HydroFusion: A Multimodal Hydrological Data Model by Fusing Knowledge Graphs and Spatiotemporal Data Cubes

Yanghui He
Changsha University of Science and Technology
Changsha, Hunan, 410000, P.R. China
15874690595@163.com

Haowei Huang
Changsha University of Science and Technology
Changsha, Hunan, 410000, P.R. China
hwhuang@csust.edu.cn

***Abstract*—Hydrological analysis increasingly relies on heterogeneous, multi-source spatiotemporal data, yet existing pipelines struggle to jointly support semantics, provenance, and high-throughput numerical analytics. This paper presents HydroFusion, a multimodal hydrological data model that couples a Knowledge Graph (KG) with a Spatiotemporal Data Cube (STDC) in a two-layer architecture. HydroFusion contributes: (1) a dynamic, process-centric hydrological ontology with container–content separation; (2) a semantic-pointer-driven hybrid storage and two-stage execution that separates graph reasoning from parallel numerical analytics; and (3) an LLM+RAG natural language interface that translates requests into Cypher and reproducible Python analytics. Experiments on CAMELS-GB, ERA5-Land, USGS NWIS, and Landsat 8 ARD show that HydroFusion achieves 53.9× and 63.4× speedups over PostGIS on cross-modal association and complex attribution queries, reaches 523.7 QPS at 32-way concurrency, and shows promising end-to-end performance on a pilot benchmark for natural-language hydrological tasks. A drought attribution case study further demonstrates provenance-aware, reproducible analysis.**

***Index Terms*—multimodal data fusion; knowledge graph; spatiotemporal data cube; dynamic ontology; semantic pointer; large language model**

I. Introduction

Water resources are increasingly studied in data-intensive settings shaped by climate change, remote sensing, IoT observation networks, and high-performance computing. Modern hydrological workflows integrate heterogeneous, high-frequency, and cross-modal data—including satellite imagery, in-situ observations, reanalysis products, and model outputs—while hydrological systems remain strongly coupled across meteorology, surface water, groundwater, and human activities. As a result, practitioners are often data rich but information poor.

Existing hydrological data management and analytics pipelines, typically based on relational spatiotemporal databases or file-system-based stitching, still struggle to support heterogeneous data organization, multi-scale spatiotemporal fusion, complex hydrological relationships, and semantic-rich, provenance-aware analysis within a unified framework.

To address these challenges, we propose HydroFusion, a multimodal hydrological data model that fuses a Knowledge Graph with a Spatiotemporal Data Cube. As shown in Fig. 1, the Knowledge Graph supports semantic modeling, association, reasoning, and provenance, while the Spatiotemporal Data Cube provides chunked storage and parallel analytics over large numerical arrays. The two layers are coupled via semantic pointers and exposed through an LLM-powered natural-language interface.

Contributions: 1) a dynamic, process-centric hydrological ontology with container–content separation to support cross-modal semantic representation and lineage tracking; 2) a semantic-pointer-driven hybrid storage architecture and a two-stage query execution mechanism that decouple semantic localization from parallel numerical computation for efficient cross-modal analytics; and 3) an LLM+RAG-based natural-language interface with a dual-tool agent that translates user requests into executable graph queries and reproducible numerical analyses.

II. Related Work

A. Spatiotemporal data management and data cubes

Relational spatiotemporal databases support spatial querying but are often constrained by I/O and limited compute pushdown on large gridded or remote-sensing data. Cloud-optimized formats and geoscience data cubes improve chunked storage, on-demand access, and scalable multidimensional analytics [1], but typically provide limited support for explicit semantic modeling and reasoning.

B. Knowledge graphs and geoscience/hydrology semantics

Knowledge graphs represent heterogeneous resources through explicit entity–relation structures and support ontology-based reasoning. GeoSPARQL standardizes geospatial semantics [2], and cross-domain KGs such as KnowWhereGraph demonstrate their value for environmental intelligence [3]. At broader geographic scale, WorldKG further shows that world-scale geographic knowledge graphs can support large-scale spatial entity integration and linking [4]. Recent reviews also document the growing role of knowledge graph construction and application across geoscientific domains [5]. Domain-focused studies in water science,

