GS-QA2: A Benchmark for Question Answering over Raster–Vector Data [Experiment]

Zhuocheng Shang
University of California, Riverside
Riverside, USA
zshan011@ucr.edu

Shahd Elmahallawy
University of California, Riverside
Riverside, USA
selma008@ucr.edu

Zabir Al Nazi
University of California, Riverside
Riverside, USA
znazi002@ucr.edu

Vagelis Hristidis
University of California, Riverside
Riverside, USA
vagelis@cs.ucr.edu

Ahmed Eldawy
University of California, Riverside
Riverside, USA
eldawy@ucr.edu

Abstract

Geospatial question answering has recently gained attention as large language models (LLMs) enable natural language interaction with geographic data. Existing benchmarks and systems focus mostly on vector data such as points, lines, and polygons, yet many real geospatial tasks also require reasoning over raster data that represents continuous surfaces such as elevation. It therefore remains unclear whether current LLM based systems can reason over heterogeneous geospatial data. We present a new benchmark for geospatial question answering over both vector and raster data, which integrates OpenStreetMap vector features with a U.S. Digital Elevation Model (DEM) to enable controlled evaluation of combined raster and vector reasoning. The benchmark introduces new question templates spanning raster-only terrain queries, vector queries augmented with terrain constraints, and new raster-vector reasoning tasks. We evaluate representative geospatial QA paradigms, including Text2SQL, retrieval augmented generation (RAG), a multi stage SQL generation framework, and a multi stage agent that writes Python code for geospatial tasks. The experiment results show that these systems answer simple point elevation queries but lose accuracy sharply on terrain derivatives and combined raster-vector tasks, revealing substantial limitations in LLM based geospatial reasoning and establishing a reproducible target for future research.

CCS Concepts

• **Information systems → Question answering**.

Keywords

Geospatial Question Answering, Large Language Models, Spatial Reasoning, Benchmarking, Raster and Vector Data

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Zhuocheng Shang, Shahd Elmahallawy, Zabir Al Nazi, Vagelis Hristidis, and Ahmed Eldawy. 2018. GS-QA2: A Benchmark for Question Answering over Raster–Vector Data [Experiment]. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Recent advances in large language models (LLMs) have renewed interest in natural language querying of spatial datasets. Rather than structured queries, users pose *geospatial questions* that can be translated into SQL queries, GIS workflows, or custom geospatial code. For example, answering *"Find a park within 100 meters of Los Angeles City Hall"* requires grounding named entities, interpreting spatial predicates, and executing the resulting query over spatial data. Recent systems have explored this problem through retrieval augmented generation (RAG) over vectorized records, text to SQL (Text2SQL) generation over spatial databases [13], and LLM powered GIS agents [1, 21].

To measure progress, the community has built benchmarks for these systems, but they cover only part of the geospatial landscape. Most benchmarks for natural language queries over spatial data focus almost exclusively on *vector* data, such as points, lines, and polygons [2, 3, 7, 12, 13, 17]. A separate line of work addresses raster inputs in the form of satellite and aerial imagery, but from a different angle, targeting visual perception tasks such as scene classification, object detection, and segmentation [8, 22] rather than structured reasoning over raster values.

As a result, neither research direction evaluates whether systems can answer questions that require reasoning over raster data or the interaction between raster and vector datasets. Unlike vector datasets, which represent discrete geographic entities with names and attributes, raster datasets represent space as grids of cells storing continuous measurements such as elevation, temperature, or vegetation indices. For example, a digital elevation model (DEM) represents terrain as a continuous elevation surface rather than a collection of named objects. Many practical tasks combine such surfaces with vector features such as roads, parks, rivers, and administrative regions. Answering *"Which roads have the steepest slope in this region?"* requires combining road and elevation data to estimate the slope of each road, while answering *"Which parks*

are located above the regional average elevation?" requires aggregating elevation across multiple raster tiles and comparing each park against the resulting average. Such questions require reasoning over continuous raster values and their interaction with vector features, which makes raster aware geospatial question answering fundamentally different from vector-only querying. Existing benchmarks largely overlook these capabilities, creating a significant gap in the evaluation of LLM based geospatial question answering systems.

This paper introduces GS-QA2, a benchmark for question answering over both raster and vector geospatial data. GS-QA2 builds on GS-QA [13], a vector-only QA benchmark, by incorporating a raster dataset derived from the U.S. DEM [5] together with new baselines and evaluation measures. Specifically, GS-QA2 adds 25 question templates spanning raster-only terrain queries, vector queries augmented with elevation or slope conditions, and joint raster-vector questions. The templates systematically cover the four map algebra operation types. Local operations retrieve raster values at specific locations. Focal operations compute neighborhood based properties such as slope. Zonal operations aggregate raster measurements over spatial regions. Global operations search across the raster surface to identify locations that satisfy a specified condition.

Starting from each template, we instantiate questions using real geographic entities including points, roads, parks, and administrative regions, and generate corresponding executable PostGIS SQL queries to obtain ground truth answers. The resulting benchmark contains 500 raster related question answer pairs that complement the 2,800 vector based pairs from GS-QA [13]. We evaluate answers using metrics tailored to each template type, including token level F1 for entity names, distance and angular error for spatial quantities, and absolute or relative error thresholds for numeric outputs.

This paper offers an unbiased evaluation of existing spatial question answering techniques. We use GS-QA2 to evaluate four representative systems, including Text2SQL, retrieval augmented generation (RAG) [13], CHESS [15], and GIS Copilot [1], using Gemini-2.5-Flash as the underlying LLM. Our evaluation examines how well these systems reason about complex geospatial queries and how their performance differs between database driven approaches such as Text2SQL and CHESS and non database approaches such as RAG and GIS Copilot.

To summarize, our contributions are as follows.
(1) We introduce GS-QA2, a benchmark for geospatial question answering over both vector and raster data. GS-QA2 extends GS-QA [13] with a large scale DEM dataset and 25 new templates spanning raster-only and raster-vector reasoning tasks, organized according to increasing levels of raster reasoning complexity.
(2) We provide a systematic evaluation framework for geospatial question answering. The benchmark covers local, focal, zonal, and global raster operations, supports diverse answer types, and includes executable PostGIS SQL ground truth together with evaluation metrics tailored to geospatial outputs.
(3) We conduct a comprehensive empirical study of representative geospatial QA paradigms, including Text2SQL, SQL generation with multiple stages, retrieval augmented generation, and agent based GIS workflows. Our results show that current systems handle vector queries at a reasonable accuracy level but struggle with raster analysis and raster-vector reasoning, particularly for complex operations.

The rest of this paper is organized as follows. Section 2 presents the GS-QA2 benchmark design, including datasets, question templates, and answer generation. Section 3 describes the evaluated systems and adaptations to GS-QA2. Section 4 presents the experiments and evaluation results followed by a discussion. Section 5 summarizes related work. Section 6 concludes the paper.

2 Benchmark Creation

This section describes our methodology for constructing a benchmark for geospatial question answering over both vector and raster data. Three principles guide how we create the GS-QA2 question and answer pairs. First, the benchmark systematically covers the fundamental vector and raster operations, so that performance can be attributed to specific spatial capabilities rather than to an arbitrary mix of queries. Second, each question is paired with an SQL query that produces it along with the ground truth answer, which makes every answer verifiable and removes the need for manual annotation. Third, generation uses templates, where a template fixes the spatial structure of a question and its parameters are sampled from the database, allowing large numbers of varied questions to be created automatically while keeping the underlying query patterns controlled.

2.1 Reference Database

Our benchmark contains three main sources of data. OpenStreetMap (OSM) datasets provide location data, a Wikipedia dataset adds more information to the locations, and a raster terrain dataset provides the Digital Elevation Model (DEM) [5] for the US. The first two datasets were used in GS-QA [13], while the raster dataset is new to GS-QA2. The OSM vector dataset consists of five OSM tables, chosen for their rich semantic attributes, broad geographic coverage, and widespread use in geospatial applications, covering `POI`, `Park`, `Lake`, `Road`, and `Region` [13]. The Wikipedia dataset is extracted for OSM entities that have a Wikipedia page, and it adds more information about OSM features, such as what year a museum was built. To support reasoning over raster data, we integrate a DEM containing 265,950 tiles covering the contiguous United States [5]. With 256×256 cells per tile, this corresponds to approximately 17.4 billion cells at 30 m resolution and about 32 GB of raster values. We focus on DEM because terrain analysis is one of the most common and interpretable forms of raster geospatial reasoning, enabling queries involving elevation, slope, and other terrain properties. Table 1 summarizes the full benchmark dataset.

