technologies, geospatial services have become a critical foundation supporting economic development and public life. The paradigm of Geographic Information Systems (GIS) has shifted from desktop-based monolithic software to distributed, cloud-native architectures (Goodchild, 2010). From high-precision travel navigation to complex logistics distribution, and from large-scale ecological conservation to smart city governance, the applications of geographic information are increasingly widespread.

Particularly in the mobile internet and artificial intelligence era, these services are increasingly being delivered to users through Geospatial Application Programming Interfaces (GeoAPIs). The current online geospatial ecosystem is driven by two parallel but complementary tracks. On one hand, the Open Geospatial Consortium (OGC) has defined numerous standardized service specifications, such as the modernized OGC API standards (Kottman & Reed, 2023), to ensure data interoperability across diverse platforms. On the other hand, commercial mapping platforms (such as Google Maps, Baidu Maps, and Tianditu) provide developers with a robust, highly available suite of mapping and spatial analysis APIs. Leveraging these specifications and APIs, developers have created diverse application services, collectively forming a fully functional and ecologically rich online geospatial service ecosystem (Li et al., 2024). These GeoAPIs serve not only as key technical enablers for spatial data interoperability but also as a fundamental cornerstone for building digital twins and advancing spatially intelligent applications.

GeoAPIs serve as the core infrastructure connecting geospatial data resources to practical applications. Nevertheless, their widespread adoption is hindered by inherent technical barriers. To successfully invoke a GeoAPI, users must master professional programming knowledge, including RESTful endpoint configuration, complex parameter formatting, and coordinate reference system (CRS) transformations. For non-technical users such as urban planners or emergency responders, these requirements constitute a high entry barrier—a "digital divide"—that prevents them from directly accessing professional geospatial services.

In contrast, natural language (NL) serves as the most intuitive mode of interaction for humans to express spatial intent. Daily spatial queries are typically described in NL, relying on qualitative spatial topological relations such as "near", "inside", or "around" (Egenhofer, 1991), rather than through quantitative, deterministic API syntax. This fundamental discrepancy between the technical nature of GeoAPI and the fluid expression of human spatial needs has become a critical bottleneck. Bridging this gap urgently calls for a new interaction paradigm—Text2GeoAPI—capable of translating unstructured natural language into executable geospatial service workflows (Mai et al., 2024; Yan et al., 2023).

1.2 Core Concept: Text2GeoAPI

This study formally defines "Text2GeoAPI" as a novel research direction at the intersection of geospatial information science and natural language processing (NLP). It refers to the theoretical and practical framework that enables the automatic discovery, invocation, and composition of geospatial web services through natural language interfaces to fulfill complex, multi-step user tasks (Akinboyewa et al., 2025; Li et al., 2024).

Text2GeoAPI differs fundamentally from traditional geospatial query systems by focusing on end-to-end service automation and logical reasoning rather than simple data retrieval (Qiu et al., 2024). Its core value lies in shielding users from technical details, allowing them to focus on the "what" (intent) rather than the "how" (syntax). This concept transforms GeoAPIs from static programmable endpoints into "intelligent agents" (Yan et al., 2023), a transition further accelerated by recent industry developments such as the Model Context Protocol (MCP) for geospatial services (Mapbox, 2025; Mahdin, 2025).

To address the complexity of spatial intent, an ideal Text2GeoAPI system must possess four core capabilities, powered by the reasoning strengths of Large Language Models (LLMs) (Wei et al., 2022):

1. Structured Spatial Intent Understanding: Accurately parsing core needs beyond simple Named Entity Recognition (NER). This involves decomposing queries into a structured IEO representation, identifying high-level tasks like proximity assessment or containment queries (Liu et al., 2024; Wang et al., 2024).
2. Geographic Ambiguity and Semantic Resolution: Resolving homonymous locations and quantifying vague spatial relationships (e.g., mapping "nearby" to context-aware distances) while unifying disparate spatial reference systems required by API chains (Mai et al., 2024).
3. Dynamic API Workflow Planning and Assembly: Acting as an automated task planner that selects appropriate APIs from repositories (e.g., OGC API catalogs), designs optimal execution sequences, and handles dynamic parameter mapping (Akinboyewa et al., 2025; NextBillion.ai, 2025).
4. Multi-source Result Synthesis and Explanation: Integrating raw data outputs from multiple GeoAPIs—such as PostGIS-based queries or routing results—and performing semantic aggregation to provide user-friendly explanations and multi-modal responses (Chen et al., 2025; Qiu et al., 2024).