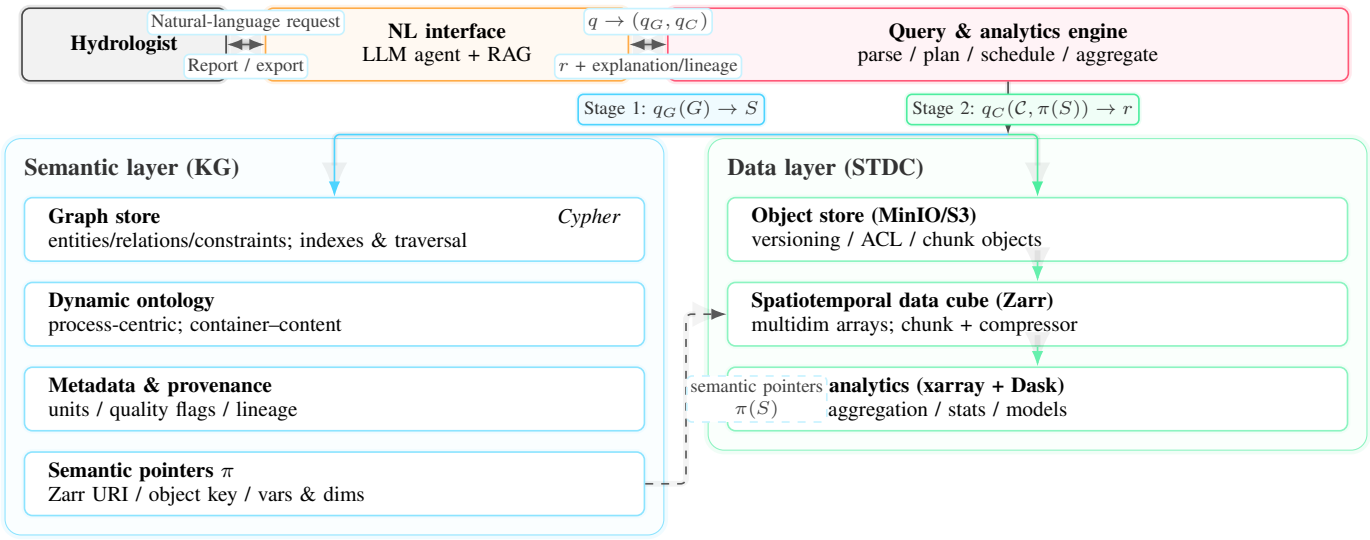

Fig. 1. HydroFusion two-layer architecture with semantic pointers and two-stage execution.

such as multi-dimensional knowledge graphs for water quality datasets, further illustrate the value of semantically organizing hydrological observations, metadata, and analysis contexts [6]. In hydrology, ontologies, provenance-aware modeling, and open KGs support intelligent applications and data reuse [7]–[9], but remain only weakly coupled to large-scale spatiotemporal numerical data.

C. Large language models and natural language data interaction

LLMs enable natural-language interaction with structured data and tool-augmented analytics, while RAG improves generation by grounding it on retrieved contexts [10], [11]. Recent geospatial studies have started to combine graph-structured knowledge with LLM-driven task planning; for example, GeoGraphRAG retrieves relevant subgraphs from an external graph knowledge base and injects both structural and semantic information into generation for automated geospatial modeling, demonstrating the value of graph-grounded generation for improving efficiency and interpretability [12]. In scientific data scenarios, the main challenge is integrating LLMs with domain knowledge and executable analytics without sacrificing interpretability, provenance, or reproducibility.

III. HydroFusion: Fusing a Knowledge Graph and a Spatiotemporal Data Cube

A. Overall model

HydroFusion consists of a semantic-layer Knowledge Graph (KG), denoted by $G = (V, E, A)$, and a data-layer Spatiotemporal Data Cube (STDC) collection, denoted by C. To decouple semantic reasoning from numerical analytics, we introduce a semantic-pointer mapping

$$\pi : V_{\text{data}} \to \mathcal{U}, \tag{1}$$

where V_{data} is the set of dataset nodes and $\mathcal{U}$ is a set of addressable resource identifiers (e.g., Zarr/object-store URIs). Given a user query q, HydroFusion decomposes it into a graph query q_G and a numerical query q_C:

$$S = q_G(G), \quad r = q_C(C, \pi(S)). \tag{2}$$

In practice, a semantic pointer encodes the minimal information required for selective chunk-level access,

$$\pi(v) = \langle u(v), \mathcal{V}(v), \mathcal{W}(v), \mathcal{K}(v) \rangle, \tag{3}$$

where $u(v)$ is the Zarr URI/object key, $\mathcal{V}(v)$ is the variable set, $\mathcal{W}(v)$ is the spatiotemporal window, and $\mathcal{K}(v)$ is the set of chunk keys. For a candidate set S, Stage 2 reads only the required chunks $\mathcal{K}(S) = \bigcup_{v \in S} \mathcal{K}(v)$, yielding

$$B(q) = \sum_{k \in \mathcal{K}(S)} |c_k|, \tag{4}$$

and an interpretable runtime decomposition

$$T(q) \approx T_G(q_G) + T_\pi(|S|) + T_{\text{IO}}(|\mathcal{K}(S)|) + T_C(|\mathcal{K}(S)|). \tag{5}$$

Stage 1 performs semantic filtering and relationship traversal, whereas Stage 2 loads only the addressed chunks and computes in parallel. In addition to numerical results, HydroFusion returns localization paths and provenance hooks that connect r back to the involved entities, constraints, and data slices.

B. Ontology and formal schema

HydroFusion adopts a dynamic, process-centric ontology for hydrological entities, representing basins, gauges, variables, events (e.g., droughts and floods), and human activities (e.g., reservoir operations) within a unified semantic framework. This design is informed by prior geoscience knowledge-graph research and water-domain case studies, which emphasize explicit semantic modeling of entities, processes, and multi-dimensional observational resources [5], [6]. Processes and evolving parameters are treated as first-class entities, enabling schema evolution and versioning of both knowledge and data

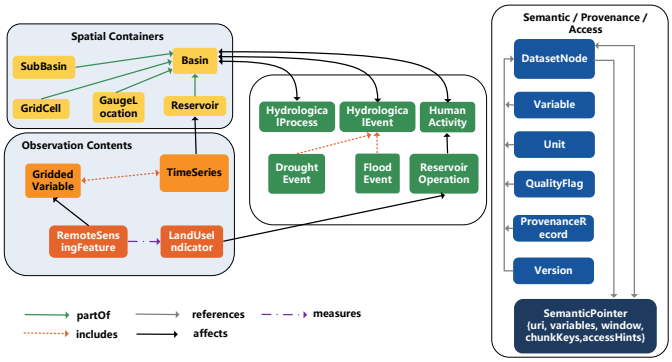

Fig. 2. Formal ontology schema of HydroFusion, showing container–content separation, semantic-pointer linkage to spatiotemporal cube resources, and provenance-aware lineage tracking.

