

Review

Not peer-reviewed version

---

# A Comprehensive Survey of Agentic AI for Spatio-Temporal Data

---

[Mohammad Hashemi](#) \* and [Andreas Züfle](#)

Posted Date: 28 January 2026

doi: 10.20944/preprints202601.2236.v1

Keywords: agentic AI; spatio-temporal data mining; LLMs



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Review

# A Comprehensive Survey of Agentic AI for Spatio-Temporal Data

Mohammad Hashemi \* and Andreas Züfle

Emory University

\* Correspondence: mohammad.hashemi@emory.edu

## Abstract

Recent advances in large language models (LLMs) have enabled agentic AI systems that go beyond single-pass generation by combining reasoning with tool-mediated actions. Spatio-temporal domains are a natural but challenging setting for this paradigm, requiring agents to integrate heterogeneous modalities, operate under spatial and temporal constraints, and interact reliably with external resources such as GIS libraries, map services, and Earth observation pipelines. This survey provides a comprehensive overview of agentic AI for spatio-temporal intelligence and introduces a unified taxonomy spanning (i) spatio-temporal *data modalities*, (ii) *core agentic capabilities*, and (iii) the *application landscape* across geospatial analysis, remote sensing, urban planning, and mobility. A detailed paper list is provided at <https://github.com/mohammadhashemii/awesome-agentic-AI-for-ST>.

**Keywords:** agentic AI; spatio-temporal data mining; LLMs

## 1. Introduction

Large language models (LLMs) have rapidly become a core building block of modern artificial intelligence, exhibiting strong capabilities in language understanding, reasoning, code generation, and multimodal perception [1]. Building on these advances, a new class of AI systems referred to as agentic AI has emerged, extending LLMs beyond single-pass generation by enabling goal-driven behavior, sequential decision-making, memory, and interaction with external tools [2,3]. While early LLM-based systems primarily produce isolated outputs, AI agents typically operate as single-entity systems that leverage tools and step-by-step reasoning to complete well-defined tasks, and agentic AI systems further generalize this paradigm through coordinated, multi-agent architectures in which specialized agents collaborate, communicate, and dynamically allocate sub-tasks toward shared objectives [4].

Spatio-temporal(ST) applications are inherently knowledge-intensive, requiring the integration of heterogeneous data sources such as maps, remote sensing(RS) imagery, spatial databases, sensor streams, and unstructured text, making them a natural fit for agentic AI [5]. A wide range of ST problems further demand multi-step, adaptive reasoning, such as decomposing high-level spatial queries into executable sub-queries [6,7], orchestrating tool-based GIS [8] and remote sensing pipelines [9–12], or dynamically coordinating specialized agents for perception, reasoning, and execution in complex workflows [13,14].

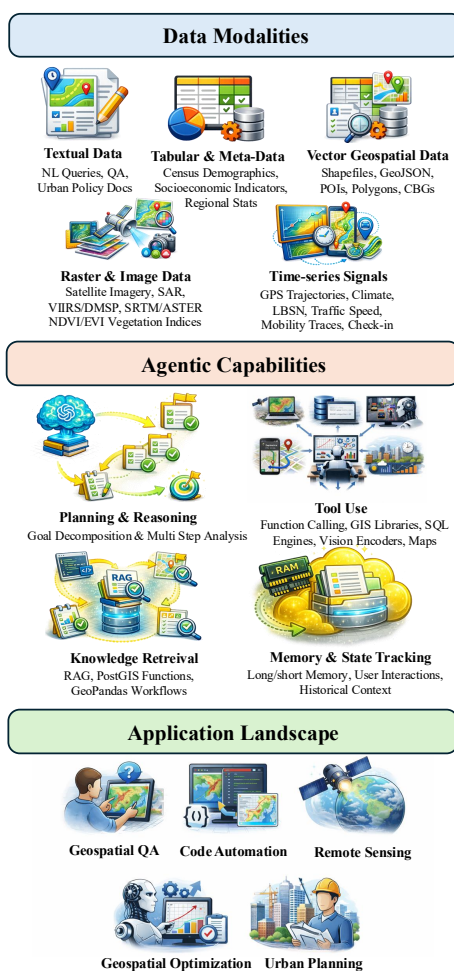
Despite its potential, agentic AI remains harder to deploy in ST settings, where mistakes in grounding or tool use can cascade across complex spatial workflows. Unlike domains where agents primarily interact with text or well-defined APIs, ST agents must reliably ground decisions in geometric relationships, spatial constraints, and temporal dynamics, often at large spatial scales and under incomplete or noisy observations. This makes autonomy particularly challenging, as small mistakes in spatial reasoning may amplify into incorrect conclusions [8,11,15]. Moreover, many ST applications demand transparency, interpretability, and user control, as practitioners must be able to inspect intermediate spatial outputs, validate reasoning steps, and retain authority over final decisions [9,14]. While agentic AI for spatio-temporal domains has advanced rapidly, a comprehensive survey of how

agentic architectures, capabilities, and tools align with spatio-temporal data characteristics remains missing. This survey fills this gap by providing the first comprehensive synthesis of agentic AI for ST data, organizing the emerging literature across input data modalities, agent capabilities, and application domains. This work clarifies current progress, highlights open challenges, and establishes a foundation for future research in agentic ST intelligence.