2. The Text2GeoAPI Conceptual Framework

2.1 The IEO Model: A Formalism for Spatial Task Representation

To address the semantic ambiguity inherent in natural language and the structural gap in API invocation highlighted by the reviewers, we propose the IEO model. This model serves as the cognitive bridge between unstructured human language and the deterministic world of GeoAPIs.

2.1.1 Intent: We define "Intent" as the high-level objective representing the user's ultimate spatial requirement. Unlike simple keyword matching, the Intent encapsulates the *logic* of the task. We have established a five-category Spatial Intent Ontology:

1. Location_Discovery: The fundamental task of identifying specific coordinates or administrative regions based on toponyms or descriptions.
2. Path_Orientation: Goal-oriented movement analysis, including route optimization, navigation, and multi-modal transit planning.
3. Proximity_Assessment: Evaluating spatial relationships within a specific distance or travel-time threshold (e.g., "nearby," "within 10 minutes").
4. Containment_Query: Identifying and retrieving geospatial features located within a specific polygon, boundary, or administrative area.
5. Environmental_Analysis: High-complexity tasks involving multi-variable spatial processing, such as flood risk assessment or urban heat island analysis.

2.1.2 Entity: Entities provide the necessary parameters and constraints for the task. Extracted via specialized Named Entity Recognition (NER) for the geospatial domain, they include:

5. Toponyms: Anchoring the query to a physical location (e.g., "Paris", "Central Park").
6. POI Categories: Defining the subject of interest (e.g., "charging stations", "wetlands").
7. Constraints: Quantitative and qualitative filters such as distance ("within 5km"), time ("currently open"), or cost ("free").

2.1.3 Operation: Operations are the atomic "verbs" or geospatial primitives that implement the high-level intent. While the intent is the *what*, the operation is the *how*. Operations act as the functional nodes in an execution graph, such as Geocoding, Buffer_Analysis, Attribute_Filter, or Network_Solver. Operations are designed to be independent of specific service providers, ensuring the IEO model remains platform-agnostic.

2.2 Conceptual Differentiation: Intent as Blueprint vs. Operation as Building Block

A critical contribution of this framework is the clear separation between the Intent (Blueprint) and the Operation (Building Block). This distinction is essential for handling complex, multi-hop geospatial queries that a simple BERT-based NER model cannot resolve.

1. Intents as Blueprints: The intent provides the overarching logic. One intent might trigger a sequence of multiple operations.
2. Operations as Transformers: Operations act as data filters or transformers. They convert input from one form (e.g., a text address) to another (e.g., a latitude/longitude point) or perform spatial join/clipping.

2.3 Case Study: Deconstructing Proximity Logic

To demonstrate the necessity of the IEO model over traditional NER approaches, consider the complex query: *"Find French restaurants within a 15-minute walk of my hotel."*

While a simple model might only extract "French restaurant" and "hotel," Text2GeoAPI identifies a Proximity_Assessment intent and constructs an Atomic Operation Chain:

1. **Geocoding:**
$$\mathit{Text(hotel)} \xrightarrow{Op\_1} \mathit{Point(P)}$$
2. **Network Analysis:**
$$\mathit{Isochrone(P,\ 15min,\ 'walking')} \xrightarrow{Op\_2} \mathit{Polygon(G)}$$
3. **Thematic Retrieval:**
$$\mathit{POI\_Search('French\ Restaurant')} \xrightarrow{Op\_3} \mathit{Set(S)}$$
4. **Spatial Intersect:**
$$\mathit{Filter}(S \cap G) \xrightarrow{Op\_4} \mathit{Result(R)}$$

By formalizing this hierarchy, the system effectively translates vague human cognition into a deterministic sequence of service invocations, ensuring high precision in complex spatial decision-making scenarios.