products. A key design principle is the separation between *spatial containers* (e.g., basin boundaries, grids, and gauge locations) and *observation contents* (e.g., time series, gridded variables, and remote-sensing features), which improves cross-modal association and provenance-aware analysis. Formally, we define the ontology as

$$\mathcal{O} = \langle \mathcal{C}, \mathcal{R}, \mathcal{D} \rangle, \tag{6}$$

where $\mathcal{C}$, $\mathcal{R}$, and $\mathcal{D}$ denote classes, object relations, and data properties, respectively. As shown in Fig. 2, the schema organizes hydrological knowledge into spatial, observational, process/event, and provenance/access components. Core relations include *contains*, *locatedIn*, *upstreamOf*, *downstreamOf*, *observes*, *hasContent*, *representedBy*, *hasPointer*, *hasProvenance*, *derivedFrom*, *versionOf*, and *relatedTo*. To connect semantic entities with cube-resident numerical resources, each dataset node may be linked to a semantic pointer

$$\pi(v) = \langle u(v), V(v), W(v), K(v) \rangle, \tag{7}$$

where $u(v)$ is the resource identifier, $V(v)$ the variable set, $W(v)$ the spatiotemporal window, and $K(v)$ the chunk-key set. This formalization makes the ontology contribution explicit and operationally aligned with the Neo4j-based semantic layer.

C. Semantic-pointer-driven storage and execution

The semantic layer is stored in Neo4j, whereas the data cubes are stored as Zarr and computed via xarray/dask. A semantic pointer $\pi(v)$ provides the minimal metadata needed to locate data at scale, including object-store URI, variables, spatial/temporal window, chunk keys, and access hints. This design cleanly decouples semantics from numerics: the KG determines *what to find* by resolving entities, constraints, and scope, while the STDC determines *what to compute* by loading only the chunks referenced by $\pi(S)$. Fig. 3 illustrates this two-stage workflow from semantic localization to parallel analytics.

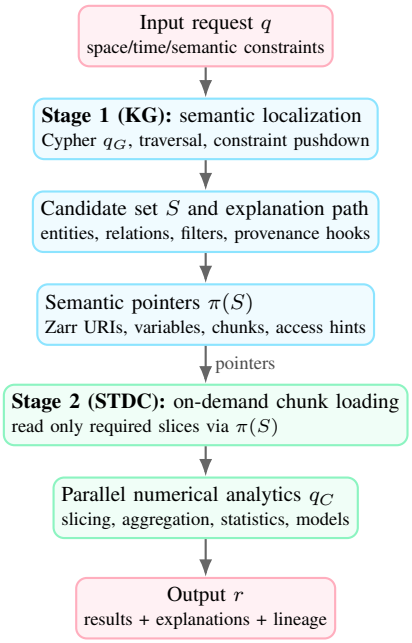

Fig. 3. Two-stage query execution: decoupling semantic localization and parallel analytics.

D. LLM+RAG natural-language interface

To support non-expert users, HydroFusion provides an agentic natural-language interface with two tools: a Cypher tool for graph queries and metadata retrieval, and a Python execution tool for reproducible analytics over xarray/dask. The RAG corpus contains three components: (i) an auto-generated KG schema snapshot, (ii) Python API signatures and docstrings for cube analytics (API anchoring), and (iii) curated few-shot exemplars. The corpus is re-indexed whenever the ontology or API surface changes, and the agent additionally queries the KG at runtime for current schema and candidate entities. We use constrained decoding (temperature $= 0.2$, top-p $= 1.0$), restrict generation to exposed tool APIs, and execute the resulting plan in a sandboxed runner. The workflow follows graph traversal $\rightarrow$ data extraction $\rightarrow$ statistical computing $\rightarrow$ result generation. Unless otherwise stated, the interface uses GPT-4-turbo; Qwen-72B-Chat is used for comparison, and Qwen-7B-Chat is used in the LLM-Small ablation.

IV. Prototype Implementation

HydroFusion is implemented in Python. The semantic layer uses Neo4j 5.11 (APOC 5.11) to store the ontology, entity relations, and pointer metadata, while the data layer stores large arrays as Zarr in MinIO (S3-compatible) and executes numerical workloads via xarray/dask with chunk-aware scheduling and parallel I/O. A query orchestrator decomposes each request into q_G and q_C, executes Cypher for semantic localization, resolves $\pi(S)$, and materializes a parallel analytics plan that loads only the required chunks. Lightweight provenance links (e.g., localization paths, pointers, and parameters) are recorded throughout execution for traceable and reproducible analyt-

ics. FastAPI exposes the service endpoints, and the natural-language interface uses the GPT-4-turbo API with comparison against locally deployed Qwen models.

V. Experiments and Case Study

A. Experimental setup

Hardware and software environment. Table I summarizes the experimental environment.