The remainder of this survey is organized as follows. We first introduce a conceptual framework for agentic AI pipelines operating on spatio-temporal data. Section 2.1 reviews data modalities, Section 2.2 surveys agentic capabilities, and Section 2.3 presents the application landscape. Finally, Section 3 discusses open challenges and future research directions.

## 2. Agentic AI for Spatio-Temporal Data

We introduce a unified conceptual framework that groups and delineates existing work on agentic AI for spatio-temporal systems, connecting heterogeneous data inputs and core agentic AI capabilities to a structured spectrum of applications. The framework comprises three key components: *Data Modalities*, *Agentic Capabilities*, and *Applications Landscape*. The overview of the proposed framework is illustrated in Figure 1.



**Figure 1.** A conceptual framework for agentic AI in spatio-temporal domains, illustrating the flow from data modalities and core agentic capabilities to an application landscape.

### 2.1. Data Modalities

The effectiveness of agentic AI systems for ST data depends on how agents perceive and interpret their environments. Unlike conventional models operating on fixed inputs, agentic systems interact with dynamic data sources and external tools, where the data modality directly shapes reasoning,

action space, and agent roles. This section reviews how different modalities define distinct perceptual interfaces for agentic systems, organized according to the categories in Table 1.

**Table 1.** Overview of data modalities, representative usage, and example agentic AI systems for spatio-temporal data. Each modality defines a distinct perceptual interface through which agents interpret spatial and temporal information and coordinate actions.

Data Modality	Input Type	Representative Usage	Example Studies
Textual Data	Task instructions, analytical queries, urban-planning documents	Geospatial analyst, urban planner	GeoAgent [16], GeoCogent [17], GeoColab [18], PlanGPT [19], GeoQA [20]
Structured Tabular & Metadata	Relational tables, optimization parameters, model metadata	Data analyst, decision-support agent	[6], REMSA [21], AgentAD [15], GeoBenchX [8]
Vector Geospatial Data	GIS layers, shapefiles, GeoJSON, spatial graphs	GIS operator, spatial analyst	ShapefileGPT [22], GeoJSON [23], GeoAgent [7], MapAgent [24]
Raster & Image Data	Optical, SAR, multi-spectral satellite and aerial imagery	Earth observation analyst	RS-Agent [11], ThinkGeo [10], GeoLLM-Engine [25], VICoT-Agent [26], Earth-agent [27]
Time-series Signals	Mobility trajectories, check-ins, navigation histories	Urban resident simulator, navigation agent	LLMob [28], PReP [29]
Map-Centric Multimodality	Map images, POIs, routes, APIs, spatial metadata	Navigation assistant, location-based assistant	MapAgent [24], MapBot [30], GeoLLM-Squad [9]

### 2.1.1. 3.1.1 Textual Data

Natural language is a primary modality for many agentic AI in ST domains, particularly in settings where users express analytical intent, planning objectives, or spatial queries in free-form text. In these environments, perception is not limited to extracting facts but involves interpreting intent, grounding language in spatial concepts, and maintaining context across multi-step interactions. GeoAgent [7] treats perception as understanding task instructions and translating them into executable geospatial code using LLMs, requiring the agent to reason over Python libraries and APIs while preserving the semantics of spatial operations. GeoCogent [17] and GeoColab [18] decompose textual requests into stages such as requirement parsing, algorithm design, and code debugging, highlighting how text perception directly governs downstream planning and tool use. In urban decision-making contexts, PlanGPT [19] operates over planning documents and professional QA benchmarks, where perception involves synthesizing regulatory text and domain knowledge through retrieval-augmented generation. Similarly, GeoQA [20] frames perception as semantic parsing of natural language queries over GIS layers, requiring precise alignment between linguistic expressions and spatial entities.

### 2.1.2. 3.1.2 Structured Tabular and Metadata

Many ST tasks rely on structured data such as relational tables, optimization parameters, or model metadata. In these settings, perception shifts from linguistic interpretation to schema-aware reasoning, where agents must understand data organization and constraints before acting.

Redd et al. [6] exemplify this modality by requiring agents to translate natural language queries into SQL over check-in datasets, grounding temporal and spatial predicates in database schemas. REMSA [21] further emphasizes metadata-centric perception, where the agent interprets user constraints and matches them against structured descriptions of remote sensing foundation models to support informed model selection. In optimization-focused environments, AgentAD [15] operates over satellite scheduling parameters, where perception is symbolic rather than sensory, involving constraint understanding and algorithmic reasoning. Benchmarks such as GeoBenchX [8] systematically evaluate

this capability across tabular, vector, and raster datasets, highlighting how structured data perception enables multi-step analytical workflows.

### 2.1.3. 3.1.3 Vector Geospatial Data

Vector-based geospatial data, including shapefiles, GeoJSON, and spatial graphs, introduces a modality where perception centers on geometry, topology, and spatial relationships. Agents must reason over spatial joins, containment, distance, and direction, often through function calls or generated code. Systems such as ShapefileGPT [22] and GeoJSON [23] Agents explicitly frame perception as interpreting structured vector representations and mapping high-level instructions to low-level geometric operations. The GeoAgent framework for interactive geospatial intelligence further integrates vector data with tabular records and text, requiring agents to coordinate SQL querying and spatial reasoning. MapAgent [24] extends this modality by embedding vector reasoning within map-centric workflows, where structured spatial outputs from APIs must be interpreted in sequence.

