Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

The S2 Hierarchical Discrete Global Grid as a Nexus for Data Representation, Integration, and Querying Across Geospatial Knowledge Graphs

Shirly Stephen[1] | Mitchell Faulk[1] | Krzysztof Janowicz[2,1] | Colby Fisher[3] | Thomas Thelen[1] | Rui Zhu[4] | Pascal Hitzler[5] | Cogan Shimizu[6] | Kitty Currier[1] | Mark Schildhauer[1] | Dean Rehberger[7] | Zhangyu Wang[1] | Antrea Christou[6]

[1]NCEAS | Department of Geography, University of California, Santa Barbara, California, USA

[2]Geoinformatics, University of Vienna, Vienna, Austria

[3]Hydronos Labs, New Jersey, USA

[4]School of Geographical Sciences, University of Bristol, UK

[5]Department of Computer Science, Kansas State University, Kansas, USA

[6]Department of Computer Science & Engineering, Wright State University, Ohio, USA

[7]Department of History, Michigan State University, Michigan, Country Name

Correspondence
Shirly Stephen, University of California, 1832 Ellison Hall, Santa Barbara, CA, USA.
Email: shirlystephen@ucsb.edu

Abstract

Geospatial Knowledge Graphs (GeoKGs) have become integral to the growing field of Geospatial AI. Initiatives like the U.S. National Science Foundations Open Knowledge Network program aim to create an ecosystem of nation-scale, cross-disciplinary GeoKGs that provide AI-ready geospatial data aligned with FAIR principles. However, building this infrastructure presents key challenges, including 1) managing large volumes of data, 2) the computational complexity of discovering topological relations via SPARQL, and 3) conflating multi-scale raster and vector data. Discrete Global Grid Systems (DGGS) help tackle these issues by offering efficient strategies for data integration and representation. The KnowWhereGraph (KWG) utilizes Google's S2 Geometry — a DGGS framework — to enable efficient multi-source data processing, qualitative spatial querying, and cross-graph integration. This paper outlines the implementation of S2 within KWG, emphasizing its role in topologically enriching and semantically compressing data. Ultimately, this work demonstrates the potential of DGGS frameworks, particularly S2, for building scalable GeoKGs.

KEYWORDS
geospatial knowledge graphs, discrete global grid systems, s2 geometry, linked data, query optimization, spatial ontologies

1 | INTRODUCTION

Knowledge Graphs (KGs), originating from Semantic Web research (Hitzler, 2021), have revolutionized the data science landscape by facilitating the integration and enrichment of vast, multi-source, and heterogeneous data while empowering AI applications to deliver high-quality, trusted insights for scientific decision-making (Ji, Pan, Cambria, Marttinen, & Philip, 2021; Abu-Salih, 2021). To support translational research — bridging the gap between fundamental research and real-world applications — initiatives like the U.S. National Science Foundations Open Knowledge Network (OKN) program[1] and the Linked Open Data (LOD) Cloud[2] are driving the development of comprehensive ecosystems of nation-scale KGs that adhere to FAIR (Findable, Accessible, Interoperable, Reusable) data principles (Wilkinson et al., 2016). These initiatives aim to create data-centric infrastructures for addressing complex cross-disciplinary challenges, ranging from climate change to economic equity. A central theme interlinking the various domain KGs in these infrastructures is the geospatial layer comprised of Geospatial Knowledge Graphs (GeoKGs), such as the KnowWhereGraph (Janowicz et al., 2022) and Urban Flooding Open Knowledge Network (UF-OKN) (Johnson et al., 2022). GeoKGs are now a critical component of modern Geospatial Artificial Intelligence

[1] https://www.proto-okn.net/
[2] https://lod-cloud.net/

(GeoAI), providing a structured and interconnected approach to managing and analyzing extensive spatial datasets from heterogeneous sources (Janowicz et al., 2022; Yan, Mai, Hu, & Janowicz, 2020). The Resource Description Framework (RDF) (Manola, Miller, McBride, et al., 2004) is the widely adopted paradigm for GeoKGs due to its semantic richness, high interoperability, and adherence to open standards. Additionally, the Open Geospatial Consortium's (OGC) GeoSPARQL standard (N. J. Car & Homburg, 2022) enhances the representation of spatial information in RDF, enabling quantitative and qualitative spatial querying. This capability of GeoKGs is leveraged in AI systems to provide more intelligent, multi-faceted, and context-aware services (Janowicz et al., 2023).

Despite the promise of GeoKGs, three primary challenges impede their effective implementation: handling and storing immense volumes of (geospatial) data, the computational complexity involved in large-scale spatial querying, and analyzing raster and vector data together at various spatial scales. Graph databases capable of managing large datasets (e.g., Neptune, Neo4j) lack any GeoSPARQL support. GeoSPARQL-compliant RDF databases (Jovanovik, Homburg, & Spasić, 2021) like Blazegraph[3], GraphDB[4], Virtuoso[5], and Stardog[6] use indexing methods such as R-trees, quadtrees, and geohashing. However, these methods are not optimized for cross-scale analyses, global datasets, and dense graphs with high degrees of spatial overlap (Huang, Raza, Mirzov, & Harrie, 2019; Theocharidis, Liagouris, Mamoulis, Bouros, & Terrovitis, 2019). High memory costs from building and maintaining extensive indexes, computational costs of traversing deep R-trees and quadtrees (Kothuri, Ravada, & Abugov, 2002), and inefficiencies in proximity searches due to boundary ambiguities in geohashing (Acharya, 2023) are just some challenges with large graphs. While these indexing methods can accelerate fundamental spatial filtering queries such as point-based queries, containment queries, and spatial joins for range and nearest-neighbor queries, complex queries involving spatial refinements, such as polygon intersection, spatial cross-matching, and identifying spatial patterns, remain highly expensive. For example, queries like "*Find the regions where earthquakes occurred in 2023*" or "*Find comparable areas that are earthquake-prone*" require a large number of spatial joins to determine relevancy, and can be inefficient or even return incorrect results (Huang et al., 2019). Experimental evaluations show that topological computation becomes highly expensive or nearly intractable in GeoKGs characterized by extensive data volume, uneven data scale, and diverse categories of data (Theocharidis et al., 2019; W. Li, Wang, Wu, Gu, & Tian, 2022). Furthermore, the qualitative spatial functions that GeoSPARQL provides are insufficient for multi-scale analytics (Manley, 2021). For instance, conflating a vector region dataset with a map scale of 1:50,000 alongside a land-use raster dataset of 30 m resolution can be problematic (Davis et al., 1991). Addressing these challenges requires innovative data integration and management strategies that can efficiently handle large, multi-scale, multi-format geospatial data and support complex topological queries in the context of GeoKGs.

Discrete Global Grid Systems (DGGS) are spatial indexing frameworks that tessellate the Earth's surface into hierarchical, regular grids (Sahr, White, & Kimerling, 2003). Unlike traditional GIS systems that use latitude and longitude coordinates, a DGGS employs cell identifiers (cell IDs) to uniquely reference discrete portions of the Earth's surface. DGGS frameworks have been utilized in geospatial applications since the early 1990s (M. Goodchild, 1994), but have more recently emerged as a key technology for next-generation GIS (M. Li & Stefanakis, 2020; Hojati, Robertson, Roberts, & Chaudhuri, 2022; Bondaruk, Roberts, & Robertson, 2020) and Digital Earth systems (Mahdavi-Amiri, Alderson, & Samavati, 2015; Robertson, Chaudhuri, Hojati, & Roberts, 2020; Yao et al., 2019; Hojati et al., 2022). This is primarily driven by the standardization of DGGS configurations by the OGC in 2017 (OGC, 2017) that spurred the proliferation of DGGS implementations and their open-source software libraries, such as Google's S2 Geometry (Veach, Rosenstock, Engle, & Manshreck, 2017), Uber's H3 (Uber, 2023), rHEalPIX (Gibb, 2016), and OpenEAGGR (Riskaware Ltd, 2017). These libraries provide (quantization) functions to assign geometries to cells, and spatial operations for cell navigation (M. Li & Stefanakis, 2020). Their algorithms leverage cell IDs as well as their inherent inter-relations to replace complex spatial computations with simple coding operations, making DGGS frameworks powerful tools for efficient data integration, retrieval, and reduction of analytical complexity across various geospatial application domains (M. Li, McGrath, & Stefanakis, 2021; Kranstauber, Weinzierl, Wikelski, & Safi, 2015; Romanov & Khvostov, 2018; Pereira, Herszenhut, Saraiva, & Farber, 2024; Lin, Zhou, Xu, Zhu, & Lu, 2018). A seminal paper by Goodchild in 2018 (M. F. Goodchild, 2018) highlighted the benefits of DGGS for multi-scale, multi-format data integration within linked data frameworks, sparking interest in the GeoKG and OKN communities (Bastrakova & Crossman, 2020). Despite its potential, the adoption of DGGS in GeoKGs remains limited. One reason is the lack of support in graph databases for DGGS library interfaces or full support for gridded representations of geospatial data. While GeoSPARQL 1.1 (N. J. C. Car et al., 2023) introduces a generic format for DGGS geometry serialization (N. J. Car & Homburg, 2022), it does