2.2 Question Templates Overview

GS-QA2 uses two sets of templates. We retain the 28 vector only templates from GS-QA [13] (V1 through V28) for comparison with prior work, and we introduce 25 new templates for raster and raster-vector reasoning. In total, the benchmark contains 53 template types. Tables 2 through 4 summarize the vector templates, while Tables 5 through 7 summarize the raster and raster vector templates.

The 25 new templates are organized into three groups that require progressively richer forms of raster reasoning. Raster only templates (R1 through R11) query the DEM directly, using POIs or roads only as spatial anchors. Extended raster vector templates

Table 1: Dataset tables summary

Table	Columns / Attributes	Geometries	Records
POI	Geometry, OSM ID, Name, Wikipedia, Address, Leisure, Amenity, Tourism, Emergency, Restaurant attributes	Points	267,612
Park	Geometry, OSM ID, Name, Wikipedia, Address, Leisure	All types	5,997,948
Lake	Geometry, OSM ID, Name, Wikipedia, Address, Water, Waterway	All types	7,988,851
Road	Geometry, OSM ID, Name, Wikipedia, Address, Highway	All types	36,827,649
Region	Geometry, OSM ID, Name, Wikipedia ID, Address, Border type, Administration level	LineStrings, Polygon	39,137
DEM (new)	**Raster tiles, Elevation (m), GiST spatial index on tile convex hull**	**Raster**	**265,950 tiles**

(VR1 through VR8) augment standard vector queries with a single terrain condition or terrain based output. New raster-vector templates (VR9 through VR14) require tighter interaction between vector and raster data, including terrain aggregation, comparison, and ranking.

As the tables show, every template has three columns beyond its ID. The *spatial predicates* column gives the spatial query type, which for raster queries we further organize by raster operation category. The *text* column gives the natural language phrasing with placeholders such as [ANCH_POI] and [DISTANCE], which we fill with real values sampled from the dataset during generation, as described in Section 2.5. The *answer type* column gives the expected output type.

Two principles guide the template design. The first is systematic coverage. We cover both the operations required to answer a question and the types of answers returned by the benchmark. For vector templates, we follow the predicate organization of GS-QA [13], including range, nearest neighbor, direction, towards, and intersects. For raster templates, we follow Tomlin's map algebra framework [16], which classifies raster analysis into four operation types. Local operations read values at specific cells. Focal operations derive values from neighboring cells. Zonal operations aggregate raster values within a region. Global operations search across the raster surface.

The answer types cover both vector and terrain outputs. Vector outputs include entity names, free text, locations, directions, counts, distances, areas, and lengths. Terrain outputs include elevation, slope, aspect, and ruggedness. Some raster vector templates return compound outputs that combine both forms of information, such as a road name together with its slope or a distance together with an elevation.

The second principle is incremental complexity. The three new template groups progress from direct raster queries, to vector queries augmented with a single terrain condition, and finally to joint raster vector queries that require reasoning across both raster and vector data.

Table 2: Vector-only entity name and free text question templates (GS-QA)

ID	Answer Type	Text	Spatial Predicates
V1	Entity Name	Can you suggest [POI CAT] within [DISTANCE] from [ANCH POI]?	Range
V2	Entity Name	Can you suggest [POI NONSPAT] within [DISTANCE] from [ANCH POI]?	Range
V3	Entity Name	Which [POI CAT] is located within [DISTANCE] in the [DIRECTION] of [ANCH POI]?	Range, Direction
V4	Entity Name	Which [POI CAT] can I find within [DISTANCE] from [ANCH POI] towards [ANCH POI]^2?	Range, Towards
V5	Entity Name	What is the nearest [POI NONSPAT] from [ANCH POI]?	Nearest Neighbor
V6	Entity Name	What is the nearest [POI NONSPAT] from [ANCH POI]?	Nearest Neighbor
V7	Free Text	What is the capacity of the nearest [POI CAT] from [ANCH POI]?	Nearest Neighbor
V8	Entity Name	What is the nearest [POI CAT] from [ANCH POI] to [ANCH POI]?	Nearest Neighbor

Table 3: Vector-only location and direction question templates (GS-QA)

ID	Answer Type	Text	Spatial Predicates
V9	Location	What is the closest [POI CAT] [DIRECTION] of [ANCH POI]?	Nearest Neighbor, Direction
V10	Location	What is the closest [POI CAT] from [ANCH POI] towards [ANCH POI]^2?	Nearest Neighbor, Towards
V11	Location	What is the largest [PARK WATW] in [REGION]?	Intersects
V12	Location	What is the longest [ROAD WATW] in [REGION]?	Intersects
V13	Location	Where can I find [POI CAT] within [DISTANCE] from [ANCH POI]?	Range
V14	Location	Where can I find [POI CAT] within [DISTANCE] from [POI NONSPAT]?	Range
V15	Location	Where can I find [POI CAT] located within [DISTANCE] in the [DIRECTION] of [ANCH POI]?	Range, Direction
V16	Location	What location has [POI CAT] within [DISTANCE] from [ANCH POI] towards [ANCH POI]^2?	Range, Towards
V17	Location	Where can I find the nearest [POI CAT] from [ANCH POI]?	Nearest Neighbor
V18	Location	Where can I find the nearest [POI NONSPAT] from [ANCH POI]?	Nearest Neighbor
V19	Location	Where is the closest [POI CAT] [DIRECTION] of [ANCH POI]?	Nearest Neighbor, Direction
V20	Location	Where is the closest [POI CAT] from [ANCH POI] towards [ANCH POI]^2?	Nearest Neighbor, Towards
V21	Direction	In which direction is [POI CAT] located within [DISTANCE] from [ANCH POI]?	Range
V22	Direction	What is the direction towards the closest [POI CAT] from [ANCH POI]?	Nearest Neighbor

2.3 Raster Template Details

Raster operation labels. Tables 5, 6, and 7 give a raster operation label for each template. We label each row by how the template accesses the DEM, not by the shape of its final answer, and the

Table 4: Vector-only numeric-answer question templates (GS-QA)

ID	Answer Type	Text	Spatial Predicates
V23	Count	How many [POI CAT] within [DISTANCE] from [ANCH POI]?	Range
V24	Count	How many [POI CAT] are there in [REGION]?	Intersects
V25	Distance	How far can I find [POI CAT] within [DISTANCE] from [ANCH POI]?	Range
V26	Distance	How far is the closest [POI CAT] from [ANCH POI]?	Nearest Neighbor
V27	Area	What is the total area of all [PARK WATB] in [REGION]?	Intersects
V28	Length	What is the total length of all [ROAD WATW] in [REGION]?	Intersects

Table 5: Raster-only question templates (GS-QA2)

ID	Answer Type	Example Text	Raster Op.	Spatial Predicates
R1	Elevation	What is the elevation at {ANCH_POI}?	Local	Point Lookup
R2	Entity Name	Which is at a higher elevation, {ANCH_POI$_1$} or {ANCH_POI$_2$}?	Local	Point Lookup
R3	Elevation	What is the elevation difference between {ANCH_POI$_1$} and {ANCH_POI$_2$}?	Local	Point Lookup
R4	Yes/No	Is {ANCH_POI} at an elevation above {ELEV_COND} m?	Local	Point Lookup
R5	Slope	How steep is the terrain at {ANCH_POI}?	Focal	Point Lookup
R6	Aspect	What is the aspect of the slope at {ANCH_POI}?	Focal	Point Lookup
R7	Ruggedness	How rugged is the terrain around {ANCH_POI}?	Focal	Point Lookup
R8	Distance	How far is {ANCH_POI} from the nearest terrain above {ELEV_COND} m?	Global	Nearest Elevation
R9	Distance	How far is {ANCH_POI} from the nearest terrain below {ELEV_COND} m?	Global	Nearest Elevation
R10	Slope	What is the average slope along {ROAD_ROUTE}?	Local	Line Sampling, Slope
R11	Entity Name	Which route is steeper, {ROAD_ROUTE$_1$} or {ROAD_ROUTE$_2$}?	Local	Line Sampling, Slope