### 2.1.4. 3.1.4 Raster and Image Data

Remote sensing imagery constitutes one of the most perceptually demanding modalities for agentic AI systems. Unlike text or structured data, image-based perception requires extracting semantic information from high-dimensional visual inputs, often through specialized vision encoders invoked as tools. Systems such as RS-Agent [11] treat perception as recognizing scenes, objects, and land-use patterns from satellite imagery, enabling tasks such as object detection and visual question answering. ThinkGeo [10] and GeoLLM-Engine [25] advance this setting by embedding perception within tool-augmented pipelines, where agents iteratively invoke detection, segmentation, or change analysis models over multi-temporal imagery. VICoT-Agent [26] further interleaves vision and reasoning by converting visual outputs into textual descriptions that guide subsequent planning steps. At a larger scale, Earth-agent [27] operates over the Earth-Bench benchmark, where perception spans multispectral imagery and derived products, supporting complex Earth observation workflows. In these environments, perception is active and iterative rather than passive, closely coupling visual interpretation with planning and tool orchestration.

### 2.1.5. 3.1.5 Time-Series Signals

Time-series signal-based perception introduces temporal structure as a first-class component of perception. Agents must interpret sequences of events, locations, and actions over time, often to model behavior or support long-horizon decision-making. In LLMob [28], perception involves understanding personal mobility trajectories annotated with semantic location categories, enabling agents to generate realistic activity sequences. PReP [29] extends this modality to navigation tasks, where agents perceive street-view images, road networks, and landmarks to guide goal-directed movement across urban environments.

### 2.1.6. 3.1.6 Map-Centric Multimodal Perception

Map-centric environments combine multiple modalities, including map imagery, spatial metadata, points of interest, and API responses. Perception in these systems involves grounding language and vision in a shared spatial context while interacting with external services. MapAgent [24] exemplifies this setting by decomposing user queries into subgoals that trigger API calls and visual grounding on maps. Similarly, MapBot [30] emphasizes interactive perception, where users provide GeoTIFF imagery and the agent performs segmentation and annotation. GeoLLM-Squad [9] adopts a multi-agent approach, where specialized agents perceive different aspects of the environment, such as maps, satellite imagery, or tabular data, and coordinate to complete complex workflows.

## 2.2. *Agentic Capabilities*

Compared to single-pass LLM-based approaches, agentic systems for ST data explicitly incorporate grounding mechanisms and iterative control loops, resulting in more reliable, interpretable, and

verifiable outcomes [4]. In this section, we outline the core technical components commonly adopted by agentic AI systems using ST data and describe how these components interact in practice. Table 2 summarizes common agentic AI architectures, their capabilities, and representative systems reviewed in this survey.

**Table 2.** Core capabilities of agentic AI systems for spatio-temporal data. Each capability represents a fundamental mechanism through which agents perceive, reason, and act in complex spatial and temporal environments.

Agentic Capability	Description	Representative Systems
Planning & Reasoning	Decomposes complex spatial or temporal objectives into ordered sub-tasks, enabling multi-step analysis, navigation, optimization, and workflow execution.	GeoAgent [16], ThinkGeo [10], MapAgent [24], GeoFlow [13], AgentAD [15], PReP [29]
Knowledge Retrieval	Grounds agent decisions in external, domain-specific sources such as geospatial codebases, spatial databases, planning documents, or model metadata to reduce hallucination and enforce spatial and semantic constraints.	GeoAgent [16], GeoCogent [17], GeoColab [18], GeoQA [20], PlanGPT [19], REMSA [21], GeoEvolue [31]
Memory & State Tracking	Maintains continuity across multi-step workflows or long-horizon tasks by tracking intermediate results, execution states, user interactions, or historical context.	GeoColab [18], ShapefileGPT [22], GeoFlow [13], GeoQA [20], LLMob [28], PReP [29]
Tool Use	Enables agents to invoke external systems such as GIS libraries, SQL engines, vision models, map services, or optimization solvers to act on real-world spatio-temporal data.	GeoAgent [16], RS-Agent [11], ThinkGeo [10], GeoLLM-Engine [25], MapBot [30], VICoT-Agent [26], ST-text-to-sql[6]

### 2.2.1. 3.2.1 Planning & Reasoning

In agentic AI systems, *planning* refers to decomposing high-level goals into executable, analytical, or computational steps, while *reasoning* denotes interpreting intermediate observations and iteratively refining actions. Compared with single-pass LLM pipelines, agentic AI systems embed explicit control loops (e.g., plan-act-observe) that support long-horizon workflows, error recovery, and auditable tool interaction.

**(i) LLM-Driven Task Decomposition** A common design uses an LLM as the planner that maps user intent into structured sub-tasks. Several systems adopt explicit decomposition patterns to improve reliability and debuggability. GeoCogent [17] organizes geospatial code generation into staged reasoning (requirement parsing, algorithm design, code synthesis, and debugging), using LLM outputs at each stage to condition the next. MapAgent [24] similarly employs a hierarchical planner (LLM or VLLM) that converts a query into a sequence of subgoals aligned with a fixed inventory of modules, producing a structured execution plan prior to tool invocation. Planner-executor designs also appear in GeoJSON Agents [23] and ShapefileGPT [22], where an LLM planner decomposes geospatial tasks into ordered subtasks that are executed via either code generation or function calling. In interactive geospatial intelligence, GeoAgent [7] performs query decomposition into multiple sub-queries spanning SQL generation, geometric reasoning, and visualization, enabling multi-step spatial reasoning over heterogeneous spatial and textual sources.