TABLE I
Experimental environment

Item	Details
CPU	Intel Xeon Gold 6248R (24 cores, 3.0 GHz)
Memory	256 GB DDR4-2933 ECC
Local cache disk	2 TB Samsung 980 PRO NVMe SSD (7000 MB/s read)
OS	Ubuntu 22.04.3 LTS (Kernel 5.15)
Graph DB	Neo4j 5.11 Community Edition (APOC 5.11)
Relational DB	PostgreSQL 15.3 + PostGIS 3.3.3
Object storage	MinIO (S3-compatible; RELEASE.2023-09-04T19-57-37Z)
Python stack	Python 3.10.12; xarray 2023.8.0; dask 2023.9.1
LLMs	GPT-4-turbo (API); Qwen-72B-Chat (local)

1) Datasets: We use four representative hydrological datasets (Table II): CAMELS-GB [13], ERA5-Land [14], USGS NWIS [15], and Landsat 8 ARD [16]. The resulting KG contains 128,456 entity nodes, 465,892 relation edges, and 892,341 attribute nodes, while the STDC totals 3.6 TB with 1,247 Zarr datasets. Chunks are stored in MinIO, with local NVMe used for indexing, caching, and intermediate results.
Subsetting and time windows. To keep the cube tractable while preserving cross-modal diversity, we construct a 2.8 TB ERA5-Land subset and an 850 GB Landsat subset. For ERA5-Land, we retain variables required by Q2/Q3/Q6 (precipitation, 2 m temperature, evaporation/ET proxies, and top-layer soil moisture), aggregate hourly fields to daily statistics, and spatially clip the archive to Great Britain (49–61°N, 8°W–2°E) and CONUS (24–50°N, 125°W–66°W). For Landsat 8 ARD, we use CONUS surface reflectance for 2013–2023, filter by cloud cover ($< 30\%$) and growing season (May–Sep), and build annual cloud-masked composites, from which we derive basin-aligned land-use and vegetation indicators (e.g., NDVI and land-cover change fractions) for Q5/Q6. Unless otherwise stated, benchmark queries target 2020, while the drought case study targets 2022. For meteorological drought characterization in the case study, we use the Standardized Precipitation Evapotranspiration Index (SPEI), a multiscalar drought index that jointly reflects precipitation deficits and atmospheric evaporative demand [17].

2) Baselines and workload: Table III summarizes the baseline systems and query workload. We design 10 variants for each query category (60 queries total). Each query is executed 100 times, with the first 10 runs used for warm-up; we report mean latency over the remaining 90 runs. P50/P95/P99 are also recorded but omitted for space. We use a 600 s timeout and count timeouts as failures. For throughput, we use a mixed workload (40% point lookup, 25% range extraction, 15% aggregation, 10% graph traversal, and 10% complex analytics) and vary client concurrency from 1 to 32. Zarr chunks are read from MinIO via an NVMe-backed local cache (LRU eviction); unless otherwise stated, latency experiments approximate cold-cache behavior by clearing the cache between query variants, whereas throughput and scale-out experiments report warm-cache steady state.

We summarize the main metrics as

$$\begin{aligned} \bar{L}(q) &= \frac{1}{n}\sum_{i=1}^{n} L_i(q), \\ \mathrm{QPS} &= \frac{N_{\mathrm{succ}}}{T_{\mathrm{wall}}}, \\ \mathrm{Speedup} &= \frac{\bar{L}_{\mathrm{base}}}{\bar{L}_{\mathrm{HF}}}. \end{aligned} \tag{8}$$

Q1/Q2 are retrieval and slicing tasks, Q3/Q4 involve aggregation and graph traversal, and Q5/Q6 represent complex cross-modal and multi-factor analytics.

a) Reporting policy for partially applicable baselines.: Because Q1–Q6 span metadata lookup, graph traversal, cross-modal association, and complex attribution, not all baselines are equally applicable to every query type. We therefore show only directly comparable systems in the main comparisons and explicitly note omitted systems where a baseline is unsupported or persistently times out.

B. Query performance

The following results focus on query execution (KG semantic localization + STDC parallel analytics), excluding LLM inference time for the natural-language interface.

1) Latency and throughput: HydroFusion consistently outperforms traditional baselines, especially on cross-modal association (Q5) and complex attribution (Q6), where it achieves 53.9× and 63.4× speedups over PostGIS (4.56 s and 4.93 s vs. 245.89 s and 312.45 s). For pure metadata discovery (Q1), HydroFusion is comparable to Neo4j-Only (0.21 s vs. 0.18 s), indicating that the main advantage comes from cross-layer analytics rather than standalone graph lookup. Under the mixed workload, HydroFusion scales to 523.7 QPS at 32-way concurrency (7.3× PostGIS).

Because Q1–Q6 span metadata lookup, graph traversal, cross-modal association, and complex attribution, not all baselines are equally applicable to every query type. MongoDB-GIS exhibits persistent timeouts on Q4–Q6 under the 600 s timeout setting, while Zarr-Only cannot answer Q4–Q6 because these tasks require relationship traversal beyond a data-cube-only baseline. Nevertheless, Zarr-Only remains a meaningful baseline for pure cube workloads: on Q2 and Q3, HydroFusion modestly outperforms it (2.15 s vs. 2.45 s; 6.73 s vs. 8.92 s) by pushing spatial filters and dataset selection to the semantic layer.