Table 6: Extended raster-vector question templates (GS-QA2)

ID	Answer Type	Text	Raster Op.	Spatial Predicates
VR1	Entity Name	Can you suggest {POI_CAT} within {DISTANCE} m from {ANCH_POI} with elevation {ELEV_COND}?	Local	Range, Elevation
VR2	Location	Where can I find {POI_CAT} within {DISTANCE} m from {ANCH_POI} with elevation {ELEV_COND}?	Local	Range, Elevation
VR3	Count	How many {POI_CAT} are within {DISTANCE} m from {ANCH_POI} with elevation {ELEV_COND}?	Local	Range, Elevation
VR4	Distance and Elevation	How far are {POI_CAT} from {ANCH_POI} within {DISTANCE} m, and what is their elevation?	Local	Range, Elevation
VR5	Distance and Slope	How far is the nearest {ROAD_TYPE} from {ANCH_POI}, and what is the average slope along that road?	Local	Nearest Neighbor, Line Sampling, Slope
VR6	Name and Slope	What is the longest {ROAD_TYPE} in {REGION}, and what is the slope along it?	Local	Intersects, Line Sampling, Slope
VR7	Name and Elevation	What is the largest {PARK_WATB} in {REGION}, and what is its maximum elevation?	Zonal	Intersects, Elevation
VR8	Area and Slope	What is the total area of all {PARK_WATB} in {REGION}, and what is the average slope of {REGION}?	Zonal	Intersects, Slope

Table 7: New raster-vector question templates (GS-QA2)

ID	Answer Type	Text	Raster Op.	Spatial Predicates
VR9	Entity Name	List {POI_CAT} in {REGION} with elevation above the area's average.	Zonal	Intersects, Elevation
VR10	Entity Name	Which {POI_CAT} in {REGION} are within {DISTANCE} m of terrain at or above {ELEV_COND} m?	Global	Intersects, Elevation Proximity
VR11	Entity Name	What are the {TOP_N} {POI_CAT} at the lowest elevations in {REGION}?	Local	Intersects, Elevation
VR12	Entity Name	Which category has the higher average elevation in {REGION}: {POI_CAT$_1$} or {POI_CAT$_2$}?	Zonal	Intersects, Elevation
VR13	Entity Name	Which {POI_CAT} in {REGION} are on slopes steeper than {SLOPE_COND}°?	Focal	Intersects, Slope
VR14	Entity Name	What are the {TOP_N} {POI_CAT} on the steepest terrain in {REGION}?	Focal	Intersects, Slope

spatial predicates capture the geometry side of the query separately. For templates that combine vector filtering with raster access, the label reflects the raster access pattern rather than the full query. For example, slope along a road (R10, R11, VR5, VR6) is labeled local, because the actual workflow samples DEM elevation at road segment endpoints, while the line sampling and slope predicates record the line geometry. VR7, VR8, VR9, and VR12 are labeled zonal, because their terrain quantities depend on region level aggregation. This convention keeps the taxonomy aligned with the four map algebra classes and avoids combined labels in the tables.

2.3.1 Raster-Only Templates (R1–R11). The raster-only templates isolate pure raster reasoning without vector filtering, giving a baseline for whether systems can retrieve and analyze terrain directly. R1 through R4 are local operations, because each reads the DEM value at one or two anchor points, such as retrieving elevation at a POI and comparing elevation between two locations. R5 through R7 are focal operations, because they derive a value from the cells surrounding a location rather than from a single cell, namely slope, aspect, and ruggedness. R8 and R9 are global operations, because they search the whole DEM surface for the nearest terrain above or below an elevation threshold rather than reading a fixed location. R10 and R11 estimate average route slope by sampling DEM elevation at road segment endpoints and averaging the per segment grade. The output is a slope, but the access pattern is local, because it reads point values along the road rather than building a slope raster or aggregating over a zone.

2.3.2 Extended Raster-Vector Templates (VR1–VR8). The extended templates add a single terrain constraint or terrain output to a standard vector query, so a system must combine vector reasoning with raster analysis. VR1 through VR4 extend range based POI queries with elevation filtering or elevation aware outputs. They are local, because once the vector condition selects the candidate POIs, the raster access reads the DEM value at each POI point. VR5 and VR6 select a road and return its slope through DEM line

sampling, which is again local for the same reason as R10 and R11, since the slope comes from elevation sampled at points along the road. VR7 and VR8 are zonal, because their terrain quantities are aggregated over a region. VR7 ranks region entities and takes the maximum elevation over the region, and VR8 combines park area with the average slope of the region.

2.3.3 New Raster-Vector Templates (VR9–VR14). The new templates require tighter interaction between vector filtering and terrain analysis. VR9 and VR12 are zonal, because they compare POIs against a region wide terrain aggregate. VR9 selects POIs above the average elevation of their region, and VR12 compares the average elevation of two POI categories within a region. VR10 is global, because it searches the surface for terrain that meets an elevation threshold near each candidate POI. VR11 is local, because it reads the DEM elevation at each POI and then ranks by lowest value. VR13 and VR14 are focal, because they filter or rank POIs by slope, which is derived from neighborhood cells of each POI. To make the constraints selective and geographically meaningful, these templates use elevated thresholds such as 1,000 to 2,000 m elevation and 30 to 45° slope.

2.4 Raster Question Parameters

The raster extension introduces several new parameters beyond those defined in GS-QA [13]. Table 8 summarizes all parameters, both inherited and new.

Road route (ROAD_ROUTE). For templates that compute slope along a specific road geometry, we restrict the road selector to short road types with at most 50 vertices, namely residential, footway, cycleway, pedestrian, service, and living street. This avoids long highways or motorways whose thousands of segments would exceed the query timeout.

Region (REGION). For all raster vector templates, we restrict regions to small administrative types, namely town, village, neighbourhood, and suburb. This keeps the number of POIs per region manageable, since larger types such as counties or states would require tens of thousands of per POI DEM lookups per query.

Elevation condition (ELEV_COND). An elevation value in meters used as a filter or comparison threshold. The range depends on the template, namely 20 to 500 m in steps of 20 m for the raster only and extended templates (R4, R9, VR1 through VR3), and 1,000 to 2,000 m in steps of 100 m for templates that target high elevation areas (R8, VR10). The higher ranges create selective filtering conditions in mountainous regions.

Slope condition (SLOPE_COND). A slope value in degrees used as a filter. For steep terrain templates (VR13), values are drawn from $\{30, 31, \ldots, 45\}^\circ$.

Top-N (TOP_N). For templates that return ranked results (VR11, VR14), N is drawn from $\{3, 5, 10\}$.

2.5 Question and Ground Truth Generation

We use an automated pipeline to construct question and answer pairs from the geospatial dataset. The pipeline first generates a natural language question and then computes its ground truth answer. We generate 20 distinct questions for each of the 25 raster related templates, giving 500 raster related benchmark questions.