**(ii) Search-Based and Algorithmic Planning** Beyond pure LLM planning, some systems incorporate search or algorithmic controllers to guide reasoning. GeoAgent [16] integrates Monte Carlo Tree Search (MCTS) as an explicit feedback-driven planning mechanism for geospatial code generation, where candidate action sequences are explored and refined based on execution outcomes. For

optimization-centric tasks, AgentAD [15] frames planning around constrained algorithm design for satellite scheduling, emphasizing symbolic reasoning over objective functions and constraints (with LLMs primarily supporting interpretation and algorithm drafting). GeoEvolve [31] further shifts planning toward automated discovery, where iterative refinement is driven by evaluation feedback in geospatial modeling (e.g., interpolation and uncertainty quantification), with LLMs contributing to hypothesis generation and search guidance.

**(iii) Tool-Interleaved and Execution-Aware Reasoning** A complementary paradigm interleaves reasoning with tool execution so that intermediate observations revise subsequent steps. ReAct-style [32] loops are central in benchmarked remote sensing settings. ThinkGeo [10] evaluates tool-augmented agents that alternate between reasoning and calls to perception/operation tools (e.g., detection, change analysis, solvers), emphasizing grounded spatial reasoning under tool feedback. Earth-agent [27] adopts a ReAct-inspired controller for long-horizon Earth observation workflows, iteratively interpreting tool outputs and updating actions to complete multi-step tasks. In database-centric settings, ST Text-to-SQL pipelines [6] implement execution-aware reasoning: agents retrieve schema context, generate SQL, execute queries, inspect results, and repair queries when outputs are empty or incorrect. GeoBenchX [8] similarly benchmarks multi-step geospatial problem solving under ReAct-style tool invocation across tabular, vector, and raster data. Retrieval-guided pipelines such as RS-Agent [11] also reflect partial planning: the controller infers task type, selects or retrieves an appropriate solution path, and executes tool chains for remote sensing tasks. GeoLLM-QA [12] further evaluates tool-augmented remote sensing agents, where stepwise reasoning is grounded in interactive tool usage over imagery and platform actions.

**(iv) Multi-Agent Planning and Collaborative Reasoning** For complex ST tasks, several systems distribute planning and reasoning across multiple agents with specialized roles. GeoLLM-Squad [9] employs a multi-agent backend built on AutoGen [33], where agents specializing in vegetation analysis, object detection, or spatial querying collaboratively solve remote sensing workflows. A coordinating agent manages communication and ensures consistency across subtasks. GeoColab [18] adopts a similar collaborative paradigm for geospatial code generation, assigning roles such as planner, developer, and debugger to different LLM agents to improve robustness and reduce individual agent failure.

GeoQA [20] introduces a router agent that classifies user intent and dispatches tasks to specialized agents responsible for textual explanation or analytical processing. Smart-city management systems further extend multi-agent planning by combining agents for spatial analysis, document reasoning, and API interaction, enabling coordinated decision support across heterogeneous urban data sources. PANGAEA GPT [34] envisions a hierarchical multi-agent design, with a supervisor agent orchestrating domain-specific agents in earth science applications.

**(v) Sequential and Reflective Reasoning in Mobility and Navigation** Agentic planning and reasoning also arise in trajectory and navigation settings, where decision-making unfolds over time. PReP [29] couples perception with *reflection with memory* and planning for goal-directed city navigation, using iterative reasoning to revise actions based on visual observations and accumulated state. In mobility generation, LLMob [28] models activity trajectories as structured temporal sequences, requiring sequential reasoning over mobility patterns and contextual cues rather than explicit plan graphs. In remote sensing analysis, VICoT-Agent [26] supports multi-step reasoning by interleaving vision-language inference with intermediate tool outputs embedded into the reasoning trace, enabling interpretable and scalable analysis over high-resolution imagery.

### 2.2.2. 3.2.2 Knowledge Retrieval

A key advance from single-pass LLMs to agentic systems in ST domains is the integration of explicit knowledge retrieval into the reasoning loop, enabling agents to ground decisions in domain-specific, non-parametric evidence rather than relying solely on parametric model knowledge. We next outline these retrieval variants and discuss how retrieved knowledge is used to support grounding and verification.

(i) **Explicit RAG-Based Knowledge Retrieval** Among existing systems, retrieval-augmented generation (RAG) is the dominant paradigm, in which agents condition planning and generation on retrieved external evidence. Code-centric agents such as GeoAgent [16] and GeoCogent [17] retrieve geospatial library documentation, API references, and executable templates from curated corpora (e.g., GeoCode) to ground code generation and debugging. In collaborative coding settings, GeoColab [18] extends this approach by retrieving semantically similar tasks and prior solution structures to inform multi-agent planning and division of labor. In policy- and governance-oriented applications, PlanGPT [19] employs PlanRAG to retrieve authoritative urban planning regulations and reports, constraining agent outputs within policy-compliant boundaries. These explicit RAG-based designs tightly couple retrieval with iterative plan-act-observe control loops, substantially reducing hallucination and enforcing domain constraints.