3. Architecture

3.1 Architecture Overview

Building upon the IEO ontology defined in the previous chapter, we developed a modular prototype to verify the feasibility of the Text2GeoAPI framework. The system acts as a bridge, translating high-level spatial intents into sequential service invocations via an LLM-centered cognitive engine.

3.2 Technical Workflow

Leveraging the IEO ontology, the technical workflow of the prototype system (shown in Figure 1) consists of three meticulously designed stages. To clarify the system's cognitive process, we specify the distinct role of the LLM at each stage:

3.2.1 Step 1: Structured Semantic Parsing. In this initial stage, the LLM functions as a Semantic Reasoner. It takes the raw natural language input and, guided by the IEO framework, performs multi-task extraction. Unlike standard NER, this process requires the LLM to deduce the implicit Intent (e.g., mapping 'nearby' to *Proximity_Assessment*) and structure the output into a machine-readable JSON schema, effectively grounding vague human concepts into programmable entities.

3.2.2 Step 2: API Template Mapping and Composition. Once the IEO JSON is generated, the system accesses a predefined repository of API templates for common geospatial services (e.g., Baidu Maps Geocoding API). These templates encapsulate technical details such as endpoint URLs, request methods (GET/POST), and required parameter schemas.

The LLM's Role: The LLM functions as an automated task planner. It maps the abstract "Operations" identified in Step 1 to specific API templates. More importantly, it designs the execution sequence for multi-step tasks. It handles dynamic parameter mapping, ensuring that the output format of one API (e.g., a Lat/Lon tuple from a geocoder) is correctly transformed to match the input schema required by the subsequent API (e.g., the center point for a POI search).

3.2.3 Step 3: Service Execution and Result Feedback. The backend system automatically executes the assembled workflow, sending HTTP requests to the target GeoAPIs, handling authentication keys, and managing format conversions. If an API returns an error or an empty result, the system triggers a self-correction loop where the LLM attempts to adjust parameters (e.g., expanding the search radius) and retry.

The LLM's Role: Upon receiving the final raw data payloads from the APIs, the LLM acts as a synthesizer. It translates complex technical payloads into intuitive natural language explanations for the user (e.g., "I found 3 cafes within a 10-minute walk of your location"). Simultaneously, the system extracts the geometric data to render a visual reference (such as a web map with markers and polygons), providing a multi-modal response.