These gains come from decoupling localization from computation. HydroFusion first uses the KG to resolve semantic scopes (e.g., basin topology, station coverage, and provenance constraints) and shrink the search space to an explainable candidate set S. It then uses semantic pointers $\pi(S)$—including Zarr URIs, variables/dimensions, and chunk keys—to drive

TABLE II
DATASETS

Dataset	Data type	Spatial coverage	Temporal coverage	Size
CAMELS-GB	Basin attributes + daily streamflow	671 basins (Great Britain)	1970–2015	2.4 GB (110M records)
ERA5-Land	Gridded meteorology	Global ($0.1^\circ \times 0.1^\circ$)	1981–2023	48 TB (subset 2.8 TB)
USGS NWIS	Gauge time series	15,234 stations (United States)	1900–2023	18 GB (620M records)
Landsat 8 ARD	Remote-sensing imagery	CONUS	2013–2023	12 TB (sample 850 GB)

TABLE III
BASELINE SYSTEMS AND QUERY WORKLOAD

(a) Baseline systems

System	Stack	Notes
PostGIS	PostgreSQL + PostGIS	Relational spatiotemporal DB
Neo4j-Only	Neo4j + APOC	Graph-only baseline; numerical data stored as node properties
MongoDB-GIS	MongoDB + geospatial indexes	Document DB with GeoJSON/2dsphere
Virtuoso	Virtuoso + GeoSPARQL	RDF triple store with SPARQL 1.1 + GeoSPARQL
Zarr-Only	Zarr + xarray + dask	Data-cube-only baseline without semantics

(b) Query workload

ID	Type	Example
Q1	Metadata discovery	Find all gauges and observed variables in the Thames basin
Q2	Spatiotemporal extraction	Extract the daily precipitation series in 2020 for a target basin
Q3	Spatial aggregation	Compute annual mean evapotranspiration for basins in southern England
Q4	Multi-hop graph query	Find basins connected by upstream–downstream relationships
Q5	Cross-modal association	Link streamflow anomalies to upstream land-use changes
Q6	Complex attribution	Attribute drought events to meteorology, land surface, and human activities

minimal reads and parallel aggregation in the STDC, avoiding large intermediate joins. By contrast, monolithic relational or RDF plans often interleave joins and numerical operations, inflating intermediate results when constraints span multiple modalities.

TABLE IV
TAIL LATENCY AND TIMEOUTS (TEMPLATE).

System	P50 (s)	P95 (s)	P99 (s)	Timeout (%)
HydroFusion	**TBD**	**TBD**	**TBD**	**TBD**
PostGIS	**TBD**	**TBD**	**TBD**	**TBD**
Neo4j-Only	**TBD**	**TBD**	**TBD**	**TBD**
Virtuoso	**TBD**	**TBD**	**TBD**	**TBD**

C. Additional evaluation

In addition to the main results on latency, throughput, scalability, and interaction accuracy, we conduct complementary analyses to further characterize the practical behavior of HydroFusion. These analyses cover end-to-end execution breakdown and I/O footprint, ingestion and storage cost, sensitivity to chunking, caching, and parallelism, tail latency and timeout behavior, provenance overhead, and end-to-end natural-language interaction cost and robustness. Detailed results are summarized in Section VI.

VI. Additional Results and Analysis

A. End-to-end breakdown and I/O characterization

To make the performance gains interpretable, we instrument HydroFusion to report stage-level latency and I/O footprint for each request. The breakdown follows the two-stage execution model, including KG localization, pointer resolution, object-store reads, numerical computing, and result serialization. We also record bytes read, chunks fetched, and cache hit rate under cold- and warm-cache settings. Figure 4 summarizes the latency breakdown and I/O footprint.

B. Ingestion and storage cost

We also measure the one-off and recurring costs of constructing and maintaining the system, including ingestion, indexing, and storage overhead for numerical payloads, semantic stores, and auxiliary metadata.

C. Parameter sensitivity

We analyze sensitivity to Zarr chunk size/layout, NVMe cache capacity, and worker parallelism using representative workloads (Q2/Q3/Q6).

D. Stability and tail latency

We report P50/P95/P99 and timeout rate under the same mixed workload used in throughput evaluation.

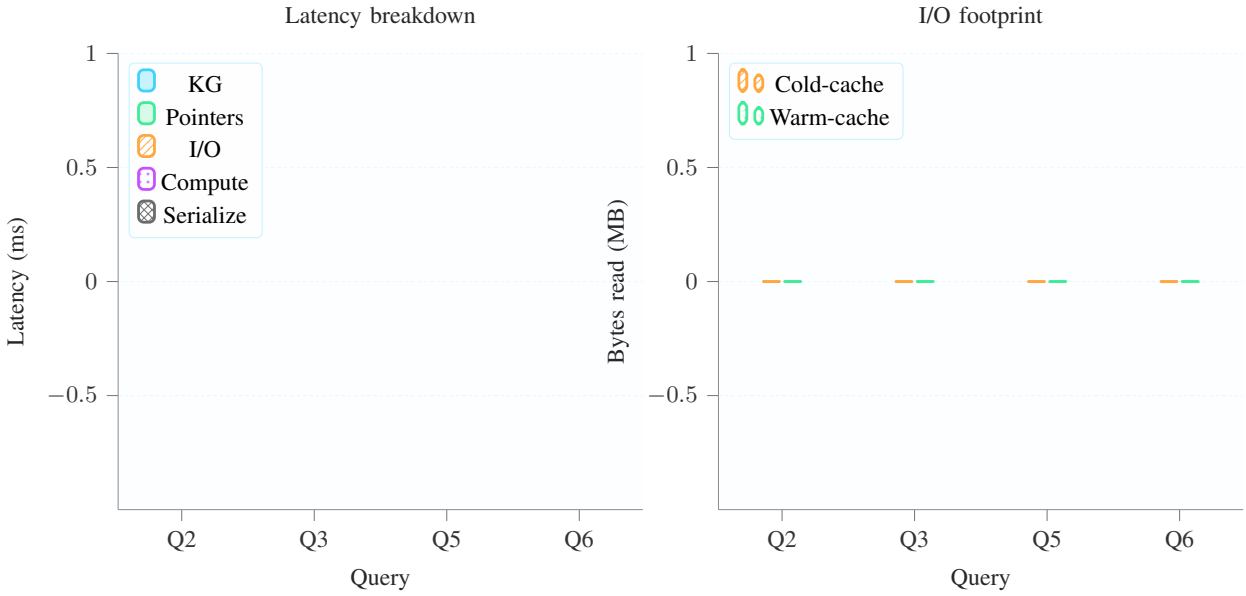

Fig. 4. End-to-end breakdown and I/O footprint (template).