(ii) **Structured and Task-Scoped Retrieval** Beyond explicit RAG over unstructured text or code corpora, several systems employ structured or task-scoped retrieval mechanisms that partially ground agent reasoning. For data-centric geospatial reasoning, GeoQA [20] and ST Text-to-SQL pipelines [6] retrieve database schemas, spatial attributes, and metadata to align natural-language intent with executable spatial queries, while interactive GeoAgent systems for geospatial intelligence [7] further combine semantic document filtering with logic-oriented retrieval to support multi-step spatial reasoning. Retrieval is also applied to non-textual and meta-level knowledge sources: REMSA [21] retrieves structured metadata from foundation model repositories to ground remote sensing foundation model selection, and GeoEvolve [31] explicitly introduces GeoKnowRAG, a knowledge-assisted retrieval framework that leverages prior model performance, dataset characteristics, and uncertainty estimates to guide automated geospatial model discovery.

### 2.2.3. 3.2.3 Memory & State Tracking

Memory enables agentic AI systems to maintain consistency, continuity, and adaptation across extended ST workflows. It supports persistent task context, coordination of intermediate results, and reflection over past actions, thereby improving robustness and interpretability. Existing work in agentic AI for ST data explores memory primarily at the level of task-scoped state, episodic interaction history, and temporal sequence modeling, rather than long-term autonomous learning.

(i) **Episodic and Interaction Memory** Episodic memory captures specific interactions, observations, and actions during task execution, enabling agents to reason over what has already been attempted and to adjust future decisions accordingly. In geospatial agent systems, episodic memory is most commonly realized as session- or task-level state tracking rather than persistent cross-task storage. For example, GeoCogent [17] maintains an explicit record of intermediate artifacts across its multi-stage pipeline, storing parsed requirements, algorithm designs, generated code, and debugging feedback to condition subsequent reasoning steps. Similarly, GeoColab [18] employs shared working memory across role-specialized agents, preserving task specifications and intermediate code states to ensure consistency and coordinated refinement during collaborative geospatial code generation.

Episodic memory also supports interactive and navigation-oriented settings. PReP [29] introduces a reflection-with-memory mechanism for goal-directed city navigation, where past observations and actions are stored and revisited to revise navigation strategies over time. In mobility modeling, LLMob [28] implicitly encodes episodic structure through temporal activity sequences, allowing the agent to condition future trajectory generation on prior states. While these approaches demonstrate the benefits of episodic memory for continuity and adaptation, they typically limit persistence to a single task or episode rather than maintaining long-term spatial histories.

(ii) **Task-Scoped and Execution-State Memory** A more prevalent form of memory in ST agentic systems is task-scoped or execution-state memory, which tracks progress through multi-step workflows. ShapefileGPT [22] and GeoJSON Agents [23] explicitly store decomposed subtasks, execution status, and intermediate outputs, enabling planners to manage complex sequences of geometric operations without redundancy. GeoQA [20] similarly preserves structured representations of user intent and routing decisions in intermediate semantic states that guide downstream agents. Other systems

maintain state implicitly through control structures rather than explicit memory modules. GeoAgent [16] tracks code revisions and execution feedback within an MCTS-based planning loop, while MapAgent [24] and GeoFlow [13] preserve execution state through structured plans or workflow graphs.

#### 2.2.4. 3.2.4 Tool Use

In ST domains, agentic AI systems rely on external tools to translate high-level reasoning into verifiable spatial operations. Effective geospatial agents invoke GIS libraries, databases, vision models, and map services to execute spatial queries, perform geometric computations, analyze remote sensing imagery, and validate results.

A dominant pattern is the use of GIS programming toolkits. Code-centric agents such as GeoAgent [16], GeoCogent [17], and GeoColab [18] expose Python environments that interface with standard geospatial libraries (e.g., GeoPandas, Rasterio, Shapely, GDAL), enabling iterative execution, debugging, and refinement of workflows such as spatial joins and raster–vector transformations. Function-calling designs, as in ShapefileGPT [22] and GeoJSON Agents [23], further abstract geometric operations into structured spatial tools, a pattern also adopted in ST Text-to-SQL systems [6] and GeoQA [20].

Tool use is equally central in remote sensing and Earth observation. Systems such as RS-Agent [11], ThinkGeo [10], GeoLLM-Engine [25], and Earth-agent [27] provide access to vision models for detection, segmentation, and change analysis, while MapBot [30] and VICoT-Agent [26] integrate models such as SAM [35] and DINOv2 [36] for interactive and interpretable visual reasoning. Map services and external APIs support navigation and planning tasks, with systems such as MapAgent [24] and smart-city agents [37] grounding agent decisions in routing services, spatial context APIs, regulatory documents, and live urban data.

### 2.3. Application Landscape

This section reviews practical applications of agentic AI systems across ST domains. Table 3 summarizes the major application categories, agent roles, and representative systems. Below, we discuss each category in turn.

**Table 3.** Application landscape of agentic AI systems for spatio-temporal data, grouped by functional role and decision scope. Each category reflects a distinct mode of interaction between agents, spatial data, and decision-making objectives.