[3] https://blazegraph.com/
[4] https://graphdb.ontotext.com/
[5] https://virtuoso.openlinksw.com/
[6] https://www.stardog.com/

not yet provide the ability to interpret specific cell IDs according to the various DGGS frameworks available. Furthermore, testing this implementation in RDF databases is still in its early stages (Habgood, Homburg, Car, & Jovanovik, 2022). Few graph databases support DGGS-based indexing, such as those using S2 and H3 (e.g., NebulaGraphDB), but these are not RDF-based or GeoSPARQL-compliant (Jovanovik et al., 2021). The commercial Foursquare GeoKG (Gundeti, 2023) utilizes H3 for location-specific business use cases, but only as an indexing framework. More sophisticated usages of DGGS in GeoKGs, such as for spatial data compression through semantic compression have been postulated (Zalewski, Hitzler, & Janowicz, 2021) but not practically explored until KnowWhereGraph.

The KnowWhereGraph (KWG) (Janowicz et al., 2022), a GeoKG within the OKN framework, employs the S2 Geometry to spatially integrate large-scale geospatial data by optimizing their storage, processing, visualization, and analysis across various data types, shapes, scales, and precision. KWG transforms siloed heterogeneous location and environmental observation data into actionable insights for environmental intelligence by *enhancing cross-domain data interoperability*, *contextualizing data*, and *generating AI-ready geospatial data*. Rather than relying on S2 as an implicit indexing method in GraphDB (the graph database where KWG is deployed), KWG represents S2 cells as a set of nodes within an S2-RDF graph. GeoSPARQL's Dimensionally Extended 9-Intersection Model (DE-9IM) relations (Egenhofer, Mark, & Herring, 1994) are employed to model cell navigation (parent, child, neighbor) and topological connections between S2 cells and other geographic features. Integrating S2 as a spatial reference framework within KWG provides a robust infrastructure that supports cross-domain data linkages and abstracts spatial analytics while fostering a rich environment for modeling and computation of geographical information.

This paper details this implementation by outlining the development of the S2-RDF Graph and describing the following two methods to quantize spatial data onto S2 cells.

1. *Topological enrichment of cells* involves materializing topological relations (based on DE-9IM) between S2 cells and geographic features having explicit geometries in the graph. This approach mitigates the tractability challenges of calculating these relationships at runtime via GeoSPARQL, even when utilizing efficient implicit spatial indexing methods.
2. *Grid-based data discretization* involves transforming non-DGGS data (both vector and raster) using techniques like statistical aggregation, decomposition, and spatial overlay to represent them as S2-cell-based gridded data. This process enables geometric simplification, scalable data representation, and integrated raster/vector analysis. In this framework, the S2 cells serve as the spatial *FeatureOfInterest* (Janowicz, Haller, Cox, Le Phuoc, & Lefrançois, 2019), linking various thematic observations.

The significant spatial computational capabilities of the S2 Geometry library (Veach et al., 2017) are utilized in a Python-based RDF processing environment to materialize statements for enrichment and discretization. Through topological enrichment, S2 cells are tagged with spatial features, thereby associating cells with observations linked to those features, e.g., S2 cells are linked to climate observations via topologically connected climate division features. Through discretization, S2 cells are enriched with temporally-scoped observations computed for the specific cell area, e.g., the area of a crop type within a cell. Once the S2 grid is augmented with each dataset using these methods, each cell node can be queried as a set of individual objects containing diverse pieces of information that can then be *spatially fused* or *conflated* for analysis. This approach is powerful because once a dataset is quantized on the S2 grid, it eliminates the need for repeated spatial analytics and related GIS processes to integrate and compare disparate datasets. Our experience indicates that leveraging S2 in KWG has enabled efficient multi-source data processing and faster qualitative spatial querying while constructing an enriched GeoKG that augments graph embeddings (Iliakis, 2022) and cross-disciplinary data integration within the OKN framework (Section. 5.3).

This paper aims to provide guidelines and examples for implementing and utilizing S2 in GeoKGs, highlighting its analytical advantages. By doing so, it seeks to advance GeoAI and offer a scalable solution for managing and analyzing large-scale geospatial data within GeoKGs.

The rest of the paper is structured as follows: Section 2 provides background on GeoKGs and DGGS frameworks, highlighting KWG and S2 Geometry. Section 3 introduces the workflow for modeling and ingesting S2 data into KWG, along with the technical strategy for quantization. Section 4 discusses the implementation of the two quantization methods in KWG. Section 5 evaluates the performance and benefits of S2 integration through various use cases, demonstrating improvements in data processing, query efficiency, and scalability. Finally, Section 6 discusses the broader implications of using DGGS, particularly S2 Geometry, within GeoKGs, including achieved benefits, current limitations, and areas for future research.

2 | THEORETICAL BACKGROUND AND RELATED RESEARCH

2.1 | Geospatial Knowledge Graphs

Geospatial knowledge graphs (GeoKGs) emerged in the late 2000s as open data ecosystems, primarily aiming to democratize data access in alignment with FAIR (Findable, Accessible, Interoperable, Reusable) data principles (Wilkinson et al., 2016). Early graphs such as GeoNames[7] and LinkedGeoData[8] predominantly comprised *explicit geographic* entities, such as named places, natural features, landmarks, roads, and historical features, with point geometries and limited semantics. In recent years, GeoKGs have expanded in scope to model *geographically themed* data across many domains, such as from environmental (Janowicz et al., 2022; Zhu, Stephen, et al., 2021), public health (Jiang et al., 2020; Zhu, Janowicz, Mai, Cai, & Shi, 2022), disaster management (Janowicz et al., 2022), transportation (Böckling, Paulheim, & Detzler, 2024; Qi, Mai, Zhu, & Zhang, 2023), production logistics (Zhao, Zhang, Chen, Qu, & Huang, 2022), and gaming domains (Jiang et al., 2020) demonstrating the versatility and impact of GeoKGs in various application areas. These GeoKGs play a pivotal role in cross-domain geospatial analytics (Janowicz et al., 2022), geospatial question answering (Mai, Janowicz, Zhu, Cai, & Lao, 2021; Weinberger, Scholz, & Wandl-Vogt, 2022), and geo-visualization (Balla et al., 2020; W. Li et al., 2023). Moreover, their rich contextual information makes them essential training data for spatial models, facilitating the development of spatial and temporal reasoning (Zhu, Janowicz, Cai, & Mai, 2022), predictive analytics, and decision support systems (Zhu, Janowicz, Cai, & Mai, 2022; Mai et al., 2022). Examples of such rich GeoKGs in the OKN ecosystem (NSF, November 2018) include the KWG[9] that focuses on environmental intelligence (Janowicz et al., 2022); the UF-OKN[10] for flood prediction, response, mitigation, and prevention (Saksena, 2022); and the Safe Agricultural Products and Water Graph[11] (SAWGraph) to understand PFAS contamination in food and water systems.

A. Preliminaries of the RDF Data Model: The structure and principles of RDF directly influence how S2 is used to synthesize, integrate, and query data within KWG, as will be discussed later in Section 4. To facilitate understanding, this section briefly introduces the basic components and structure of RDF and SPARQL. For a more comprehensive introduction to RDF and SPARQL, refer to (Manola et al., 2004; Hitzler, Krötzsch, & Rudolph, 2010; World Wide Web Consortium and others, 2013).

Definition 1. *RDF triple*: An RDF triple $t(s, p, o)$ is the basic atomic entity of a knowledge graph. It expresses a single statement about semantic data. The subject (s) and object (o) are considered graph vertices referring to a resource or a simple value (called literal), and the property (p) (a.k.a. predicate) is the directed edge that connects the two vertices. The *resources* in $t(s, p, o)$ are given uniform resource identifiers (URI) or are blank nodes (denoting an unknown resource).