TABLE V
END-TO-END NL INTERACTION COST AND ROBUSTNESS (TEMPLATE).

Setting	Acc. (%)	E2E lat. (s)	Tokens (in/out)	Avg. turns
No RAG, 1-turn	**TBD**	**TBD**	**TBD**	**TBD**
Dyn. schema+API, 1-turn	**TBD**	**TBD**	**TBD**	**TBD**
Dyn. schema+API, clarification	**TBD**	**TBD**	**TBD**	**TBD**
Full RAG+few-shot, 1-turn	**TBD**	**TBD**	**TBD**	**TBD**

E. End-to-end NL cost and robustness

We report full end-to-end latency for the natural-language interface, including retrieval, model inference, backend execution, and response generation, together with token usage and robustness under ambiguity.

These measurements complement the main results by explaining the speedups through reduced data movement and selective chunk access, while also characterizing system stability and end-to-end NL interaction cost.

F. Ablation study

To quantify the role of each component, we evaluate seven variants on Q5 and Q6, repeating each configuration 30 times and reporting mean±std with paired t-tests ($\alpha = 0.05$). End-to-end NL accuracy is defined by the correctness of both the executable plan and the final answer; variants without an LLM are marked as N/A. Variants include Full, w/o KG, w/o STDC, w/o SP, KG-Lite, w/o LLM, and LLM-Small. For reproducibility, LLM-Small uses the same prompt template, RAG contexts, tool APIs, and constrained decoding settings (temperature $= 0.2$, top-$p = 1.0$) as the full system, replacing only GPT-4-turbo with Qwen-7B-Chat.

Removing the KG increases Q6 latency from 4.93 s to 145.67 s (29.5×), highlighting the importance of semantic localization and relationship traversal. Removing the STDC raises Q6 latency to 234.89 s and peak memory from 18.4 GB to 45.6 GB, reintroducing I/O and materialization bottlenecks. Removing semantic pointers increases Q6 latency to 34.56 s, indicating substantial cross-layer matching overhead. KG-Lite reduces NL accuracy to 82.1%, while LLM-Small further lowers it to 62.5%, suggesting that ontology expressiveness and model capacity mainly affect the interaction layer rather than backend execution.

G. Scalability and natural-language interface evaluation

Fig. 6 summarizes scalability. For scale-up, we expand the base KG/STDC while preserving schema and chunk layout, increasing from 128K/360 GB to 6.4M/18 TB; Q6 latency grows sublinearly (4.93 s → 34.56 s at 50×). For scale-out, we deploy HydroFusion on a 16-node Kubernetes cluster and scale the data-layer analytics (dask workers) from 1 to 16 while keeping the KG store centralized; throughput reaches 498.7 QPS at 16 workers (10.9× over 1 worker) with 68.4% parallel efficiency:

$$\eta(p) = \frac{\mathrm{QPS}(p)}{p\,\mathrm{QPS}(1)} \times 100\%. \tag{9}$$

Sublinear scale-up is enabled by graph indexing, selective Zarr chunk access, and dask lazy evaluation, whereas scale-out efficiency declines at larger worker counts because of shared object-store bandwidth, scheduler overhead, and the centralized KG store.

Fig. 7 reports pilot evaluation of the natural-language interface on 45 expert-authored questions. HydroFusion achieves 80% end-to-end accuracy overall, with 100% on simple lookup, 80% on aggregation, and 60% on complex analysis. These results should be interpreted as an initial feasibility study rather than a comprehensive robustness characterization.