Application Category	Agent Role	Representative Systems
Geospatial Reasoning & Question Answering	Spatial Analyst GIS Query Agent	GeoAgent [7], GeoQA [20], MapAgent [24], Spatio-Temporal NL-to-SQL [6], GeoBenchX [8]
Programmatic GIS & Code Automation	GIS Programmer Workflow Executor	GeoAgent [16], GeoCogent [17], GeoColab [18], ShapefileGPT [22], GeoJSON [23]
Remote Sensing & Earth Observation	Earth Observation Analyst Vision Reasoning Agent	RS-Agent [11], ThinkGeo [10], GeoLLM-Engine [25], GeoFlow [13], VICoT-Agent [26], Earth-agent [27], GeoLLM-Squad [9]
Planning, Optimization & Decision Support	Planning Agent Optimization Assistant	AgentAD [15], GeoEvolve [31], REMSA [21], [37], PlanGPT [19]
Human-Centric Mobility & Urban Interaction	Navigation Agent Urban Behavior Simulator	LLMob [28], PReP [29], MapBot [30]

#### 2.3.1. 3.3.1 Geospatial Reasoning and Question Answering(QA)

A large class of agentic systems focuses on geospatial reasoning and QA, where agents act as spatial analysts that translate natural-language queries into structured spatial operations. These systems support tasks ranging from simple retrieval (e.g., locating nearby POIs) to complex logical reasoning involving spatial predicates, joins, and temporal constraints.

Early systems such as GeoAgent [7] and GeoQA [20] enable agents to parse natural-language questions, retrieve relevant GIS layers or documents, and generate executable SQL or spatial queries. GeoAgent further supports multi-step reasoning by decomposing queries into sub-questions and coordinating SQL execution with visualization generation. MapAgent [24] extends this paradigm by incorporating map APIs and visual map context, enabling agents to reason jointly over textual descriptions, structured map responses, and rendered map imagery. Benchmark efforts such as GeoBenchX [8] formalize these tasks, evaluating agents on increasingly complex spatial reasoning scenarios, including neighborhood inference, proximity reasoning, and multimodal map understanding.

### 2.3.2. 3.3.2 Programmatic GIS and Code Automation

Another prominent application category centers on programmatic GIS and code automation, where agents function as GIS programmers or workflow executors. These systems translate high-level user intent into executable geospatial code, automate repetitive GIS operations, and debug or refine spatial workflows.

GeoAgent [16] and GeoCogent [17] exemplify this direction by combining natural-language understanding with retrieval-augmented code generation. GeoAgent retrieves relevant API documentation and example scripts from the GeoCode corpus and uses Monte Carlo Tree Search to iteratively refine Python programs for tasks such as change detection or spatial analysis. GeoColab [18] extends code generation into a collaborative, multi-role setting, coordinating requirement parsing, algorithm design, and implementation across specialized agents. ShapefileGPT [22] and GeoJSON Agents [23] focus on structured vector data manipulation, enabling agents to perform geometric operations, spatial queries, and file transformations via function calling or code synthesis.

### 2.3.3. 3.3.3 Remote Sensing(RS) and Earth Observation(EO)

Agentic AI has also been widely adopted in RS and EO, where agents operate over large-scale raster imagery and multimodal sensor data. In this setting, agents act as vision reasoning assistants that orchestrate perception models, spatial analysis tools, and iterative reasoning loops.

Systems such as RS-Agent [11] and ThinkGeo [10] automate tasks including object detection, land-use classification, damage assessment, and environmental monitoring across diverse datasets. GeoLLM-Engine [25] provides a realistic environment in which agents interact with hundreds of vision and GIS tools to solve long-horizon EO tasks, such as land-cover classification over millions of satellite images. GeoFlow [13] frames EO pipelines as workflow graphs, where a meta-agent generates execution plans and delegates subtasks such as object detection or classification to tool-augmented agents. VICoT-Agent [26] interleaves vision and language reasoning by generating intermediate visual descriptions and grounding them through tool-based verification. More recent systems such as Earth-agent [27] and GeoLLM-Squad [9] further emphasize scalable orchestration, coordinating multiple agents and tools to handle complex, multi-stage remote sensing workflows in urban analysis, agriculture, and disaster response.

### 2.3.4. 3.3.4 Planning, Optimization, and Decision Support

Beyond analysis, several agentic systems target planning, optimization, and decision support problems that require long-horizon reasoning under constraints. In these applications, agents act as planning assistants or optimization solvers rather than purely analytical tools. AgentAD [15] addresses satellite scheduling by decomposing the problem into algorithm design, implementation, optimization, and validation stages coordinated by specialized agents. GeoEvolve [31] focuses on automated geospatial model discovery, using agentic search to evaluate spatial interpolation and uncertainty quantification methods. REMSA [21] tackles foundation model selection in remote sensing, helping users navigate trade-offs among model capabilities and resource constraints. PlanGPT [19] and smart-city agents [37] extend decision support to urban planning by grounding recommendations in retrieved regulations, planning documents, and structured urban data.

### 2.3.5. 3.3.5 Human-Centric Mobility and Urban Interaction

Finally, emerging applications emphasize human-centric mobility and urban interaction, where agents model, simulate, or assist human behavior over space and time. These systems often operate in interactive or longitudinal settings, making temporal reasoning and state tracking particularly important. LLMob [28] treats large language models as simulated urban residents, generating realistic mobility trajectories conditioned on personal activity patterns and spatial constraints. PReP [29] studies goal-directed city navigation without explicit instructions, integrating visual perception from street-view imagery with reflective planning mechanisms. MapBot [30] enables interactive geospatial analysis by allowing users to guide segmentation and annotation through clicks on map imagery.