Definition 2. *RDF graph*: An RDF graph is a collection of RDF triples. Such a graph can essentially be viewed as a directed graph $G = (V, E)$ where V denotes the set of vertices that can be resources or literals, and E denotes the set of directed edges.

Definition 3. *SPARQL query*: A SPARQL query Q is defined as tuple $Q = (E, G, R)$. E denotes the algebraic expression built from graph patterns and solution modifiers. G is the RDF graph being queried. R denotes the result form, such as SELECT, CONSTRUCT, DESCRIBE, or ASK, which specifies how the query results are processed and presented. The algebraic expression E can include various graph patterns and solution modifiers like PROJECTION, DISTINCT, LIMIT, or ORDER BY. The simplest graph pattern in SPARQL is the triple pattern, which matches triples in the RDF graph.

B. GeoSPARQL: In the RDF domain, OGC's GeoSPARQL standard (N. J. Car & Homburg, 2022) is the leading specification for representing and querying geospatial data. It is implemented as an extension by many RDF graph databases (Jovanovik et al., 2021). GeoSPARQL leverages OGCs Simple Features ontology[12] for defining spatial entities as shown in Figure. 1. The *SpatialObject* class includes any resource that can have a spatial representation. The *Feature* class specifically represents spatial objects with concrete geographical shapes, linked to their geometries (serialized as GML or WKT) via the *hasGeometry* property. GeoSPARQL supports qualitative relations such as containment and overlap for evaluating topological relationships between entities, with *spatialRelation* subsuming specific properties. Additionally, the specification provides properties for non-topological spatial operations (e.g., distance, buffer) and spatial aggregation functions.

[7] https://www.geonames.org/
[8] http://linkedgeodata.org/
[9] https://knowwheregraph.org/
[10] https://ufokn.com/
[11] https://sawgraph.github.io/
[12] https://www.ogc.org/standard/sfa/

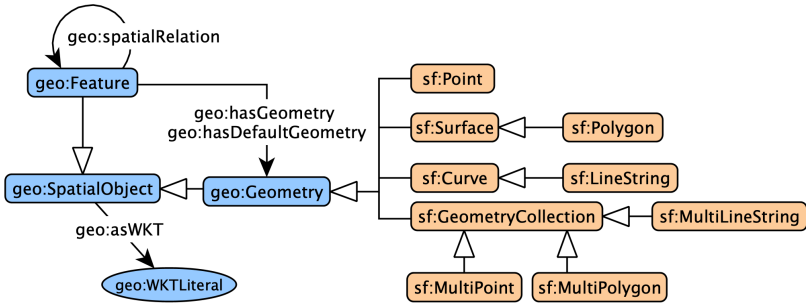


FIGURE 1 GeoSPARL's core concepts are indicated using blue boxes. Geometry classes from Simple Features are indicated using beige boxes. White headed arrows indicate *rdfs*:*subClass* relations and filled arrows indicate object/data properties.

C. Spatial Query Challenges in GeoKGs: The size of GeoKGs can proliferate when dealing with a multitude of complex geometries, such as multi-polygons and polylines. For example, the geometric representation of Greece (from the GADM dataset) is a multi-polygon that is a collection of 695 polygons having approximately **29,900** nodes, resulting in a WKT (Well-Known Text) representation that is 9MB large. High precision can adversely affect the graph's performance in terms of storage, indexing, and query efficiency (Regalia, Janowicz, & McKenzie, 2017), while reducing precision can impact the quality of spatial querying. Algorithms determining topological relations using geometries operate with polynomial time complexity relative to the number of nodes in the geometries being tested (Rigaux, Scholl, & Voisard, 2002). Graph databases that implement popular space-dividing indexing mechanisms, such as quadtrees and geohashing (e.g., GraphDB), still encounter limitations in spatially pruning search spaces. Geohashing can split regions of interest across multiple cells, leading to boundary issues where a single spatial entity is divided by cell boundaries. This affects the accuracy and efficiency of spatial queries, as additional computation is needed to merge or handle entities spanning multiple geohashes (Papadias, Zhang, Mamoulis, & Tao, 2003). This is particularly problematic for linear features (e.g., rivers) or extensive areas (e.g., country boundaries). Each geohash string is independent, lacking a hierarchical relationship between different levels of resolution, and they are not ordered based on spatial proximity. This non-intuitive ordering complicates range queries and neighborhood searches, especially for features such as political borders that often span multiple geohash cells (Acharya, 2023). Quadtrees provide adaptive resolution indexing capabilities and improve query performance by reducing the number of nodes that need to be inspected. However, high-resolution data can result in very deep quadtree structures, which can lead to performance bottlenecks, particularly in large-scale graphs (Dinkins, 2023). Experimental evaluations have also shown that spatial joins for complex geometries are not optimized in GeoSPARQL-compliant RDF databases (Huang et al., 2019; Ioannidis, Garbis, Kyzirakos, Bereta, & Koubarakis, 2021; W. Li et al., 2022). These size-related query challenges of GeoKGs re-emphasize prior statements (Regalia, Janowicz, & McKenzie, 2019) that storing accurate representations of geospatial data in their original vector or raster format, including complex geometries as RDF literals, while reasonable, may not be suitable for storage and query efficiency.

2.2 | The KnowWhereGraph

Envisioned as a *gazetteer of gazetteers* that links heterogeneous data via joint place identifiers, KnowWhereGraph (KWG) empowers decision-makers with on-demand and comprehensive location insights derived from a diverse range of data. By minimizing data processing overhead, KWG provides semantically rich, contextualized, and analysis-ready data for environmental intelligence analytics, with a particular focus on two key use cases. The humanitarian relief use case (Zhu, Cai, et al., 2021) is tailored to aid humanitarian organizations in swiftly identifying and mobilizing relevant personnel with suitable expertise to respond effectively to imminent or ongoing disasters. The farm-to-table use case (Janowicz et al., 2022) aims to address concerns regarding the safety of crops affected by smoke and ash from wildfires, providing crucial insights for stakeholders along the supply chain. The graph is implemented in GraphDB, an RDF graph database that supports SPARQL 1.1 and GeoSPARQL. The SPARQL endpoint to the graph is `https://stko-kwg.geog.ucsb.edu/graphdb/sparql`.

KWG includes data about a wide range of *region identifiers*, encompassing global administrative regions, climate divisions, weather forecast zones, census statistical areas, federal judicial districts, and zip code areas. Other feature data include hazard events, road segments, and public health departments. The graph also integrates a set of geographically themed data that significantly influences the use cases. These encompass observation and measurement data, such as climate observations, air quality indices, public health metrics, and natural hazard impacts. Figure 2 is an abstracted pattern of the KWG

Total number of triples in the graph	28,643,164,592
Number of regions (i.e., instances of *kwg-ont*:*Region*) in the ABox	438,927
Number of features (i.e., instances of *geo*:*Feature*) in the ABox	48,397,918
Number of features of interest (*sosa*:*FeatureOfInterest*) in the ABox	6,705,844
Number of geometries (i.e., instances of *geo*:*Geometry*) in the ABox	48,077,280
Number of point geometries (i.e., instances of *sf*:*Point*) in the ABox	8,424,208
Number of lines (i.e., instances of *sf*:*LineString*) in the ABox	741,214
Number of simple polygon geometries (i.e., instances of *sf*:*Polygon*) in the ABox	38,546
Number of complex polygon geometries (i.e., instances of *sf*:*MultiPolygon*) in the ABox	37056438
Number of observations (i.e., instances of *sosa*:*Observation*) in the ABox	254,006,244

TABLE 1 Overview of the KnowWhereGraph (version Santa Barbara or v3.0) statistics (from 09/15/2023).