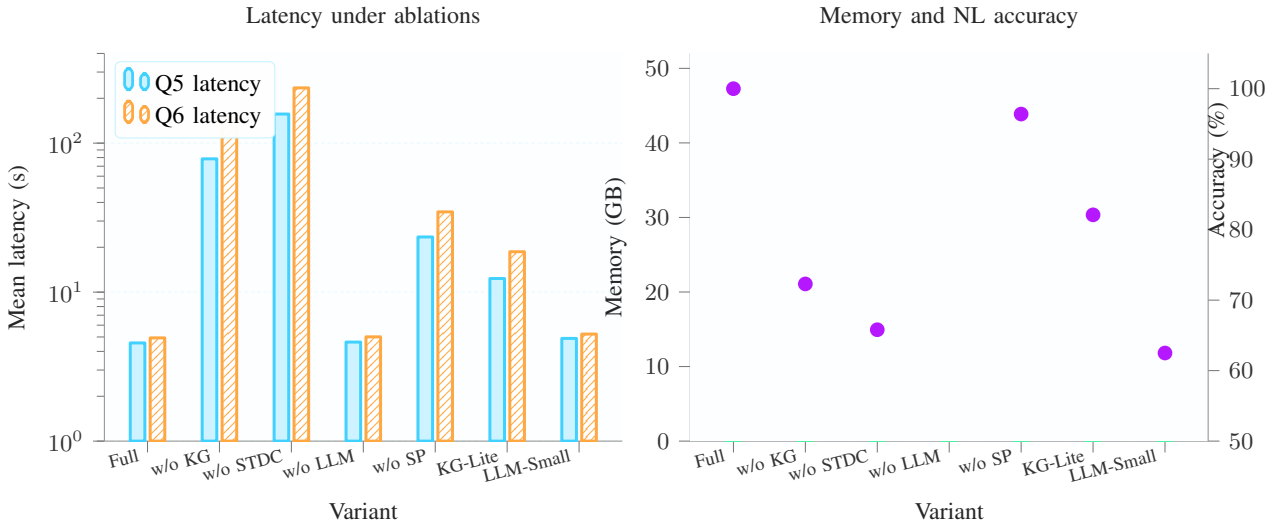

Fig. 5. Ablation results. Left: Q5/Q6 latency across variants (log scale). Right: peak memory (bars) and NL end-to-end accuracy (points); w/o LLM has no NL accuracy.

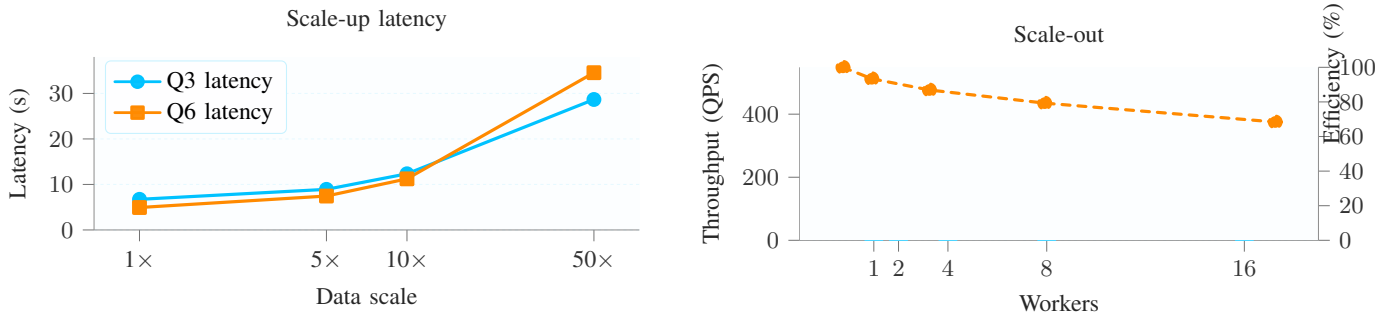

Fig. 6. Scalability of HydroFusion.

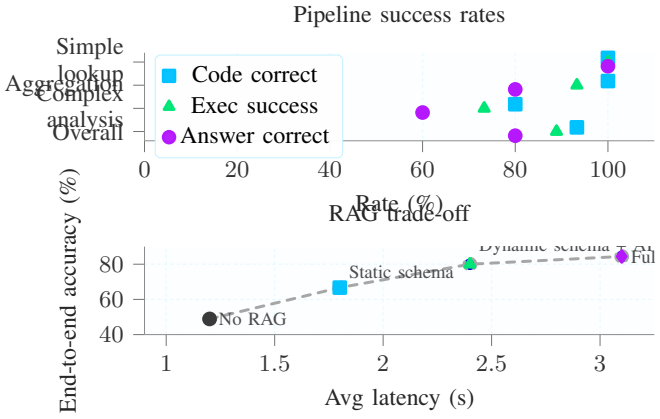

Fig. 7. Pilot NL evaluation. Left: code, execution, and answer correctness. Right: accuracy–latency trade-off across RAG settings.

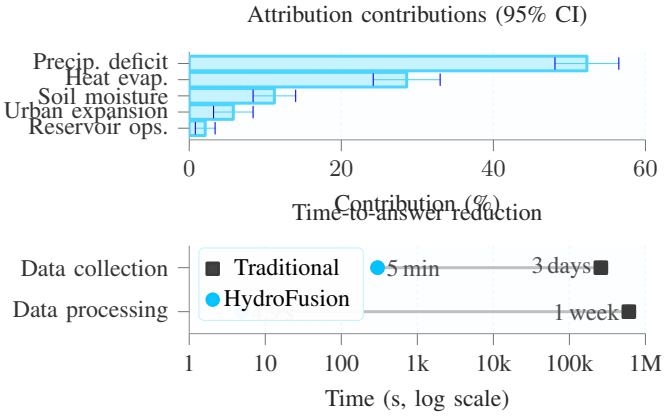

Fig. 8. Drought attribution case study: factor contributions and time-to-answer reduction.

Within this setting, dynamic schema + API anchoring provides the best balance (80.0% at 2.4 s) compared with no RAG (48.9% at 1.2 s), while full RAG + few-shot further improves accuracy at higher latency. End-to-end correctness is defined as

$$\mathrm{Acc}_{\mathrm{e2e}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[c_i \wedge e_i \wedge a_i]. \tag{10}$$

Error analysis reveals three main failure modes: 3 code-

generation errors, 2 execution failures, and 4 answer deviations concentrated in complex analysis. The last category mainly reflects error accumulation across schema/entity grounding, underspecified temporal or spatial constraints, and multi-step reasoning after intermediate retrieval or computation steps.