## 3. Future Directions & Opportunities

This survey examined recent advances in agentic AI for spatio-temporal intelligence, spanning perception modalities, agentic capabilities, and application domains such as geospatial analysis, remote sensing, urban planning, and mobility. We showed how agentic designs extend foundation models beyond single-pass inference through iterative reasoning, tool use, and long-horizon interaction with spatial and temporal environments. Despite rapid progress, challenges remain:

### 3.0.1. Explainability and Transparent Reasoning

As agents perform multi-step reasoning and execution, explainability becomes critical for trust and debugging. Existing systems often expose plans or tool calls, but explanations remain informal and difficult to audit. In spatio-temporal contexts, explanations must clarify not only *what* was decided, but also *where* and *when*. Approaches that interleave reasoning traces with spatial evidence, such as VICoT-Agent [26] and GeoAgent [16], point toward more transparent and inspectable agent behavior.

### 3.0.2. Generalization of Spatial Foundation Models

A key future direction for agentic AI in ST domains is the development and integration of geospatial foundation models [38]. While current agents often rely on general-purpose language or vision-language models, future systems will benefit from models explicitly trained to capture spatial relations, temporal dynamics, and geographic constraints across diverse regions and sensing conditions. Benchmarks such as GeoLLM-Engine [25] and GeoBenchX [8] provide initial evaluation frameworks, but broader assessments across cities, sensors, and time spans are still needed. Advancing scalable training, continual adaptation, and tighter coupling between geospatial foundation models and agentic reasoning loops remains an open challenge.

## References

1. Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology* **2024**, *15*, 1–45.
2. Acharya, D.B.; Kuppan, K.; Divya, B. Agentic ai: Autonomous intelligence for complex goals—a comprehensive survey. *IEEE Access* **2025**.
3. Gridach, M.; Nanavati, J.; Abidine, K.Z.E.; Mendes, L.; Mack, C. Agentic ai for scientific discovery: A survey of progress, challenges, and future directions. *arXiv preprint arXiv:2503.08979* **2025**.
4. Sapkota, R.; Roumeliotis, K.I.; Karkee, M. Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges. *arXiv preprint arXiv:2505.10468* **2025**.
5. Wang, S.; Cao, J.; Philip, S.Y. Deep learning for spatio-temporal data mining: A survey. *IEEE transactions on knowledge and data engineering* **2020**, *34*, 3681–3700.
6. Redd, M.; Zhe, T.; Wang, D. From Queries to Insights: Agentic LLM Pipelines for Spatio-Temporal Text-to-SQL. In Proceedings of the Proceedings of the 1st ACM SIGSPATIAL International Workshop on Generative and Agentic AI for Multi-Modality Space-Time Intelligence, 2025, pp. 6–14.
7. Hu, J.; Sun, L.; Liu, X. GeoAgent: An Agentic AI Framework for Spatial Query Understanding and Interactive Geospatial Intelligence. In Proceedings of the Proceedings of the 1st ACM SIGSPATIAL International Workshop on Generative and Agentic AI for Multi-Modality Space-Time Intelligence, 2025, pp. 54–60.