Table 8: Question Parameters Summary

Parameter	Description
ROAD_ROUTE	Short road segment (≤ 50 vertices) of type: residential, footway, cycleway, pedestrian, service, or living_street, within CONUS
REGION	Small administrative region (town, village, neighbourhood, or suburb) within the contiguous US (CONUS)
ELEV_COND	Elevation condition in meters. Range varies by template: 20–500 m (R4, R9, VR1–VR3), 200–500 m (VR11), or 1,000–2,000 m (R8, VR10)
SLOPE_COND	Slope condition in degrees: 30–45° (VR14)
TOP_N	Number of top results to return: 3, 5, or 10
ANCH_POI	Anchor POI name from one of these categories: aquarium, attraction, viewpoint, art gallery, theme park, museum, gallery, zoo, hotel, restaurant, hospital, university, café, park, beach resort, golf course, nature reserve, garden, stadium
POI_CAT	POI category name: same set as ANCH_POI
PARK_WATB	Park or water body type: recreation ground, nature reserve, park, garden, golf course, marina, bay, harbour, lake, or reservoir
ROAD_TYPE	Road type for KNN/intersects queries: road, secondary, pedestrian, primary, track, or motorway
DISTANCE	Distance: 500–5,000 m for range queries

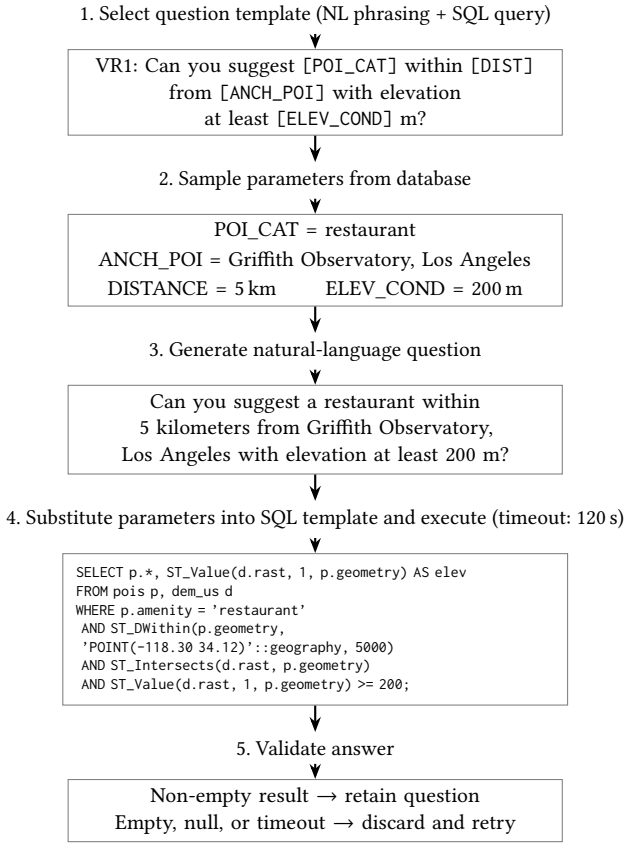
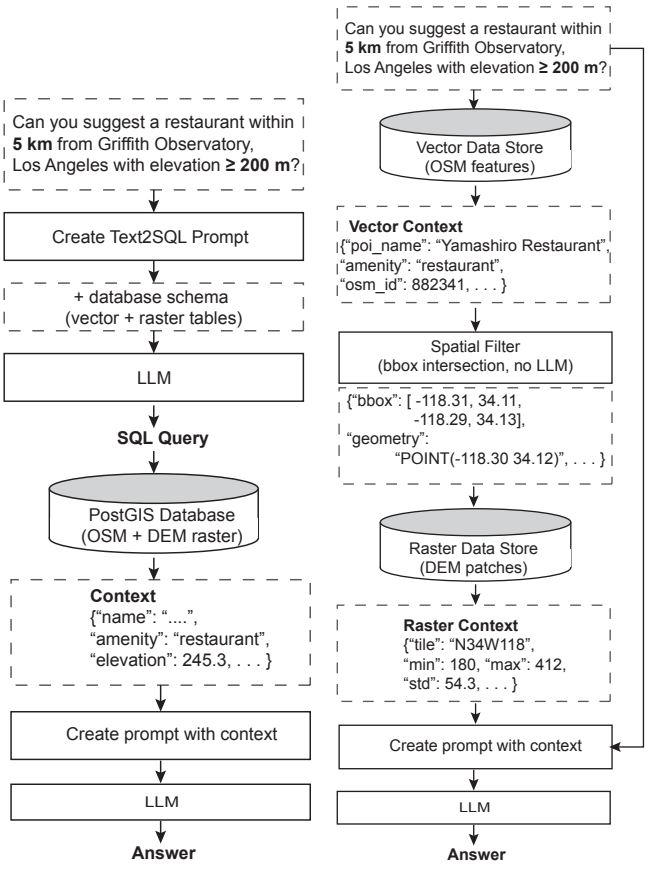


Figure 1: Example of Generating a Question from a Template

These complement the 2,800 vector questions inherited from GS-QA. Figure 1 illustrates the pipeline using template VR1, which combines a vector range query with a raster elevation condition.

To generate a question, we fill each placeholder in the template with a real entity sampled from the dataset. A POI placeholder



(a) Pipeline with Text2SQL **(b) Pipeline with RAG**

Figure 2: Overview of the two question answering pipelines.

may be filled with a specific hospital, restaurant, or museum, and a region placeholder with an actual administrative region such as a town or neighborhood. To keep the benchmark diverse, we detect duplicates by normalizing each SQL query into a canonical form and comparing it against previously generated instances. A final grammar correction step using LanguageTool [11] fixes article insertion, noun pluralization, and other surface issues.

Each template is paired with an SQL query template whose parameters mirror the natural language question. We execute this query against the benchmark database to produce the ground truth answer, and the SQL follows the raster operation label of the template. We discard and regenerate any question that returns an empty or invalid result, including cases where no entity satisfies the constraints or a sampled location falls outside valid raster coverage. Each ground truth query runs under a 120 second timeout, and queries that exceed it are discarded and regenerated. Only questions with valid non empty answers are kept.

3 Baselines

We evaluate four representative baselines, including Text2SQL, RAG, CHESS, and GIS Copilot. The first two follow the original GS-QA evaluation [13]. Text2SQL generates and runs a PostGIS SQL query, while RAG retrieves relevant records and answers from them. CHESS [15] and GIS Copilot [1] are more recent agent systems built from multiple stages. All baselines use Gemini-2.5-Flash. Below we describe each baseline and the adaptations required for this benchmark.

3.1 Text2SQL

In the Text2SQL baseline, the question is first combined with the database schema to create a Text2SQL prompt. Based on the prompt, the LLM generates a PostGIS SQL query, which is executed against the database. The execution result is then formatted as context and passed to a second LLM call, which produces the final natural language answer. Figure 2a illustrates this process.

This baseline tests whether an LLM can map natural language geospatial questions to executable SQL, including raster functions such as slope. The main challenge is that DEM raster operations are largely absent from standard text to SQL training data, so the model must generalize beyond common relational query patterns. We adapt the GS-QA Text2SQL prompt by adding raster table information and metadata, including tile size, resolution, coordinate reference system, and spatial coverage. We exclude example SQL and raster specific instructions, so the results reflect the ability of the model to infer the required operations from the schema and question.

3.2 Retrieval-Augmented Generation

The RAG baseline answers questions by retrieval rather than SQL generation. We build a vector store over database records. Given a question, the system retrieves the top k most relevant records and passes them to the LLM as context for answer generation. This avoids requiring the LLM to generate PostGIS syntax, but it also has an important limitation. Dense retrieval measures textual similarity, not spatial relationships [13]. A record can be textually relevant but geographically irrelevant. This limitation is stronger for raster queries, because terrain values such as elevation and slope are numeric surface properties rather than naturally retrievable text.

We adapt RAG by indexing both vector features and raster patches. Vector features are embedded as structured text of their names, categories, and geometry types, while their full geometry records are kept as metadata. For raster data, we extract metadata from each GeoTIFF file, including CRS, band information, spatial extent, and summary statistics such as minimum, maximum, and standard deviation, then split each raster into 256×256 patches. All patches from a file share one embedding derived from this metadata, while each keeps its own spatial extent and raster window for spatial filtering.

At query time, we retrieve the top $k = 5$ vector records by dense similarity, then use their stored geometries to find raster patches whose bounding boxes intersect the retrieved features. This intersection runs in geospatial code, not the LLM. The matching patches are ranked by embedding similarity, and the final prompt gives the LLM the original question together with the retrieved vector and raster context.