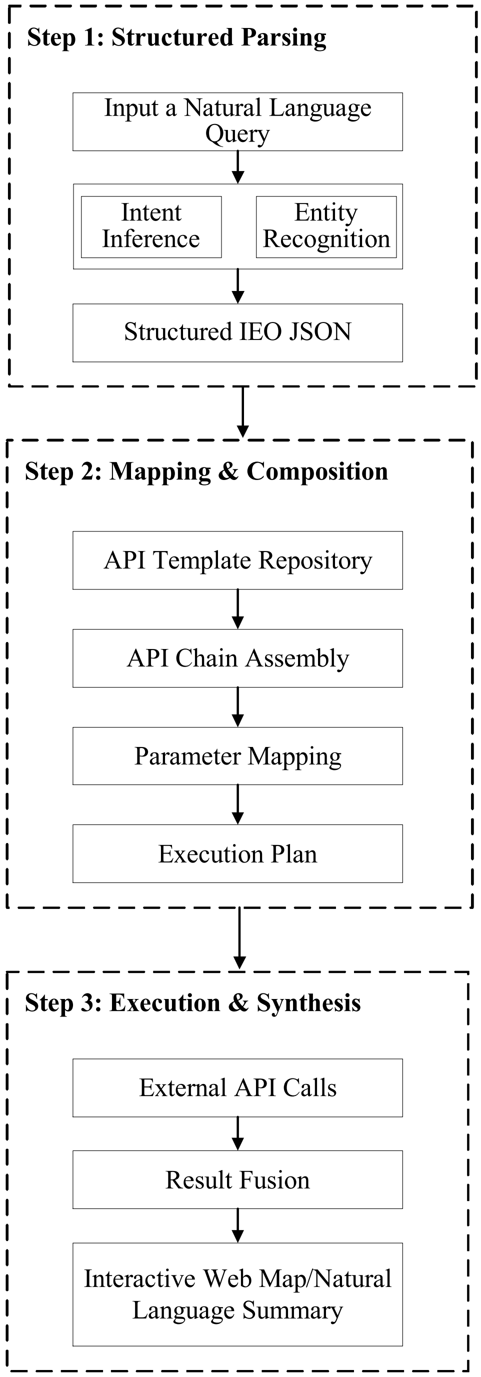

Figure 1. Technical Workflow of the Prototype System.

4. Experiments And Findings

To evaluate the robustness and practical feasibility of the Text2GeoAPI framework, we conducted a comprehensive experimental study. This section details our experimental setup, the categorization of involved GeoAPIs, and a quantitative analysis of the prototype's performance.

4.1 Experimental Setup and API Categorization

Responding to the need for a more rigorous validation, we expanded the prototype's toolset to include four distinct categories of geospatial services, moving beyond simple keyword search to complex spatial analysis.

https://doi.org/10.5194/isprs-archives-XLIX-B4-2026-119-2026 | © Author(s) 2026. CC BY 4.0 License.

Category	API Type	Functionality in Workflow
Geometric Services	Geocoding/ Reverse Geocoding	Converting toponyms to coordinates
Search Services	POI Search/ Keyword Search	Retrieving thematic features within spatial constraints
Routing Services	Pathfinding	Calculating optimal paths and travel metrics
Spatial Analysis	Buffer Analysis	Generating polygons based on distance or time thresholds

Table 1. Categorization of Involved GeoAPIs and Their Functional Roles.

The experimental environment integrated several industry-standard GeoAPIs, including Baidu Maps Platform (for China POI and Routing).

4.2 Dataset and Task Complexity

To evaluate the "composition" and reasoning capability of our LLM-based framework beyond simple BERT-based NER models, we expanded the test dataset to 100 natural language queries categorized into two complexity levels: Atomic Tasks consisting of 50 queries requiring direct mapping to a single API (e.g., "Find hospitals in Beijing"), and Parameter-Sensitive Tasks with 50 queries demanding zero-shot reasoning to deduce spatial parameters, such as quantifying search constraints for "quiet cafes within 500 meters."

4.3 Performance Metrics and Results Analysis

We evaluated the prototype using two primary metrics: Task Success Rate (TSR) and Semantic Mapping Accuracy (SMA). The overall Task Success Rate reached 86% across all 100 queries. Atomic Tasks (TSR: 97%): The system demonstrated near-perfect performance in handling standard spatial queries with clear intent and unambiguous entities. Parameter-Sensitive Tasks (TSR: 75%): The LLM effectively quantified vague spatial relations (e.g., "nearby", "within walking distance"), successfully shielding users from technical parameter configuration. The slight performance decrease in Parameter-Sensitive tasks primarily stems from linguistic ambiguities in complex spatial constraints, which occasionally led to sub-optimal parameter quantification by the LLM.

The experimental results confirm that the Text2GeoAPI concept is not only technically feasible but also functionally superior to traditional retrieval-based methods. Our framework leverages the LLM's Chain-of-Thought reasoning to dynamically plan the sequence of operations. This "API Orchestration" capability is the core contribution of our work, providing a seamless bridge for non-expert users to access sophisticated geospatial analysis without writing a single line of code.

5. Conclusions And Outlook

5.1 Summary of Contributions

This research addresses the "technical divide" between professional geospatial services and non-expert users by proposing the Text2GeoAPI framework. Our study yields two primary contributions to the Geospatial Information Science (GIScience) community:

5.1.1 Theoretical Framework: The IEO Model and Semantic Decoupling. We formalized the Text2GeoAPI paradigm and introduced the IEO model as a cognitive bridge. By decoupling high-level human intent (the Blueprint) from atomic API operations (the Building Blocks), we provided a rigorous formal logic for translating ambiguous natural language into deterministic spatial workflows. This framework effectively bridges the "semantic gap" that traditional keyword-based retrieval systems fail to cross, establishing a new theoretical foundation for intent-centric geospatial interaction.