Ontology (Shimizu et al., 2023) depicting how: 1) entities with explicit geometries are modeled using GeoSPARQL, and 2) environmental observations are modeled using the SSN/SOSA ontology (Janowicz et al., 2019; Haller et al., 2019). At the heart of this pattern is the core class *Region*, which encapsulates dataset-specific subclasses of regions. Qualitative spatial relations or simply topological relations between features are established using the set of DE-9IM relations denoted as $\mathcal{T} = \{sfWithin, sfContains, sfTouches, sfCrosses, sfOverlaps, sfIntersects\}$. Relations in $\mathcal{T}$ are sub-properties of the generic *spatialRelation* within the `kwg-ont` namespace. Each relation follows the naming convention of its topological counterparts in GeoSPARQL, e.g., *kwg-ont*:*sfContains* is semantically analogous to *geo*:*sfContains*. However, the spatial relations in the `kwg-ont` namespace are decoupled from GeoSPARQL's relations (i.e., they are not axiomatically related) to prevent interference with inferencing in the latter's inbuilt functions. Much of the observation data in KWG are not feature-based but have spatial information encoded through identifiers such as ZIP code, FIPS code, and geoid. These identifiers allow thematic data to be linked to their associated region identifiers. Observation data are georeferenced to their corresponding geographical feature of interest using SOSA's *hasFeatureOfInterest* property as shown in Figure 2. This observation-centric linking paradigm is uniformly applied across all data with a spatial dimension (Janowicz, 2012).

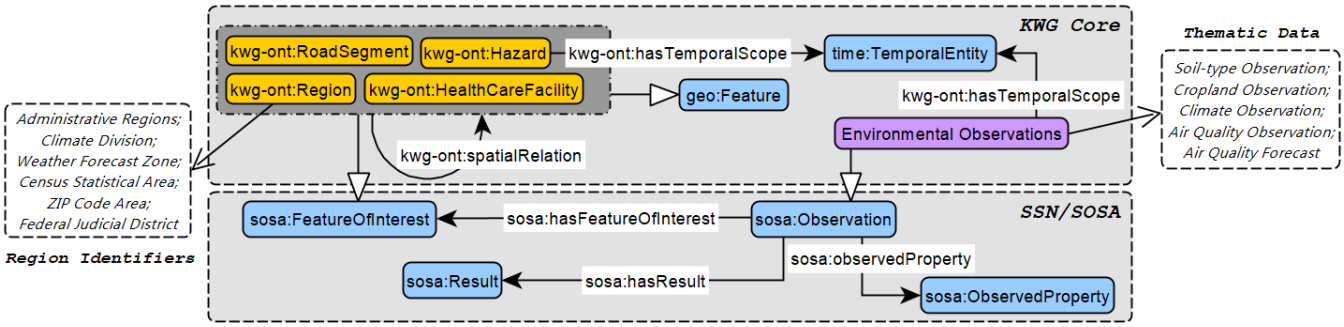


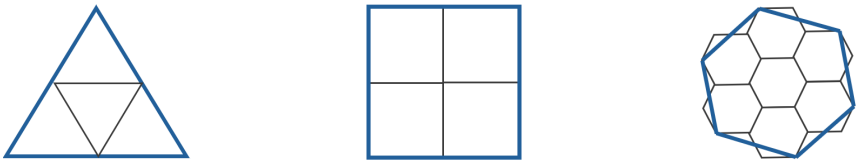
FIGURE 2 Schema diagram that denotes the core concepts/themes in KWG (explicit geographic features denoted using yellow boxes and geographically themed observations denoted using the purple box) and their extension of SSN/SOSA, GeoSPARQL, and OWL-Time ontologies (denotes using blue boxes). The SSN/SOSA pattern denotes the generic graph to query KWG observations. White-headed arrows denote *rdfs*:*subClass* relationships.

Regions are spatially cross-integrated, allowing thematic data to also be cross-linked. For instance, census statistics or climate observations can be analyzed together, but this can be conflicting since different types of regions are of different scales but also discretized for distinct purposes (e.g., weather forecast zones are delineated based on differences in weather, while census statistical areas are delineated based on the number of inhabitants and urbanization). By using the S2 Geometry all the disconnected themes and layers are harmonized onto a common discretized spatial unit of analysis that is unbiased by natural and human processes. This quantization of KWG data onto S2 cells will be discussed later in Section. 4.

Table 1 provides an overview of KWG graph statistics (in terms of the number of total triples, feature classes, feature nodes, and geometry nodes). The number of complex geometries (polylines/multi-polygons) is an indication of the size and complexity of the graph.

2.3 | Discrete Global Grid Systems (DGGS)

A Discrete Global Grid (DGG) is a spatial reference system that tessellates the Earth's surface into a series of discrete, connected, and well-aligned cells. A DGGS is a hierarchical composition of multi-resolution DGGs. At the core of a DGGS are the concepts of cells, levels, and coverings. A *cell* is a fundamental unit in a DGG and represents a specific geographic region with a fixed location and area. Each DGG layer corresponds to a *level* representing the granularity of cells in that layer. Cell sizes and cell counts vary between levels. Each cell at a parent level is linked to a cell at the next level with a finer predefined resolution, allowing for efficient hierarchical operations. Cells are uniquely identified by stable cell identifiers (cell IDs) that provide information regarding the hierarchical level, and structural connections to parents and neighboring cells. The collection of cells that covers a given region or shape is a *covering*. A DGGS is *congruent* if and only if each cell at resolution k consists of a union of cells at resolution $k + 1$. A DGGS is *aligned* if and only if each point within a cell at resolution k is also with its child cell at resolution $k + 1$ (Sahr et al., 2003). The congruence and alignment properties of cells differ between various DGGSs (see a comparison presented in Figure 3).



(a) e.g., OpenEAGGR **(b)** e.g., S2 **(c)** e.g., H3

FIGURE 3 Congruence (i.e., parent-child containment relationships), and aggregation or decomposition resolutions (e.g., squares can be aggregated in groups of four to form coarser resolution objects) in DGGS with varying polyhedron shapes.

The design of a DGGS is primarily governed by the base polyhedron (cube, dodecahedron, icosahedron, octahedron, and tetrahedron) adopted for the spherical approximation of the Earth, and the method adopted to partition the polyhedron into finer cells. The latter choice determines the cell type (square, triangle, or hexagon), aggregation or decomposition resolutions, and congruence properties as compared in Figure 3. Other design choices such as methods of orienting the polyhedron relative to the surface of the Earth, and methods to transform the partitioned planar cells to the sphere are equally important as they determine the level of area, shape, and angular distortion that can impact downstream spatial analysis. DGGSs primarily use hierarchy-based, coordinate-based, or space-filling curve-based indexing systems (Mahdavi Amiri, Alderson, & Samavati, 2019). DGGS frameworks that use a space-filling curve, for instance Hilbert curve (adopted in S2), and Gosper curve (adopted in H3), provide an explicit structural ordering of cells along the path of the curve. A cell's position (or coordinates) along this curve uniquely determines its cell ID, and resolution. Space-filling curves map n-dimensional spatial data to a one-dimensional sequence of cell IDs, approximately preserving the spatial locality of the data points (Uher, Gajdoš, Snášel, Lai, & Radeckỳ, 2019). This transformation provides an efficient method for indexing and querying geographic information. Different DGGSs offer different performance capabilities for different objectives, e.g., rHEALPix for latitudinal data analysis (Bowater & Stefanakis, 2019) and area-based statistics (Gibb, 2016), H3 for modeling of movements through the grid (Kmoch, Vasilyev, Virro, & Uuemaa, 2022), and S2 for smoother multi-resolution analysis (Kmoch et al., 2022).

The OGC's DGGS Standard (OGC, 2017) specifies three core operational requirements for DGGS specifications: quantization, spatial relation, and interoperability operations. *Quantization operations* assign raw or synthesized vector and raster spatial data to DGGS cells. *Spatial relation operations* utilize the hierarchical and connected structure of cells and cell typologies to facilitate cell traversal and spatial analysis functions. Interoperability operations are designed to communicate with end-users or other spatial data infrastructures via standard APIs and data formats. Most DGGSs have open-source software libraries as well as bindings in various programming languages that provide methods for these operations. For example, the H3 library[13] is implemented in C and the S2 library[14] in C++. Existing works have reviewed the operations supported by the state-of-the-art DGGS libraries, their adherence to the OGC standard (M. Li & Stefanakis, 2020), and trade-offs of their different characteristics designs (Bondaruk et al., 2020; Kmoch et al., 2022).