3.3 CHESS

CHESS [15] is a text to SQL framework that runs in several stages and is designed for large databases. In its original design, CHESS

first retrieves relevant database values and schema context, then selects the tables and columns most relevant to the question. It then generates and revises candidate SQL queries, asks the LLM to write natural language unit tests that distinguish the candidates, and uses a further LLM evaluation to select the final output. Compared with direct Text2SQL, CHESS provides more opportunities to ground values, ground the schema, recover constraints, and correct errors.

The original CHESS system targets SQLite on the BIRD benchmark. We adapt it to PostGIS for geospatial question answering. First, we use a preprocessed static PostGIS schema of OSM and raster DEM data rather than discovering the schema at runtime, which avoids high latency on large geospatial tables. Second, we generate PostGIS SQL rather than SQLite SQL, so the queries can use spatial predicates and raster functions. We do not use SpatiaLite because its raster support is weaker than that of PostGIS. To stay within the limit of six minutes per query, our CHESS-PostGIS configuration turns off table and column selection, uses two generation prompts sampled once each to produce two candidate queries, applies one revision step to any candidate that fails execution, and selects the final query through execution result clustering and unit tests generated by the LLM.

3.4 GIS Copilot

GIS Copilot [1] differs from the three baselines that generate SQL. Rather than generating database queries, it uses an LLM agent to construct geospatial analysis workflows by selecting GIS tools, configuring their parameters, and producing executable Python code. The original system is designed for QGIS application and uses an iterative generation and repair loop, allowing up to five rounds to fix errors in the generated workflow. The workflows can use QGIS processing tools as well as geospatial Python libraries such as GeoPandas and Rasterio. We include GIS Copilot as an alternative that generates code rather than database queries, testing whether a geospatial agent that calls tools can perform vector and raster reasoning outside SQL.

We adapt GIS Copilot to run in a headless command line setting and revise its prompts to include dataset paths, OSM schemas, and DEM inputs. Because it processes files directly rather than querying a database, the generated workflows often operate over large GeoJSON and raster files using sequential scans and temporary in memory processing rather than persistent spatial indexes. Although code generation and repair usually finish within the time limit, workflow execution frequently exceeds the runtime budget. We therefore exclude GIS Copilot from the main quantitative comparison.

4 Experiment

4.1 Experiment Setup

Experiments were run on a single node with 2× Intel Xeon E5-2609 v4 CPUs at 1.70GHz, 16 CPU cores, and 125GB RAM. We evaluate all baselines using Gemini-2.5-Flash and require each query to complete answer generation within six minutes.

All SQL based baselines share the same PostGIS database, raster tiles, metadata, and spatial indexes. This design isolates differences in query generation and reasoning from differences in storage layout or database configuration.

Table 9: Vector-only output types and evaluation metrics.

Output Type	Metric	Tolerance
Entity name	Token F1	≥ 0.8
Location	Geodesic distance error	≤ 5 m
Direction	Circular angular error	$\leq 5^\circ$
Area, length, distance, or count	Relative error	≤ 0.05

Table 10: Raster and raster-vector output types and evaluation metrics.

Output Type	Metric	Tolerance
Elevation: point, difference, or statistic	Absolute error	≤ 10 m
Elevation coverage	Absolute percentage-point error	≤ 5 pp
Threshold label	Exact match	correct label
Slope: point, route, or area statistic	Absolute error	$\leq 5^\circ$
Aspect	Circular angular error	$\leq 5^\circ$
Ruggedness	Relative error	≤ 0.05
Entity name or comparison answer	Token F1	≥ 0.8
Location	Geodesic distance error	≤ 5 m
Area, length, distance, or count	Relative error	≤ 0.05
Compound outputs	Conjunction of component metrics	all components pass

The following sections first describe the evaluation metrics and their tolerances, then analyze performance on vector only and raster related queries. Appendix A reports SQL execution times, token usage, and failure analysis, including unanswered queries and incorrectly generated SQL. The benchmark and source code are available at https://github.com/ZhuochengShang/QARV.

4.2 Evaluation Metrics

Following GS-QA [13], we evaluate each query according to its output type, since different outputs require different metrics. Tables 9 and 10 summarize the metrics and thresholds. The entity name, location, direction, and numeric thresholds follow GS-QA [13], while the raster thresholds are grounded in reported DEM accuracy.

For free text answers such as entity names, we report token level F1 together with recall. Before comparison, we remove punctuation and lowercase all tokens. Let T_{pred} and T_{gt} denote the token multisets of the predicted and ground truth answers, and let M be the number of matching tokens between them:

$$M = |T_{\text{pred}} \cap T_{\text{gt}}| \tag{1}$$

Precision, recall, and F1 are then defined as

$$P = \frac{M}{|T_{\text{pred}}|} \qquad R = \frac{M}{|T_{\text{gt}}|} \qquad F_1 = \frac{2PR}{P+R} \tag{2}$$

If either answer is empty or no tokens match, P, R, and F_1 are set to zero. We use F1 rather than exact string match because entity names often differ only in articles, ordering, or minor wording, and a parsed entity name is counted correct when $F_1 \geq 0.8$.

For direction queries, we compute the smallest angular difference between the predicted bearing $\hat{\theta}$ and the ground truth bearing θ:

$$E_{\text{angle}} = \min\left(|\hat{\theta} - \theta| \bmod 360^\circ, 360^\circ - (|\hat{\theta} - \theta| \bmod 360^\circ)\right) \tag{3}$$

This circular formulation correctly handles the 0° and 360° boundary. A direction is considered correct when $E_{\text{angle}} \leq 5^\circ$.

Table 11: Vector-only entity name and free text answer quality on attempted questions (Gemini-2.5-Flash). Boldface marks the best parsed F1 score.

ID	Text2SQL		CHESS-PostGIS		RAG	
	Text Recall	Parsed F1	Parsed Recall	Parsed F1	Text Recall	Parsed F1
V1	0.11	**0.96**	0.90	0.71	0.05	0.75
V2	0.24	0.78	0.91	**0.87**	0.11	0.53
V3	0.21	**0.61**	0.96	0.44	0.05	0.48
V4	0.03	0.37	0.89	0.36	0.03	**0.62**
V5	0.24	0.70	0.77	**0.72**	0.14	0.19
V6	0.39	0.61	0.72	**0.67**	0.17	0.23
V7	0.06	0.07	0.84	**0.80**	0.10	0.26
V8	0.16	0.31	0.64	**0.63**	0.10	0.26
V9	0.51	**0.74**	0.74	0.39	0.14	0.18
V10	0.07	**0.18**	0.42	**0.18**	0.04	0.05
V11	0.16	**0.83**	0.68	0.74	0.10	0.38
V12	0.15	0.64	0.65	**0.65**	0.17	0.45
AVG	0.19	0.57	0.76	**0.60**	0.10	0.36

For numeric quantities such as area, length, distance, count, and ruggedness, we use relative error:

$$E_{\text{rel}} = \frac{|\hat{y} - y|}{|y|} \tag{4}$$

where $\hat{y}$ is the predicted value and y is the ground truth value. A numeric answer is considered correct when $E_{\text{rel}} \leq 0.05$. When the ground truth value is zero, a predicted zero is correct and any nonzero prediction is incorrect. Mean relative errors in the result tables are capped at 1.0 and averaged over attempted questions.

For raster terrain with elevation and slope values, we use absolute error:

$$E_{\text{abs}} = |\hat{y} - y| \tag{5}$$

Elevation outputs, including point values, differences, and summary statistics, are considered correct when $E_{\text{abs}} \leq 10$ m, which matches the reported vertical accuracy of open source DEM products [4, 14]. For slope and aspect, we use a stricter 5° threshold, roughly half the slope error reported for open source DEMs [4].

Raster-vector and extended queries may return multiple values, such as an entity name and its elevation, or a road name together with distance and slope. Each required value is evaluated independently using its corresponding metric, and a query is considered correct only when every required output satisfies its threshold.

4.3 Vector-Only Results Analysis

Table 11 reports entity-name results. CHESS-PostGIS has the best mean F1 (0.60), followed by Text2SQL (0.57) and RAG (0.36), but the winner varies by template. These queries require two decisions. The system must ground the anchor entity, then select the target satisfying the spatial predicate. Text2SQL performs best when both steps reduce to simple attribute and spatial filters, but it fails when names or OSM tags do not match the database exactly. CHESS-PostGIS benefits from schema linking and revision, which can recover missing constraints before execution. RAG is competitive only when lexical retrieval finds similar names, but it cannot reliably enforce spatial predicates.