8. Krechetova, V.; Kochedykov, D. GeoBenchX: Benchmarking LLMs in Agent Solving Multistep Geospatial Tasks. In Proceedings of the Proceedings of the 1st ACM SIGSPATIAL International Workshop on Generative and Agentic AI for Multi-Modality Space-Time Intelligence, 2025, pp. 27–35.
9. Lee, C.; Paramanayakam, V.; Karatzas, A.; Jian, Y.; Fore, M.; Liao, H.; Yu, F.; Li, R.; Anagnostopoulos, I.; Stamoulis, D. Multi-Agent Geospatial Copilots for Remote Sensing Workflows. *arXiv preprint arXiv:2501.16254* **2025**.
10. Shabbir, A.; Munir, M.A.; Dudhane, A.; Sheikh, M.U.; Khan, M.H.; Fraccaro, P.; Moreno, J.B.; Khan, F.S.; Khan, S. ThinkGeo: Evaluating Tool-Augmented Agents for Remote Sensing Tasks. *arXiv preprint arXiv:2505.23752* **2025**.
11. Xu, W.; Yu, Z.; Mu, B.; Wei, Z.; Zhang, Y.; Li, G.; Peng, M. RS-Agent: Automating Remote Sensing Tasks through Intelligent Agent. *arXiv preprint arXiv:2406.07089* **2024**.
12. Singh, S.; Fore, M.; Stamoulis, D. Evaluating tool-augmented agents in remote sensing platforms. *arXiv preprint arXiv:2405.00709* **2024**.
13. Bhattaram, A.; Chung, J.; Chung, S.; Gupta, R.; Ramamoorthy, J.; Gullapalli, K.; Marculescu, D.; Stamoulis, D. GeoFlow: Agentic Workflow Automation for Geospatial Tasks. In Proceedings of the Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems, 2025, pp. 1150–1153.
14. Chen, Y.; Wang, W.; Lobry, S.; Kurtz, C. An llm agent for automatic geospatial data analysis. *arXiv preprint arXiv:2410.18792* **2024**.
15. Chen, J.; Chen, Y.; Pham, D.T.; Song, Y.; Wu, J.; Xing, L.; Chen, Y. A Large Language Model-based Multi-Agent Framework to Autonomously Design Algorithms for Earth Observation Satellite Scheduling Problem. *Engineering* **2025**.
16. Huang, C.; Chen, S.; Li, Z.; Qu, J.; Xiao, Y.; Liu, J.; Chen, Z. Geoagent: To empower llms using geospatial tools for address standardization. In Proceedings of the Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 6048–6063.
17. Hou, S.; Jiao, H.; Liang, J.; Shen, Z.; Zhao, A.; Wu, H. GeoCogent: an LLM-based agent for geospatial code generation. *International Journal of Geographical Information Science* **2025**, pp. 1–34.
18. Wu, H.; Jiao, H.; Hou, S.; Liang, J.; Shen, Z.; Zhao, A.; Qing, Y.; Jin, F.; Guan, X.; Gui, Z. GeoColab: an LLM-based multi-agent collaborative framework for geospatial code generation. *International Journal of Digital Earth* **2025**, *18*, 2569405.
19. Zhu, H.; Chen, G.; Zhang, W. PlanGPT: Enhancing urban planning with a tailored agent framework. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), 2025, pp. 764–783.
20. Feng, Y.; Zhang, P.; Xiao, G.; Ding, L.; Meng, L. Towards a Barrier-free GeoQA Portal: Natural Language Interaction with Geospatial Data Using Multi-Agent LLMs and Semantic Search. *arXiv preprint arXiv:2503.14251* **2025**.
21. Chen, B.; Bök, T.E.; Rasti, B.; Markl, V.; Demir, B. REMSA: An LLM Agent for Foundation Model Selection in Remote Sensing. *arXiv preprint arXiv:2511.17442* **2025**.
22. Lin, Q.; Hu, R.; Li, H.; Wu, S.; Li, Y.; Fang, K.; Feng, H.; Du, Z.; Xu, L. ShapefileGPT: A Multi-Agent Large Language Model Framework for Automated Shapefile Processing (arXiv: 2410.12376). *arXiv*, 2024.
23. Luo, Q.; Lin, Q.; Xu, L.; Wu, S.; Mao, R.; Wang, C.; Feng, H.; Huang, B.; Du, Z. GeoJSON Agents: A Multi-Agent LLM Architecture for Geospatial Analysis-Function Calling vs Code Generation. *arXiv preprint arXiv:2509.08863* **2025**.
24. Hasan, M.H.; Dihan, M.L.; Hashem, T.; Ali, M.E.; Parvez, M.R. MapAgent: A Hierarchical Agent for Geospatial Reasoning with Dynamic Map Tool Integration. *arXiv preprint arXiv:2509.05933* **2025**.
25. Singh, S.; Fore, M.; Stamoulis, D. Geollm-engine: A realistic environment for building geospatial copilots. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 585–594.
26. Wang, C.; Luo, Z.; Liu, R.; Ran, C.; Fan, S.; Chen, X.; He, C. VICoT-Agent: A Vision-Interleaved Chain-of-Thought Framework for Interpretable Multimodal Reasoning and Scalable Remote Sensing Analysis. *arXiv preprint arXiv:2511.20085* **2025**.
27. Feng, P.; Lv, Z.; Ye, J.; Wang, X.; Huo, X.; Yu, J.; Xu, W.; Zhang, W.; Bai, L.; He, C.; et al. Earth-agent: Unlocking the full landscape of earth observation with agents. *arXiv preprint arXiv:2509.23141* **2025**.
28. Wang, J.; Jiang, R.; Yang, C.; Wu, Z.; Onizuka, M.; Shibusaki, R.; Koshizuka, N.; Xiao, C. Large language models as urban residents: An llm agent framework for personal mobility generation. *Advances in Neural Information Processing Systems* **2024**, *37*, 124547–124574.

29. Zeng, Q.; Yang, Q.; Dong, S.; Du, H.; Zheng, L.; Xu, F.; Li, Y. Perceive, reflect, and plan: Designing llm agent for goal-directed city navigation without instructions. *arXiv preprint arXiv:2408.04168* **2024**.
30. Weiss, M.; Rahaman, N.; Pal, C. MapBot: A Multi-Modal Agent for Geospatial Analysis. In Proceedings of the Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, 2025, pp. 3059–3061.
31. Luo, P.; Lou, X.; Zheng, Y.; Zheng, Z.; Ermon, S. GeoEvolve: Automating Geospatial Model Discovery via Multi-Agent Large Language Models. *arXiv preprint arXiv:2509.21593* **2025**.
32. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.R.; Cao, Y. React: Synergizing reasoning and acting in language models. In Proceedings of the The eleventh international conference on learning representations, 2022.
33. Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In Proceedings of the First Conference on Language Modeling, 2024.
34. Pantiukhin, D.; Shapkin, B.; Kuznetsov, I.; Jost, A.A.; Koldunov, N. Accelerating earth science discovery via multi-agent LLM systems. *Frontiers in Artificial Intelligence* **2025**, *8*, 1674927.
35. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment anything. In Proceedings of the Proceedings of the IEEE/CVF international conference on computer vision, 2023, pp. 4015–4026.
36. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* **2023**.
37. Kalyuzhnaya, A.; Mityagin, S.; Lutsenko, E.; Getmanov, A.; Aksenkin, Y.; Fatkhiev, K.; Fedorin, K.; Nikitin, N.O.; Chichkova, N.; Vorona, V.; et al. LLM Agents for Smart City Management: Enhancing Decision Support Through Multi-Agent AI Systems. *Smart Cities (2624-6511)* **2025**, *8*.
38. Hashemi, M.; Zufle, A. From Points to Places: Towards Human Mobility-Driven Spatiotemporal Foundation Models via Understanding Places. *arXiv preprint arXiv:2506.14570* **2025**.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.