H. Discussion

HydroFusion is most beneficial for workflows requiring both semantic reasoning (e.g., basin topology, multi-hop relationships, provenance constraints) and large-scale numerical analytics (e.g., gridded meteorology, remote sensing). In these scenarios, the two-stage design reduces repeated scans and join blow-ups by using the KG to constrain scope and the STDC to compute only on the addressed chunks. In contrast, for pure metadata discovery (Q1), performance gains are modest since it is dominated by standalone graph lookup.

VII. Conclusion and Future Work

HydroFusion fuses a Knowledge Graph and a Spatiotemporal Data Cube to unify semantic reasoning with high-throughput numerical analytics for hydrological data. Semantic pointers and two-stage execution enable efficient cross-modal queries, while experiments demonstrate strong gains on complex analytics, good scalability, and practical value in a drought attribution case study. The LLM+RAG interface further improves accessibility for non-expert users, although its reliability remains lower on complex multi-step analysis than on constrained lookup tasks.

Future work will focus on streaming ingestion and incremental ontology updates, more robust NL interaction with clarification and plan verification, tighter coupling with hydrological models, and distributed graph sharding and query planning for improved horizontal scalability.

References

[1] P. Yue, Z. Cao, B. Shangguan, S. Zhao, L. Jiang, L. Hu, and Z. Fang, "A multi-source spatio-temporal data cube for large-scale geospatial analysis," *International Journal of Geographical Information Science*, vol. 36, no. 9, pp. 1853–1884, 2022.

[2] N. J. Car and T. Homburg, "Geosparql 1.1: Motivations, details and applications of the decadal update to the most important geospatial lod standard," *ISPRS International Journal of Geo-Information*, vol. 11, no. 2, p. 117, 2022.

[3] K. Janowicz, P. Hitzler, W. Li, D. Rehberger, M. Schildhauer, R. Zhu, C. Shimizu, C. K. Fisher, L. Cai, G. Mai *et al.*, "Know, know where, knowwheregraph: A densely connected, cross-domain knowledge graph and geo-enrichment service stack for applications in environmental intelligence," *AI Magazine*, vol. 43, no. 1, pp. 30–39, 2022.

[4] A. Dsouza, N. Tempelmeier, R. Yu, S. Gottschalk, and E. Demidova, "Worldkg: A world-scale geographic knowledge graph," in *Proceedings of the 30th ACM international conference on information & knowledge management*, 2021, pp. 4475–4484.

[5] X. Ma, "Knowledge graph construction and application in geosciences: A review," *Computers & Geosciences*, vol. 161, p. 105082, 2022.

[6] J. D. Rondón Díaz and L. M. Vilches-Blázquez, "Characterizing water quality datasets through multi-dimensional knowledge graphs: A case study of the bogota river basin," *Journal of Hydroinformatics*, vol. 24, no. 2, pp. 295–314, 2022.

[7] O. Baydaroglu, K. Kocak, Z. Duran, and I. Demir, "A comprehensive review of ontologies in the hydrology towards guiding next generation artificial intelligence applications," *Journal of Environmental Informatics*, vol. 42, no. 2, pp. 90–107, 2023.

[8] A. S. Lippolis, G. Lodi, and A. G. Nuzzolese, "The water health open knowledge graph," *Scientific Data*, vol. 12, no. 1, p. 274, 2025.

[9] P. Yue, M. Zhang, X. Zhai, J. Gong, and L. Jiang, "Model provenance tracking and inference for integrated environmental modelling," *Environmental Modelling & Software*, vol. 103, pp. 120–133, 2018.

[10] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman, Z. Zhang, and D. Radev, "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*, 2018, pp. 3911–3921.

[11] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 9459–9474.

[12] J. Liang, S. Hou, H. Jiao, Y. Qing, A. Zhao, Z. Shen, L. Xiang, and H. Wu, "Geographrag: A graph-based retrieval-augmented generation approach for empowering large language models in automated geospatial modeling," *International Journal of Applied Earth Observation and Geoinformation*, vol. 142, p. 104712, 2025.

[13] G. Coxon, N. Addor, J. P. Bloomfield, J. Freer, M. Fry, J. Hannaford, N. J. K. Howden, R. Lane, M. Lewis, E. L. Robinson, T. Wagener, and R. Woods, "CAMELS-GB: Hydrometeorological time series and landscape attributes for 671 catchments in great britain," *Earth System Science Data*, vol. 12, pp. 2459–2483, 2020.

[14] J. Muñoz-Sabater, E. Dutra, A. Agustí-Panareda, C. Albergel, G. Arduini, G. Balsamo *et al.*, "ERA5-Land: A state-of-the-art global reanalysis dataset for land applications," *Earth System Science Data*, vol. 13, pp. 4349–4383, 2021.

[15] U.S. Geological Survey, "National water information system (NWIS)," https://waterdata.usgs.gov/nwis, 2026, accessed: Jan. 2026.

[16] D. P. Roy, M. A. Wulder, T. R. Loveland, C. E. Woodcock, R. G. Allen, M. C. Anderson *et al.*, "Landsat-8: Science and product vision for terrestrial global change research," *Remote Sensing of Environment*, vol. 145, pp. 154–172, 2014.

[17] S. M. Vicente-Serrano, S. Beguería, and J. I. López-Moreno, "A multiscalar drought index sensitive to global warming: The standardized precipitation evapotranspiration index," *Journal of Climate*, vol. 23, no. 7, pp. 1696–1718, 2010.