Table 12: Vector-only strict accuracy for templates with location and direction answers (Gemini-2.5-Flash).

ID	Text2SQL Acc.	CHESS-PostGIS Acc.	RAG Acc.
Location: geodesic error ≤ 5 m			
V13	0.010	**0.710**	0.160
V14	0.000	**0.320**	0.060
V15	0.000	0.000	**0.070**
V16	0.000	**0.530**	0.120
V17	0.010	**0.370**	0.050
V18	0.000	**0.190**	0.010
V19	0.000	0.000	0.000
V20	0.000	**0.130**	0.000
AVG	0.003	**0.281**	0.059
Direction: circular angular error ≤ 5°			
V21	**0.570**	0.510	0.160
V22	**0.420**	0.390	0.030
AVG	**0.495**	0.450	0.095

Table 13: Vector-only numeric-answer evaluation (Gemini-2.5-Flash). Accuracy uses relative error ≤ 0.05. Mean relative error is capped at 1.0 and averaged over attempted questions.

Output	ID	Text2SQL		CHESS-PostGIS		RAG	
		Acc.	Rel. Err.	Acc.	Rel. Err.	Acc.	Rel. Err.
Count	V23	0.17	0.64	**0.35**	0.51	0.00	0.96
	V24	0.04	0.89	**0.13**	0.83	0.06	0.91
Distance	V25	**0.69**	0.19	0.17	0.57	0.04	0.84
	V26	**0.64**	0.23	0.52	0.34	0.00	0.89
Area	V27	0.00	–	**0.22**	0.20	0.00	1.00
Length	V28	0.00	0.88	**0.12**	0.52	0.00	1.00
AVG	–	**0.26**	0.57	0.25	0.50	0.02	0.93

Table 12 reports location and direction results. For location outputs, CHESS-PostGIS is clearly strongest, averaging 0.281 against 0.003 for Text2SQL and 0.059 for RAG. Location is scored against the centroid of the reference OSM geometry under a 5 m threshold. Once CHESS-PostGIS selects the correct feature, its geometry-derived coordinate usually matches this centroid closely. Text2SQL often returns no parseable location, and when it returns an address or point, it is usually a geocoded address or nearby facility rather than the target centroid. RAG can slightly improve over Text2SQL when retrieved text contains a place name or address, but geocoded text stays unreliable at a 5 m threshold because it does not expose the exact OSM geometry.

For direction outputs, Text2SQL (0.495) and CHESS-PostGIS (0.450) perform similarly, and both are far above RAG (0.095). Direction is a scalar relation between two geometries. After the anchor and target are selected, the bearing can be computed directly, and small rounding differences usually remain within the 5° tolerance. RAG must infer direction from text that contains no computed azimuth, so it often predicts the wrong angle.

Table 14: Raster-related strict accuracy grouped by raster operation (Gemini-2.5-Flash). RAG is included for completeness, but retrieved context usually lacks the pixel-level values required for raster computation.

Raster Op.	ID	Query Type	Text2SQL Acc.	CHESS-PostGIS Acc.	RAG Acc.
Local	R1	elev. + POI	**1.000**	0.000	0.000
	R2	elev. cmp. + two POIs	**0.450**	0.000	0.000
	R3	elev. diff + two POIs	**1.000**	0.000	0.000
	R4	elev. threshold + POI	**0.950**	0.200	0.000
	VR1	range + name + elev. cond.	**0.600**	0.000	0.000
	VR2	range + loc. + elev. cond.	0.000	0.000	0.000
	VR3	range + count + elev. cond.	**0.600**	0.050	0.000
	VR4	range + dist. + elev.	**0.200**	0.000	0.000
	VR11	elev. lowest *n* POIs	**0.300**	0.150	0.000
Focal	R5	slope + POI	0.000	0.000	0.000
	R6	aspect + POI	0.250	**0.550**	0.000
	R7	ruggedness + POI	**0.300**	**0.300**	0.000
	R10	slope + route	0.000	0.000	**0.050**
	R11	slope cmp. + route	0.000	0.000	0.000
	VR13	slope cond. + name	0.000	0.000	0.000
	VR14	slope steepest *n* POIs	0.050	**0.100**	0.000
Zonal	VR7	intersect area + max elev.	0.000	0.000	0.000
	VR8	intersect area + avg slope	0.000	0.000	0.000
	VR9	elev. region name (avg)	0.000	0.000	0.000
	VR12	elev. cmp. cat. + name	0.000	0.000	0.000
Global	R8	nearest high terr. + POI	**0.150**	0.000	0.000
	R9	nearest low terr. + POI	0.000	0.000	0.000
	VR5	knn + dist. + slope	0.000	0.000	0.000
	VR10	elev. prox. zone + name	0.000	0.000	0.000
Overall Accuracy			**0.234**	0.054	0.002

Table 13 summarizes numeric answers. Text2SQL (0.26) and CHESS-PostGIS (0.25) nearly tie in mean accuracy, but their strengths differ. CHESS-PostGIS does better on count, area, and length queries, while Text2SQL does better on distance queries. CHESS-PostGIS also has lower mean relative error (0.50 against 0.57), so its answers are often closer even when they miss the 5% threshold. The main lesson is that executable SQL is not enough. A query can run but still measure the full feature instead of the clipped intersection, use degrees instead of meters, or count before applying all filters. Area and length are most exposed because they require intersection, clipping, combining pieces, and unit-correct measurement. Distance is less exposed because, after grounding the source and target classes, it usually reduces to a nearest-distance computation.

4.4 Raster-Related Results Analysis

Table 14 reports raster related results grouped by raster operation. Across all baselines, raster related templates are substantially more difficult than vector only templates, indicating that raster reasoning remains a major challenge for current LLM based geospatial question answering systems. Text2SQL leads overall with 0.234 accuracy, while CHESS-PostGIS reaches only 0.054. RAG is omitted because retrieved text does not contain the pixel-level values needed for raster computation.

Local operations are the easiest for Text2SQL because they usually reduce to sampling one DEM value at a grounded geometry. Text2SQL is perfect on point elevation R1 and elevation difference R3, and strong on elevation threshold R4. These templates require a short chain. The system grounds the POI, finds the covering raster tile, and then returns, compares, or subtracts the sampled value. CHESS-PostGIS scores zero on R1 and R3, suggesting that its multi-stage pipeline can replace the exact point-to-cell lookup with broader operations such as clipping, merging tiles, or aggregating over an expanded geometry. Text2SQL also keeps partial accuracy when local lookup is combined with simple vector predicates, as in VR1 and VR3, but fails on VR2 because location output must match the reference centroid within 5 m.

Focal operations are harder because they derive terrain attributes from a cell neighborhood rather than reading a single DEM value. For R5 through R7, the query must choose the correct terrain function, use the right scale and units, and sample the derived raster at the target point. Slope at a POI (R5) fails for both systems, while aspect (R6) is the main exception. CHESS-PostGIS reaches 0.550 against 0.250 for Text2SQL, which suggests that staged generation helps with selecting the correct PostGIS raster function. Ruggedness (R7) stays low but nonzero for both at 0.300. R10 and R11 also output slope, but they are route-slope templates rather than focal point operations. Their ground truth samples DEM values at road segment endpoints, computes the per-segment grade, and averages by segment length, so small errors in geometry, segmentation, units, or weighting change the final value. RAG has one correct R10 case (0.05), but this is an isolated tolerance match from retrieved textual context rather than evidence that retrieval can compute route slope.

Zonal operations fail completely for both systems. They require selecting the correct polygonal support through region selection, overlay, or intersection, and then aggregating terrain values over that zone. Neither baseline reliably constructs the right zone and statistic.

Global operations are similarly difficult. Text2SQL keeps only limited accuracy on nearest high terrain R8, while R9, VR5, and VR10 are zero for both systems. These templates search the raster surface for terrain meeting a condition and then add distance or proximity reasoning, combining the grounding, function-selection, and measurement errors seen in the other groups.