DGGS in Traditional Geospatial Applications: In the last two decades, DGGSs have been instrumental in actualizing the Digital Earth vision (M. F. Goodchild, 2000) through their implementations in sensor data integration (Kraft et al., 2019); multi-dimensional vector big data management (Robertson et al., 2020; Sirdeshmukh, Verbree, Oosterom, Psomadaki, & Kodde,

[13] https://h3geo.org/
[14] https://s2geometry.io/

2019); storing pre-aggregated results (Kraft et al., 2019; Robertson et al., 2020); advanced multi-resolution geo-visualization (Raposo, 2019; M. Li, McGrath, & Stefanakis, 2022); incorporating spatial uncertainty in big data analysis (Robertson et al., 2020); and storing, fusing, and rendering raster data (Strassburg et al., 2010; M. Li et al., 2021; Rawson, Sabeur, & Brito, 2022; Bousquin, 2021; Miao et al., 2023). Specific application domains implementing DGGS include terrain data modeling (M. Li et al., 2021; Lin et al., 2018), ecological studies (Kranstauber et al., 2015; Birch, Oom, & Beecham, 2007), environmental observations modeling (Lin et al., 2018; Romanov & Khvostov, 2018), ride-hailing services (Pereira et al., 2024), and geo-social networking (Woźniak & Szymański, 2021).

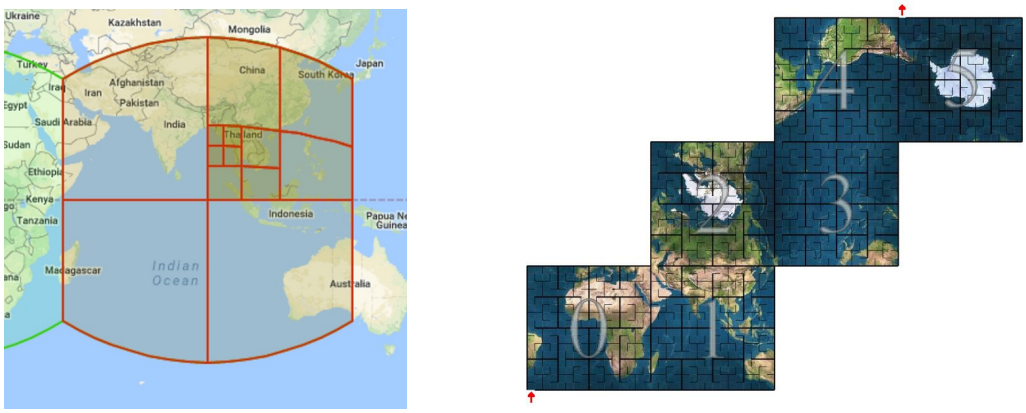
DGGS in GeoKGs: More recently DGGSs have been adopted in GeoKGs for many disciplines and applications. The ISEA3H DGGS is adopted to provide a standardized representation of national multi-source terrain data in (M. Li et al., 2021). Elevation data from multiple raster datasets are quantized on the grid by resampling with bilinear interpolation at the cell centroid locations. (Han, Qu, Huang, Wang, & Pan, 2022) uses the GeoSOT-3D DGGS to integrate spatiotemporal information in the airspace environment for selecting airport sites in response to emergencies. The Spatial-Temporal Knowledge Graph (STKG) in (Böckling et al., 2024) adopts KWG's methodology to transform Open Street Map (OSM) data onto the H3 grid by topologically linking OSM geometries to grid cells via spatial predicates defined by GeoSPARQL. However, the relations between the grid cells use a set of non-topological relations. Due to the non-congruency of H3 cells, inferring spatial relations across multiple resolutions is not straightforward. This necessitates precomputing the links between OSM geometries and grid cells at several levels, resulting in a very large graph. AugGKG is another example that integrates spatio-temporal data on the GeoSOT for spatio-temporal question answering by creating sub-knowledge graphs on each time slice (Han et al., 2023). Wahi, a discrete global grid gazetteer (Adams, 2017) links places in the GeoNames database to each level of three DGGS (ISEA3H, ISEA4H, ISEA4T). Links were created using two types of spatial footprints of places in the gazetteer. The first type is defined by the grid cells that spatially intersect with the source geometry, which can be polygons (when available) or points, and will always cover the source polygon. The second type is defined by the spatial intersection of the centroids of grid cells with geographic features represented by polygons. If no centroid intersects, the single cell with the closest centroid is matched to the feature. This second type of footprint is always smaller or equal in size to the first type and is not guaranteed to cover the entire source polygon. Due to the current unavailability of the server hosting Whi, we were unable to determine the exact predicate used to bin point data into cells. The data is stored in a PostgreSQL PostGIS database, with the option to export data into GeoJSON-LD format.

2.4 | The S2 Geometry

Google's S2 Geometry defines a sophisticated system for tessellating the Earth into a hierarchical mosaic of S2 cells, which are specialized spherical quadrilaterals (Figure. 4a). The edges of these cells are geodesics, meaning they follow the shortest path along the sphere's surface. This hierarchical mosaic is constructed through a nested structure, where each cell acts as a node within a quadtree, and each non-child node can be recursively subdivided to form four child cells. S2 cells are sequentially indexed on a Hilbert space-filling curve that traverses the unit sphere's surface projected onto the six faces of the cube. This projection ensures a relatively even distribution of cells across the spheres surface, maintaining geometric consistency and balance. Each S2 cell is uniquely identified by a 64-bit `S2CellID`, which encodes the cell's location on the curve and its level within the S2 hierarchy. The average area of cells within each level ranges from $\sim 8.5\text{x}10^7 km^2$ (level 0) to $\sim 0.74 cm^2$ (level 30). The structure of the S2 Geometry is formally defined as follows:

Definition 4. *Hierarchical Nested Grid (S2 Geometry)*: Let C_L be a hierarchical nested grid system defined over a spherical surface. C_L is a collection of sets where each level L contains a set of cells. Let $C_0 \subset C_L$ be the set of six base cells at level 0, covering the entire Earth. Then $C_0 = \{c_{0,1}, c_{0,2}, c_{0,3}, c_{0,4}, c_{0,5}, c_{0,6}\}$. Each cell $c_{L,i}$ at level L is recursively subdivided into four child cells at level $L+1$: $c_{L,i} = \{c_{L+1,4i-3}, c_{L+1,4i-2}, c_{L+1,4i-1}, c_{L+1,4i}\}$. Therefore, the set of cells at level L can be represented as $C_L = \bigcup_{i=1}^{6\times 4^L} \{c_{L,i}\}$. Each cell $c_{L,i}$ has a specific level L and a corresponding resolution that defines its spatial extent. The resolution decreases as L increases.

For quantization operations and semantic compression in GeoKGs, S2 offers significant advantages (Zalewski et al., 2021). Unlike traditional GIS, which represents data in flat two-dimensional projections, the S2 Geometry represents data on a three-dimensional sphere, ensuring more reliable topological relationships. Its regular hierarchical structure maintains point-in-cell relationships across resolutions, making qualitative spatial querying highly scalable. The near-equal area of cells in spherical representation is critical for statistical aggregations, for instance, approximating quantities of a cell as a sum over all child cells.



(a) **(b)**

FIGURE 4 **(a)** Hierarchical tessellation of the Earth using the S2 Geometry. **(b)** The S2Cell hierarchy projecting the Earth onto six "base cells". These illustrations are adopted from the S2 Library documentation (Google, n.d.).

The core S2 Geometry library, licensed under the open-source Apache License 2.0, is written in C++ (Veach et al., 2017; Google, n.d.). Software libraries for S2 are available in Python, Java, JavaScript, R, and other languages at varying levels of maturity. The primary purpose of the S2 library is to provide operations for computational geometry and spatial indexing on the sphere. The S2 library supports the following geospatial functionalities:

- Representations of angles, intervals, and geometric shapes (e.g., points, polylines, polygons) both on the unit sphere and in latitude–longitude space.
- Constructive operations (e.g., union, difference, buffer), boolean predicates for testing topology relationships (e.g., containment, intersection), and mathematical predicates for testing spatial relationships (e.g., distance within a limit, point in region) for arbitrary geometric objects.
- Algorithms for measuring areas, calculating distances, computing centroids, simplifying geometry, and finding nearby objects.
- Functions for cell navigation (retrieving parent, children, and neighbor cells), and traversing cells at the same level along the Hilbert curve.
- Support for spatial indexing (conversion methods from geographic coordinates to cells and back), including the ability to approximate regions as unions of cells (Figure. 4b).

3 | METHODOLOGY

This section details the technical intricacies underpinning the access and utilization of S2 within KWG. It specifically presents an overview of how S2 is modeled and ingested into KWG. Following this, the section discusses the general technical strategy adopted to implement the various quantization methods, which will be elaborated upon in Section 4.