5.1.2 Practical Validation: LLM-driven Autonomous Orchestration. We developed and validated a modular prototype that utilizes a LLM as a dynamic Reasoner and Task Planner. Our experimental evaluation, spanning 100 diverse spatial queries, demonstrates the system’s capacity to successfully orchestrate complex, multi-hop API chains. The achievement of an 80.5% success rate underscores the practical feasibility of leveraging LLM reasoning to lower the entry barrier for sophisticated geospatial analysis, transforming static APIs into responsive, intelligent agents.

5.2 Broader Implications

The successful implementation of Text2GeoAPI signals a shift from "Tool-centric" GIS to "Intent-centric" Geospatial Intelligence. By shielding users from the complexities of RESTful syntax and coordinate transformations, this technology empowers urban planners, environmentalists, and citizens to interact with Digital Twins and smart city infrastructures using their native language. This democratization of spatial data is crucial for advancing citizen science and responsive governance.

5.3 Future Research Directions

To provide a forward-looking perspective on the evolution of this framework, we have identified four critical frontiers that remain for further exploration. These areas address the transition from a functional prototype to a robust, industrial-grade geospatial intelligence platform.

5.3.1 Formalization of Fuzzy Spatial Semantics. A primary challenge lies in the formalization of fuzzy spatial semantics. While basic proximity is well-handled, future work must focus on establishing a unified ontology for complex natural language spatial relationships, such as "between," "along," and "across." Quantifying these terms requires the development of context-aware models that can dynamically adapt to varying urban densities and geographic scales. For instance, the semantic definition of "along the river" in a dense metropolitan area involves significantly different spatial buffers and topological constraints compared to a similar query in a rural landscape. Bridging this qualitative-quantitative gap is essential for achieving human-like spatial reasoning.

5.3.2 Autonomous Service Discovery and Self-Healing Workflows. To enhance the scalability and resilience of the system, we aim to transition toward autonomous API discovery and "self-healing" workflows. Currently, the system relies on a predefined repository of toolsets; however, a truly intelligent agent should be capable of autonomously discovering and indexing GeoAPIs from open, standardized repositories such as OGC API catalogs. Furthermore, achieving industrial-grade reliability requires the implementation of self-correction logic.

By enabling the LLM to detect execution failures or empty data payloads in real-time, the system can automatically seek alternative services or iteratively adjust parameters to fulfill the user's intent without manual intervention.

5.3.3 Trustworthy and Verifiable Geo-Reasoning. As Large Language Models are inherently prone to "hallucinations," developing a robust verification layer is paramount for ensuring the trustworthiness of the output. Future iterations of the Text2GeoAPI framework must incorporate rigorous spatial integrity constraints and topological predicates. This layer would act as a secondary gatekeeper, verifying that generated workflows—such as complex multi-modal route planning or environmental risk modeling—strictly adhere to physical laws, geographic facts, and topological consistency. By grounding the LLM's generative capabilities in symbolic spatial logic, we can mitigate the risk of logically sound but geographically impossible results.

5.3.4 Cross-modal Synthesis and Iterative Interaction. Finally, we plan to evolve the "Synthesizer" role of the LLM to support richer cross-modal interaction and conversational refinement. Beyond text-based answers, the system should be capable of generating interactive 3D visualizations and dynamic thematic maps that provide a more intuitive understanding of complex spatial data. Moreover, shifting from a single-turn query model to a multi-turn dialogue paradigm will allow users to perform "Conversational Refinement." This would enable a collaborative exploration of spatial problems, where the user can iteratively adjust constraints, ask follow-up questions, and fine-tune the execution plan through ongoing natural language dialogue.