4.5 Discussion

The central challenge in geospatial question answering is not producing executable SQL, but preserving spatial semantics over complex geospatial data. A single question may involve vector geometries such as points, lines, and polygons, raster cells, and terrain properties such as slope and aspect, each associated with its own spatial support, coordinate system, and units. Correctness therefore depends on composing the right operations in the correct order rather than merely generating syntactically valid SQL.

Vector query difficulty largely depends on the amount of spatial processing required after entity grounding. Entity name, direction, and distance queries are the most tractable because the answer often follows once the relevant entities are identified correctly. Location queries are more challenging because the predicted output must correspond to the exact database geometry rather than a nearby geocoded location. Area and length queries are harder still because they require geometry intersection, aggregation, and unit aware measurement. These additional operations create opportunities for errors even when the generated SQL is executable.

Raster queries expose a substantially larger gap, and the gap widens sharply with the raster operation type. Local operations such as point elevation retrieval achieve the strongest performance because they reduce to sampling a single DEM cell. Focal operations are far harder, because the system must select and apply the correct

terrain function, such as slope, aspect, or ruggedness, and sample it at the right location. Zonal operations add another level of difficulty by requiring aggregation over the correct spatial support, and global operations are the hardest because they combine a search across the raster surface with further spatial constraints. Accuracy therefore falls steeply as raster reasoning moves from local lookup to focal, zonal, and global analysis, and the zonal and global cases collapse to near zero for every baseline.

Taken together, these results point to a single underlying difficulty. A correct answer requires understanding the geospatial data, not just producing runnable SQL. A system must map the spatial meaning of the question to the right operations, apply those operations in the correct order, and return the answer type the question asks for. Raster reasoning over a structured database brings all of these demands together, which makes it the central challenge that remains for future geospatial question answering systems.

5 Related Work

This work relates to three lines of research. The first is geospatial question answering benchmarks, which evaluate how well systems answer natural language questions over geographic data. The second is text to SQL benchmarks, which test query generation over relational databases. The third is vision based geospatial benchmarks, which assess foundation models on satellite and aerial imagery. Across all three, raster terrain reasoning over a structured database remains unaddressed, which is the gap our benchmark targets.

Geospatial Question Answering Benchmarks. Geospatial question answering benchmarks assess the ability of systems to answer natural language questions about geographic entities, spatial relationships, and raster data attributes using geographic data. Early benchmarks such as GeoQuestions201 [12] and GeoQA2 [6, 7] map natural language questions to SPARQL queries over geospatial knowledge graphs such as YAGO2geo, covering topological and attribute queries but not executable database queries. GeoGLUE [9] shifts to language understanding tasks such as geolocating a place from a textual description and classifying feature types such as roads and points of interest, but does not address free text spatial query answering. More recent benchmarks move toward real world map data. MapQA [2] uses OpenStreetMap geometries, MapVerse [3] evaluates multimodal question answering over rendered maps, and EVGeoQA [17] targets vector based routing and connectivity queries. GS-QA [13] grounds evaluation in a PostGIS database populated from OpenStreetMap, generating template based questions that are scored against SQL computed answers. Despite these advances, all of these benchmarks evaluate vector representations exclusively, and none assess raster data such as terrain attributes with elevation and slope.

Our work fills this gap by augmenting a PostGIS benchmark built on OSM with a Digital Elevation Model (DEM) raster layer and 25 query templates that require raster only and raster vector reasoning. Concurrent agent oriented benchmarks, GeoAgentBench and GeoAnalystBench [18, 20], combine raster and vector data but focus on multi step planning and analytical workflows, evaluating how well different LLMs orchestrate tool use rather than how accurately they answer structured spatial queries. Our benchmark is complementary in that we isolate SQL level spatial reasoning as a controlled, reproducible evaluation target.

Text to SQL Benchmarks. A parallel line of work answers free text queries through SQL generation over relational databases. Spider [19] established cross domain evaluation across databases with diverse schema, and BIRD [10] extended this by incorporating database content awareness and scoring answers by execution results. Both benchmarks operate over standard relational schema with no spatial types or geometry columns. GS-QA [13] brought this paradigm to geospatial data by introducing PostGIS spatial SQL over OpenStreetMap vector tables, but its queries are limited to vector data and it was never tested on sophisticated systems. Our benchmark extends GS-QA by adding a DEM raster layer and query templates that require composing vector predicates with terrain operations, a query type absent from all existing text to SQL benchmarks. We evaluate both a Text2SQL baseline and the CHESS multi agent framework [15] to establish initial performance bounds.

Vision Based Geospatial Benchmarks. A separate line of work evaluates foundation models on satellite and aerial imagery. GeoBench and GeoXBench [8, 22] cover downstream perception tasks including object detection, classification, and segmentation over raster imagery. These benchmarks target visual understanding rather than database backed question answering and are orthogonal to our scope.

6 Conclusion

We presented GS-QA2, a benchmark for geospatial question answering over both vector and raster data. GS-QA2 extends GS-QA with a DEM raster dataset and 25 new templates that evaluate raster and raster vector reasoning. The benchmark provides executable SQL ground truth, systematic coverage of vector and raster operations, and evaluation metrics tailored to heterogeneous geospatial outputs. Our evaluation reveals a clear progression in difficulty. Vector queries are generally manageable once the relevant entities and spatial predicates are identified correctly. Raster queries introduce substantially greater challenges. Systems perform reasonably well on simple elevation retrieval tasks, but accuracy declines sharply for terrain analysis, regional aggregation, and raster vector reasoning. Performance is particularly poor on focal, zonal, and global raster operations that require combining multiple spatial operations in the correct order over the right cells and regions. These findings suggest that the primary challenge in geospatial question answering is not generating executable SQL, but understanding the geospatial semantics expressed in natural language and translating them into the correct sequence of spatial operations. Future progress will require systems that can reason the intersection between vector and raster data, compute over the right spatial regions, and correctly apply analysis and aggregation operations. GS-QA2 provides a reproducible benchmark for measuring progress toward this goal.

References

[1] Temitope Akinboyewa, Zhenlong Li, Huan Ning, and M Naser Lessani. 2025. GIS copilot: Towards an autonomous GIS agent for spatial analysis. *International Journal of Digital Earth* 18, 1 (2025), 2497489.

[2] Christian Michael Arnold, Andrew Alini, Jonathan Wang, Pieter M Feenstra, Conner Arnold, Jan DeWitt, Natalie C Ritsema, Jung Hyun Yae, Boris Katz,

Andrei Barbu, et al. 2026. MapQA: A Map-Question-Answering Benchmark for Visual Language Model Reasoning. (2026).

[3] Sharat Bhat, Harshita Khandelwal, Tushar Kataria, and Vivek Gupta. 2026. MapVerse: A Benchmark for Geospatial Question Answering on Diverse Real-World Maps. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 8168–8178.

[4] M El Hage, E Simonetto, G Faour, and L Polidori. 2012. Evaluation of elevation, slope and stream network quality of SPOT DEMs. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 1 (2012), 63–67.

[5] European Space Agency (ESA). 2023. Copernicus Digital Elevation Model (COP-DEM). doi:10.5270/ESA-c5d3d65

[6] Nikolaos Karalis, Georgios Mandilaras, and Manolis Koubarakis. 2019. Extending the YAGO2 knowledge graph with precise geospatial knowledge. In *International Semantic Web Conference*. Springer, 181–197.

[7] Sergios-Anestis Kefalidis, Dharmen Punjani, Eleni Tsalapati, Konstantinos Plas, Maria-Aggeliki Pollali, Pierre Maret, and Manolis Koubarakis. 2024. The question answering system GeoQA2 and a new benchmark for its evaluation. *International Journal of Applied Earth Observation and Geoinformation* 134 (2024), 104203.

[8] Alexandre Lacoste, Nils Lehmann, Pau Rodriguez, Evan Sherwin, Hannah Kerner, Björn Lütjens, Jeremy Irvin, David Dao, Hamed Alemohammad, Alexandre Drouin, et al. 2023. Geo-bench: Toward foundation models for earth monitoring. *Advances in Neural Information Processing Systems* 36 (2023), 51080–51093.