3.1 | Conceptual Schema for the S2 Geometry in KnowWhereGraph

GeoSPARQL 1.1 (?, ?) introduces the *geo:asDGGS* property to link a DGGS cell to its cell ID serialization in the *geo:dggsLiteral* datatype. However, GeoSPARQL does not natively interpret DGGS literals, necessitating external implementations for handling them. Currently, GeoSPARQL-DGGS (David Habgood, n.d.) is the only library available that leverages RDFlib to support GeoSPARQL's topological functions on DGGS literals, but it is limited to the rHEALPix DGGS. Given this limitation, we chose to represent S2 cells as vector geometries (with cell vertices forming a polygon) using the WKT specification. This approach ensures compatibility with existing geospatial standards and tools, providing a robust solution for integrating S2 cells within GeoSPARQL-extending frameworks such as KWG.

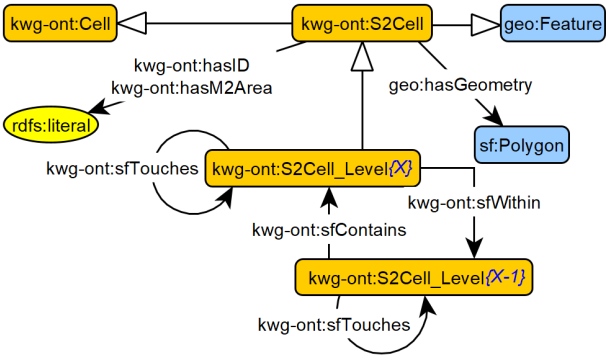


FIGURE 5 Schema diagram that denotes the modeling of the S2 Geometry as an extension of GeoSPARQL's classes.

The schema diagram in Figure. 5 represents the conceptual modeling of the S2 Geometry in KWG. The primary class is *S2Cell*, denoted as a subclass of *geo*:*Feature* and *Cell*, which generalizes cells in a DGG (Shimizu et al., 2021). The geometry of each S2 cell is a simple feature polygon. The more detailed geometric representation of S2 cells is adopted in KWG primarily for visualization needs, but if visualization and runtime topological querying is not a priority this geometry can be abstracted to a point. *S2Cell* is specialized into subclasses denoted as *S2Cell_Level*{*X*}, where *X* signifies various levels in the S2 hierarchy. Cells are structurally connected within a level and between levels via DE-9IM relation from the KWG ontology (Figure 2). Hierarchical *parent–child* relationships between S2 cells are indicated by *sfContains* and *sfWithin*, while *neighbor* relationships are represented by *sfTouches*. Hierarchical connectivity is explicitly precomputed up to one level, while connectivity between cells in all levels is inferred through the transitive property defined on *sfContains* and *sfWithin*. In a GraphDB repository with reasoning enabled, these inferred or implicit statements are pre-materialized during the data loading stage via forward-chaining. Further, each S2 cell is annotated with details about its identifier and area using data properties *hasID* and *hasM2Area*, respectively, while cell resolution is implicit in the class name.

The `s2sphere`[15] python implementation of the S2 Geometry library is used in generating the S2-RDF graph, a subgraph in KWG. Figure 6 illustrates one S2 cell from the graph displayed on a map via KWG's dereferencing interface. The snapshot also displays the attributes and linkages with other S2 cells in the graph.

3.2 | Challenges

Using S2 cells as polygon geometry representations in the WKT serialization introduces certain ambiguities. The differences between edges in planar space and spherical space lead to indexing issues and visualization artifacts, which are discussed here.

Indexing: The default coordinate reference system for WKT literals is WGS84. Geographic coordinates in WGS84 span from 180° to 180° longitude and -90° to 90° degrees latitude on a two-dimensional plane. Conversely, S2 maps points on the earth onto a mathematical sphere and then projects the sphere's surface onto the six sides of a cube (Figure 4b). Consequently, some S2 cells span the antimeridian (±180°) and the poles. Since indexing in GraphDB uses WGS84, we encounter indexing issues with cells that cross the antimeridian or the poles because they are not normalized (either through wrapping or splitting) and thus become topologically invalid. Figure 7a illustrates an example of an S2 cell that crosses the antimeridian, but, cannot be indexed in the graph. This issue occurs for all geometries when the difference between the longitudes of 2 consecutive points in the path is >=180°. In a spatial database, such as GraphDB, this ambiguity needs to be resolved to ensure accurate representation and querying. Alternative options to circumvent this issue would be to split S2 cells at the antimeridian or represent them as points.

Visualization Artifacts: When S2 cells are visualized on maps with a Mercator projection, as in most web maps, their true shape and size on the sphere may not be accurately reflected, with more pronounced distortion at higher latitudes. S2 uses geodesic edges, meaning that the edges of the cells follow the shortest path on the spherical Earth approximation. On a Mercator projection, geodesic paths in the east–west direction, except at the Equator, appear as curves bulging towards the poles (Fig. 8). Consequently, the "horizontal" edges of S2 cells appear curved towards the poles. When S2 cells are represented by straight

[15] `http://s2sphere.readthedocs.io/`

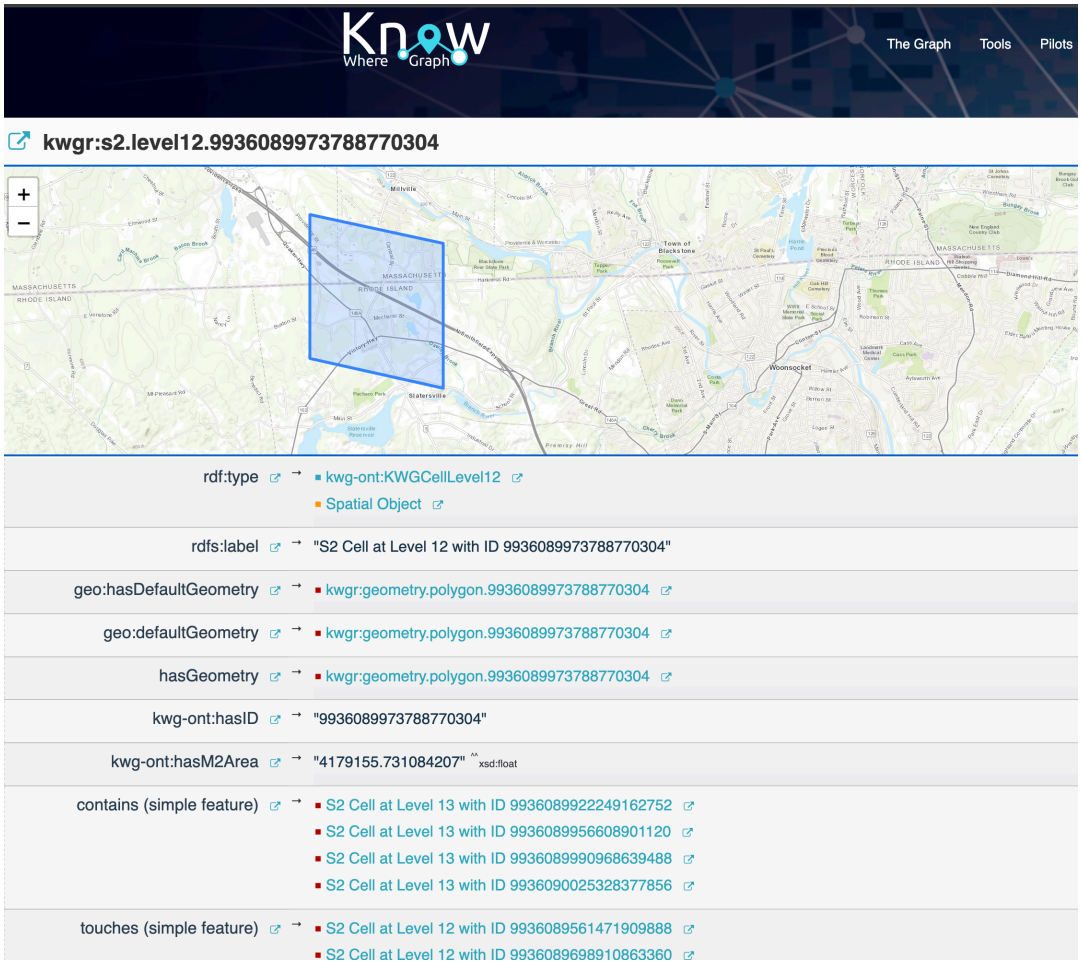
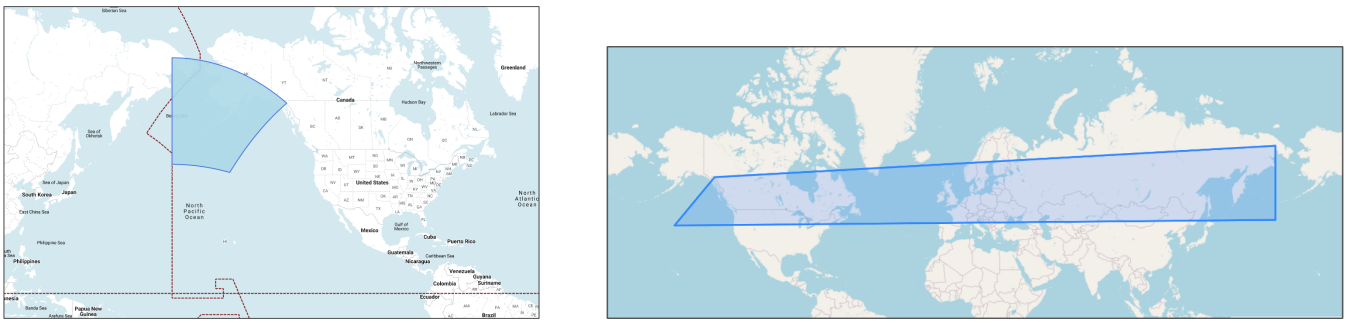