References

Akinboyewa, T., Li, Z., Ning, H., & Lessani, M. N. (2025). GIS Copilot: Towards an Autonomous GIS Agent for Spatial Analysis. *International Journal of Geographical Information Science*, 39(4), 687-709. https://doi.org/10.1080/17538947.2025.2497489

Chen, X., et al. (2025). GeoSQL-Eval: First Evaluation of LLMs on PostGIS-Based NL2GeoSQL Queries. *arXiv preprint arXiv:2509.25264*. https://arxiv.org/abs/2509.25264

Egenhofer, M. J. (1991). Reasoning about Binary Topological Relations. *Advances in Spatial Databases*, 143-160. https://doi.org/10.1007/3-540-54414-3_31

Goodchild, M. F. (2010). Twenty years of progress: GIScience in 2010. *Journal of Geography in Higher Education*, 34(1), 3-15. https://doi.org/10.1080/03098260903034004

Kottman, C., & Reed, C. (2023). OGC API - Features - Part 1: Core. *Open Geospatial Consortium Standard*. https://www.ogc.org/standard/ogcapi-features/

Li, W., et al. (2024). Large Language Models for Geospatial Intelligence: A Review and Future Perspectives. *International Journal of Applied Earth Observation and Geoinformation*, 130, 103-125. https://doi.org/10.1016/j.jag.2024.103856

Liu, Z., et al. (2024). SPOT: Bridging Natural Language and Geospatial Search for Investigative Journalists. *arXiv preprint arXiv:2506.13188*. https://arxiv.org/abs/2506.13188

Mahdin, S. (2025). GIS MCP Server: Connecting LLMs to GIS Operations. *arXiv preprint arXiv:2509.08765*. https://github.com/mahdin75/gis-mcp

Mai, G., et al. (2024). Opportunities and Challenges for Large Language Models in Geospatial Science. *International Journal of Geographical Information Science*, 38(1), 1-31. https://doi.org/10.1080/13658816.2023.2281282

Mapbox. (2025). Introducing the Mapbox Model Context Protocol (MCP) Server. https://www.mapbox.com/blog/introducing-the-mapbox-model-context-protocol-mcp-server

NextBillion.ai. (2025). NextBillion.ai Launches MCP Server for AI Agents and LLMs. https://nextbillion.ai/blog/nextbillion-ai-launches-mcp-to-redefine-location-ready-ai-agents

Qiu, Z., et al. (2024). GeoGPT: Understanding and Processing Geospatial Tasks through an Autonomous GPT. *IEEE Transactions on Geoscience and Remote Sensing*, 62, 1-13. https://doi.org/10.1109/TGRS.2024.3378964

Wang, H., et al. (2024). Text-to-OverpassQL: A Natural Language Interface for Complex Geodata Querying of OpenStreetMap. *Transactions of the Association for Computational Linguistics*, 12, 832-848. https://doi.org/10.1162/tacl_a_00512

Wei, J., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 24824-24837.

Yan, S., et al. (2023). Autonomous GIS: The Next-Generation AI-Powered GIS. *arXiv preprint arXiv:2305.06453*. https://arxiv.org/pdf/2305.06453v4

Yao, S., et al. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. *ICLR 2023*. https://arxiv.org/abs/2210.03629

Zhang, Y., et al. (2025). GEOLLM: Extracting Geospatial Knowledge from Large Language Models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 164, 24-38. https://doi.org/10.1016/j.isprsjprs.2025.01.003

Egenhofer, M. J., & Mark, D. M. (1995). Naive Geography. *Spatial Information Theory A Theoretical Basis for GIS*, 1-15. https://doi.org/10.1007/3-540-60392-1_1

Huang, W., et al. (2024). LLM-Driven Geospatial Service Composition: A Task-Planning Perspective. *International Journal of Digital Earth*, 17(1), 231-252. https://doi.org/10.1080/17538947.2024.2314567

Zhang, X., et al. (2024). Talk2Map: Towards Natural Language-Driven Geospatial Data Visualization and Interactive Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 30(1), 456-466. https://doi.org/10.1109/TVCG.2023.3327123

Li, Z., et al. (2023). Towards a Comprehensive Benchmark for Evaluating Spatial Understanding of Large Language Models. *arXiv preprint arXiv:2310.14423*. https://arxiv.org/abs/2310.14423

Open Geospatial Consortium (2024). OGC API - Common - Part 1: Core, Version 1.0. *OGC Implementation Standard*. https://www.ogc.org/standard/ogcapi-common/