[9] Dongyang Li, Ruixue Ding, Qiang Zhang, Zheng Li, Boli Chen, Pengjun Xie, Yao Xu, Xin Li, Ning Guo, Fei Huang, et al. 2023. Geoglue: A geographic language understanding evaluation benchmark. *arXiv preprint arXiv:2305.06545* (2023).

[10] Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. 2023. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems* 36 (2023), 42330–42357.

[11] Daniel Naber and LanguageTool Contributors. 2026. LanguageTool: Open Source Proofreading Software. https://languagetool.org Accessed May 2026.

[12] Dharmen Punjani, Kuldeep Singh, Andreas Both, Manolis Koubarakis, Iosif Angelidis, Konstantina Bereta, Themis Beris, Dimitris Bilidas, Theofilos Ioannidis, Nikolaos Karalis, et al. 2018. Template-based question answering over linked geospatial data. In *Proceedings of the 12th workshop on geographic information retrieval*. 1–10.

[13] Majid Saeedan, Muhammad Shihab Rashid, Ahmed Eldawy, and Vagelis Hristidis. 2026. GS-QA: A Benchmark for Geospatial Question Answering. arXiv:2605.22811 [cs.DB] https://arxiv.org/abs/2605.22811

[14] Danang Budi Susetyo. 2023. Vertical accuracy assessment of various open-source DEM data: DEMNAS, SRTM-1, and ASTER GDEM. *Geodesy and Cartography* 49, 4 (2023), 209–215.

[15] Shayan Talaei, Mohammadreza Pourreza, Yu-Chen Chang, Azalia Mirhoseini, and Amin Saberi. 2024. Chess: Contextual harnessing for efficient sql synthesis. *arXiv preprint arXiv:2405.16755* (2024).

[16] C Dana Tomlin et al. 1990. *Geographic information systems and cartographic modeling*. Vol. 249. Prentice Hall Englewood Cliffs, NJ.

[17] Jianfei Wu, Zhichun Wang, Zhensheng Wang, and Zhiyu He. 2026. EVGeoQA: Benchmarking LLMs on Dynamic, Multi-Objective Geo-Spatial Exploration. *arXiv preprint arXiv:2604.07070* (2026).

[18] Bo Yu, Cheng Yang, Dongyang Hou, Chengfu Liu, Jiayao Liu, Chi Wang, Zhiming Zhang, Haifeng Li, and Wentao Yang. 2026. GeoAgentBench: A Dynamic Execution Benchmark for Tool-Augmented Agents in Spatial Analysis. *arXiv preprint arXiv:2604.13888* (2026).

[19] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 conference on empirical methods in natural language processing*. 3911–3921.

[20] Qianheng Zhang, Song Gao, Chen Wei, Yibo Zhao, Ying Nie, Ziru Chen, Shijie Chen, Yu Su, and Huan Sun. 2025. GeoAnalystBench: A GeoAI benchmark for assessing large language models for spatial analysis workflow and code generation. *Transactions in GIS* 29, 7 (2025), e70135.

[21] Yifan Zhang, Cheng Wei, Zhengting He, and Wenhao Yu. 2024. GeoGPT: An assistant for understanding and processing geospatial tasks. *International Journal of Applied Earth Observation and Geoinformation* 131 (2024), 103976.

[22] Yushuo Zheng, Jiangyong Ying, Huiyu Duan, Chunyi Li, Zicheng Zhang, Jing Liu, Xiaohong Liu, and Guangtao Zhai. 2026. GeoX-Bench: Benchmarking Cross-View Geo-Localization and Pose Estimation Capabilities of Large Multimodal Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 13485–13493.

A Runtime and Failure Mode Summary

Table 15: Mean answer generation cost per query (Gemini-2.5-Flash)

Query Type	Method	Time	Tokens	# Queries > 6 min
Vector-only	Text2SQL	20.5 s	4.8K	12
	CHESS-PostGIS	107.2 s	31.2K	499
	RAG	16.9 s	4.3K	0
Raster-related	Text2SQL	8.0 s	3.2K	1
	CHESS-PostGIS	110.4 s	28.9K	8
	RAG	5.7 s	5.5K	0

Table 16: Final SQL execution time for successful executions only (Gemini-2.5-Flash)

Query Type	Baseline	Mean	P95	Max
Vector-only	Text2SQL	8.8 s	22.6 s	358.0 s
	CHESS-PostGIS	21.8 s	60.2 s	82.3 s
	RAG	N/A	N/A	N/A
Raster-related	Text2SQL	11.8 s	88.9 s	165.9 s
	CHESS-PostGIS	2.4 s	6.3 s	85.1 s
	RAG	N/A	N/A	N/A

This appendix supplements the results in Section 4 with generation cost, SQL execution cost, and failure mode statistics for each baseline. Table 15 separates answer generation time from SQL execution time. Generation time includes the full process of producing an answer or SQL query, while execution time measures only the runtime of the final SQL query executed against PostGIS.

Generation cost. CHESS-PostGIS has the highest generation cost, averaging 107.2s for vector queries and 110.4s for raster related queries. It also consumes roughly six to seven times more tokens than Text2SQL or RAG. This overhead comes from its multiple stages of pipeline. Text2SQL and RAG use fewer LLM stages and remain below 21s on both vector and raster related queries.

SQL execution cost. SQL execution time reflects the structure of the generated query rather than answer correctness. For vector queries, CHESS-PostGIS has a higher average execution time than Text2SQL (21.8s against 8.8s), because its selected queries often contain broader joins and more complex spatial predicates. The maximum of 358.0s for Text2SQL corresponds to a single large spatial scan rather than typical behavior. For raster related queries, CHESS-PostGIS has a lower average execution time than Text2SQL (2.4s against 11.8s), but lower execution time does not imply better performance. Many CHESS-PostGIS raster queries access only a small portion of the raster data, return NULL values, or perform incomplete raster operations. Text2SQL more often generates queries involving raster union, clipping, and regional aggregation, which are more expensive but closer to the intended computation. As a result, execution time and answer quality often diverge.

Table 17: [Vector-only] Summary of SQL execution outcomes (Gemini-2.5-Flash)

Outcome	Text2SQL	CHESS-PostGIS
Ran successfully	1758	1944
Timed out	3	534
Invalid SQL / execution error	1039	322

Table 18: [Raster-related] Summary of SQL execution outcomes (Gemini-2.5-Flash)

Outcome	Text2SQL	CHESS-PostGIS
Ran successfully	265	164
Timed out	29	153
Invalid SQL / execution error	205	183
No SQL generated	1	0

Table 19: [Vector-only] Summary of RAG answer outcomes (Gemini-2.5-Flash)

Outcome	RAG
Correct under strict evaluation	192
Insufficient retrieved evidence	1377
Generated but incorrect	1231

Table 20: [Raster-related] Summary of RAG answer outcomes (Gemini-2.5-Flash)

Outcome	RAG
Correct under strict evaluation	1
Insufficient retrieved evidence	309
Generated but incorrect	190

Failure modes. Tables 17, 18, 19, and 20 summarize the major failure modes across baselines. Text2SQL produces the largest number of invalid SQL statements and execution errors, reflecting the fragility of direct SQL generation. CHESS-PostGIS reduces invalid SQL through revision, but still has timeouts and raster specific failures, including invalid raster algebra expressions and incompatible PostGIS function arguments. These results indicate that generating syntactically valid SQL is insufficient for reliable geospatial reasoning. RAG fails differently. It avoids SQL errors entirely, but is limited by what the retrieval storage holds. For vector queries, retrieval may return plausible entities without enforcing the required spatial constraints. For raster queries, the storage holds only metadata and spatial coverage rather than the pixel values the question needs, so raster accuracy stays near zero.

Overall, the runtime and failure statistics reinforce a central finding of the paper. Generating executable SQL and producing correct geospatial answers are fundamentally different challenges. A query may execute successfully and efficiently while still selecting the wrong spatial objects, applying the wrong spatial operations, or computing over the wrong spatial regions.