FIGURE 6 S2 cell with cell ID 9936049085700112384 (level 8) as modeled in KWG and visualized using KWG's phuzzy dereferencing interface.



(a) With wrapping (using BigQuery Geo Viz) **(b)** Without wrapping (using wktmap.com)

FIGURE 7 S2 cell at level 2 with cell id 9007199254740992000. The WKT serialization of the cell is "`POLYGON((180 67.3801, 180 45, -157.3801 42.7093, -135 59.4910, 180 67.3801))''^^geo:wktLiteral`.

lines connecting their four corner vertices in a Mercator projection, their east–west edges are incorrectly rendered as rhumb lines, or paths that follow a constant bearing relative to north, like lines of latitude. Both the shape and edge distortions can lead to potential misinterpretations of the spatial data. For example, points that are topologically within an S2 cell may appear to lie outside the cell, and vice versa (Fig. 8). Additionally, S2 quadrilaterals do not tessellate perfectly on planar maps, leading

to gaps or overlaps between cells. Figure 7 denotes another visualization artifact, where an S2 cell that crosses the antimeridian appears wrapped around the map when represented as a WKT map.

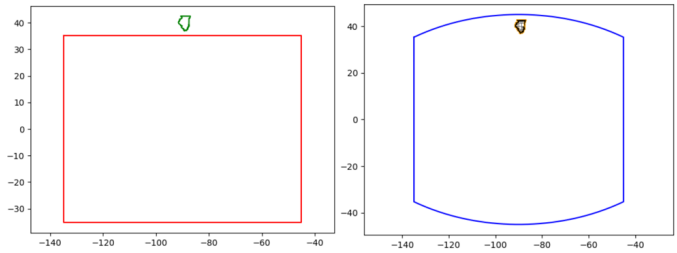


FIGURE 8 Visualization artifacts when geodesic edges of S2 cells are considered as rhumb lines, or paths of constant bearing relative to north. (a) Illinois is not inside S2 cell, level 0, hex 9, with straight-edge boundaries. (b) Illinois is within S2 cell, level 0, hex 9, with geodesic boundaries. hex refers to a hexadecimal representation of an S2 cell identifier.

3.3 | Technical Strategy for Data Quantization in KnowWhereGraph

S2 in KWG facilitates efficient data storage and retrieval through different data quantization strategies. As previously mentioned, indexing and query operations provided by the S2 Geometry are not natively available within GraphDB. Quantization is adapted in the KWG environment by way of view materialization (Ibragimov, Hose, Pedersen, & Zimányi, 2016), where triples — spatial, temporal, thematic, or their combinations — are pre-computed using the `s2sphere` library and persisted within the graph. These materialized views enhance the scalability of query processing, particularly for complex spatial analytic operations, by circumventing direct access to geometric data (e.g., WKT literals) within the graph. We employ tailored patterns (Gangemi & Presutti, 2009) to guide the selection of views for materialization (Lan, Wu, & Theodoratos, 2022). These patterns delineate different ways to quantize data to S2 cells. Nodes in a pattern depict frequently accessed relationships (based on use-case needs) or spatial joins earmarked for materialization, thereby mitigating query latency.

Precomputation of triples also facilitates data conflation by consolidating vector, raster, and other geographically themed datasets at the intersection of the S2 grid. This process enriches the graph, improving the efficacy of graph-based algorithms such as similarity detection and clustering, and fostering serendipitous discoveries across interconnected data entities. Unlike the space costs associated with relational databases, materializing views in a graph database also allows for data compression, which would be impossible if data were ingested with their original structure and precision. Whether for enrichment or compression, the choice of patterns depends on a nuanced consideration of factors such as on-the-fly computation overheads, graph size, and graph maintenance.

4 | IMPLEMENTING DATA QUANTIZATION ON S2 IN KNOWWHEREGRAPH

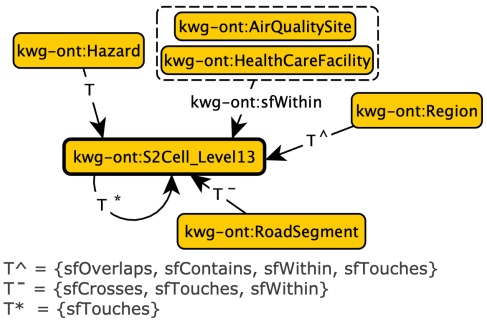
In this section, we elaborate on the two quantization strategies adopted in KWG, the ontology patterns adopted for materializing views, and their specific benefits.

4.1 | Topological Enrichment of Vector Data

Research on computational time and space trade-offs in GeoKGs has demonstrated that precomputation of topological relations can yield significant performance benefits (Regalia, Janowicz, & Gao, 2016; Regalia et al., 2019). Traditionally, places (represented as *Region* nodes in KWG) have been central for topologically linking other spatial features (e.g., *Hazard* and *RoadSegment*) within GeoKGs (Regalia et al., 2019). However, place boundaries are inherently dynamic, meaning their topological relationships with intersecting features can change over time. For instance, before 2011, South Sudan was spatially *within* Sudan, but after gaining independence, the relationship changed to a border-sharing or *touches* relationship. In contrast, S2 cell boundaries are fixed and immutable, providing a stable spatial framework for linking features and supporting multi-resolution and hierarchical spatial reasoning (Timpf & Frank, 1997).

The S2 grid at level 13 is selected as the reference layer, offering an appropriate level of detail (~1.27 km^2 per cell) for topologically interlinking KWG data. From this point forward, cells in this layer will be referred to as S2 reference cells. *Topological enrichment of vector data* is achieved by precomputing and materializing topological relations between entities with explicit geometries and S2 reference cells in the graph, creating a dense network of interconnected spatial objects. The spatial relations utilized are a subset of topological relations from $\mathcal{T}$ (see Section. 2.2), which are sub-properties of *kwg-ont:spatialRelation*. Figure. 9a illustrates the ontology pattern for this view materialization. Certain datasets ingested into KWG (e.g., public health departments, and pharmacies) provide addresses or other location-based information, such as ZIP codes, which are geocoded into point coordinates for topological linking to S2 cells.

Figure. 9b represents the OWL properties defined over the topological relations in $\mathcal{T}$. We precompute most of the implicit triples induced by properties to reduce graph initialization costs (*total materialization* at load time). We leave the triples resulting from the transitive property (on *kwg-ont:sfWithin*) to be entailed from forward chaining during graph initialization.



(a)

kwg-ont:spatialRelation ($\mathcal{T}$)	**geometry combination**	**property**	**precomputed vs. inferred**
kwg-ont:sfTouches	all except P/P	symmetric	precomputed
kwg-ont:sfOverlaps	P/P, L/L, A/A		
kwg-ont:sfContains ; *kwg-ont:sfWithin*	all	inverse	precomputed
		transitive	inferred
kwg-ont:sfEquals ; *kwg-ont:sfDisjoint*	Not precomputed or inferred, but relations are included in the ontology for completeness		

(b)

FIGURE 9 **(a)** Pattern for topologically enriching vector geographic entities. **(b)** Table showing the properties asserted on topological relations in KWG for various geometry type combinations (P-Point, L-Curve, A-Surface) and the corresponding A-Box triples that are materialized.

This enrichment, although it may significantly increase the number of triples in the graph, makes feature geometries obsolete, thereby enhancing querying and data processing speed. Table. **??** shows the spatial query speedups ????? TO ADD once the table details are filled in.

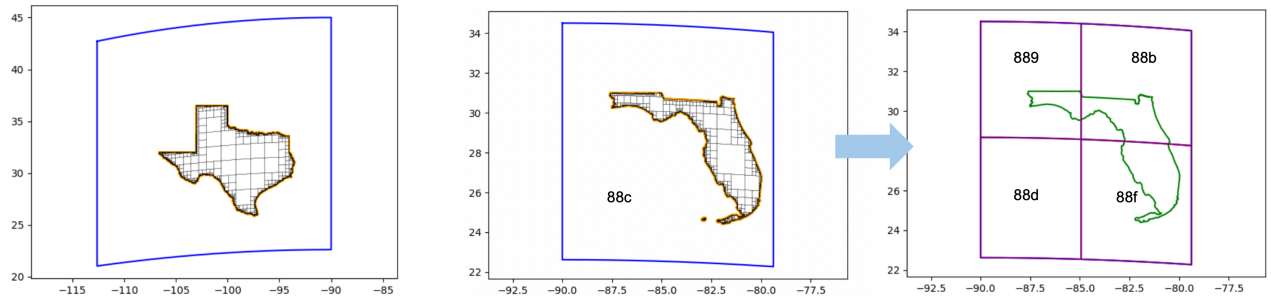
Query type	**Query details**	**No. results retrieved (via S2)**	**Query time (via S2)**	**No. results retrieved (via GeoSPARQL)**	**Query time (via GeoSPARQL)**
P/A	hospitals in Santa Barbara [Q1]	22	0.3 secs	23	30 secs
L/A	roads intersecting Santa Barbara [Q2]	1747	5.7 secs	–	Timeout
A/A	statistical areas overlapping Santa Barbara [Q3]	4	1.9 secs	4	5.8 secs

TABLE 2

A. Optimized Topological Enrichment:

Classic topological enrichment utilizes S2 cells at level 13. However, enriching datasets with large polygons, such as drought and smoke plume areas, at this same resolution leads to significant data volumes, making computation and storage inefficient. For instance, a single temporally sliced multi-polygon representing drought can span the entire contiguous United States, covering an area of up to 6.8 million square kilometers. To address this inefficiency, a discrete and compressed enrichment framework (Zalewski et al., 2021) is employed, which combines DE-9IM with the hierarchical nesting of S2 cells. This approach links large polygons to maximally filling S2 cells at coarser resolutions. The transitivity of within/containment relations in the spatial ontology allows for the inference of all triples describing the spatial relationship between the region and S2 reference cells. This approach significantly reduces the data volume required for representation and speeds up computations by limiting the number of cells involved in each query.

B. Computational Method for Boolean Topological Relationships: The S2 Geometry Library provides classes for representing shapes (points, curves, and regions) on the Earth's surface. These shapes are defined using points and geodesic curves, which presents challenges when working with vector data serialized in the WGS84 format, as these data do not inherently



(a) **(b)**

FIGURE 10
Examples for optimized topological enrichment. (a) Texas is optimally filled with S2 cells at various levels. (b) Florida optimally filled with S2 cells at various levels, while also contained within one S2 cell, level 3, hex 88c, and overlapping four S2 cells, level 4, hex 889, 88b, 88d, 88f.

conform to geodesic boundaries. To address this incompatibility, we developed a custom algorithm to compute topological relationships (based on the DE-9IM) between the S2 reference cells and geometric objects in KWG. This algorithm leverages the Python wrapper for the S2 library alongside Shapely, which handles planar geometric objects. Key methods from the S2 librarys `S2RegionCoverer` class, such as `GetCovering` and `GetInteriorCovering`, were instrumental in generating `S2CellUnion` objects that either cover or are contained within the specified region. The technical steps of the algorithm are as follows:

1. **Geometry approximation:** Each WGS-84 geometry ([multi-]point, [multi-]line, [multi-]polygon) is approximated as its analogous S2 object(s) on the unit sphere. In the case of polygons, the boundary is segmented by adding vertices such that the distance between any two consecutive vertices does not exceed a specified tolerance. This ensures that linear edges are replaced by geodesic curves that closely approximate the original planar shapes.
2. **Creating coverings:** The `S2RegionCoverer` is then used to approximate the S2 object as unions of S2 reference cells, known as cell coverings. This includes ordinary coverings (cells at various levels that cover the geometry), homogeneous coverings (cells that cover the geometry at a specified level), and interior coverings (cells at various levels strictly within the geometry).
3. **Topological relationship computation:** For point data, each point is directly mapped to an `S2Point` and associated with an `S2CellID` at the maximum resolution level. For line and polygon data, topological relationships are computed by identifying overlapping or contained cells using the coverings generated. For example, to identify the cells at a specified level that are strictly within the given shape, an interior covering is used. Overlaps are determined by the difference between homogeneous and interior coverings.

4.2 | Grid-Based Data Discretization

Data discretization involves transforming non-DGGS spatial data (vector or raster) through different transformation strategies (statistical aggregation, decomposition, spatial overlay, etc.) to represent them as S2 cell–based gridded data for flexible representation and analysis while maintaining spatial relationships and scalability across different resolutions. Cells are indicated as the *FeatureOfInterest* (Janowicz et al., 2019) to which various data values are mapped, via object properties or data properties. These data values can be 1:1 grid mapping (for raster cells), aggregated summaries, or multi-level decomposition — see examples in Figure 11. Quantized values can be quantitative (e.g., area covered by a smoke plume) or qualitative details (e.g., categorical data related to soil porosity).

Vector data in KWG span a wide range of scales, from large-scale features like smoke plumes covering extensive areas to highly localized data such as wildfire events represented as points or small polygons. Although wildfires are a primary source of smoke plumes, their spatial and temporal distributions provide different insights. Wildfire distribution can reveal persistent hotspots and frequent fire activity, while smoke plume distribution is essential for assessing correlations with health impacts. Analyzing both datasets together enhances understanding of regional air quality and supports the development of

smoke forecast models. However, comparing these disparate features without appropriate scaling and context can lead to spatial mis-contextualization, where localized patterns or anomalies may be obscured, potentially resulting in incorrect interpretations of their interrelationships. Identifying the optimal scale for analysis — whether for correlation or hotspot detection — is not always straightforward, underscoring the need for data solutions that support multi-scale analysis (Manley, 2021). Discretizing large-scale vector features by decomposing them into varying resolutions of S2 cells can help reveal critical relationships and patterns that may be obscured at other scales.

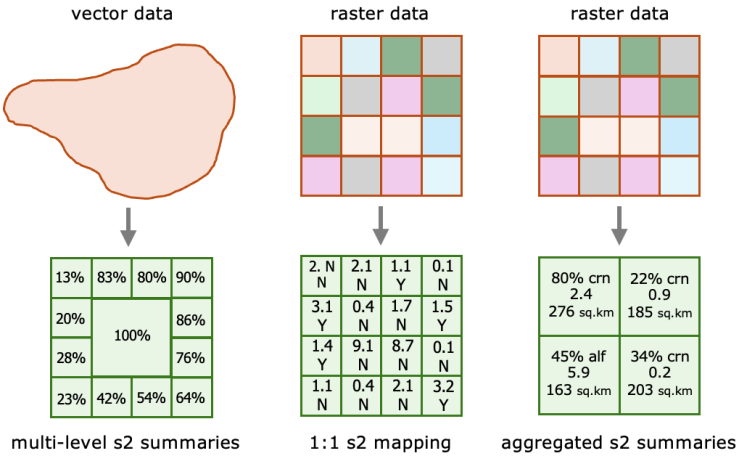


FIGURE 11 Examples of data transformation methods to discretize raster and vector data on the S2 grid.

Non-DGGS data can also be aggregated on the S2 grid as summaries (e.g., kernel density, spatial statistics, histograms, frequency transformations) to offer insights into aggregated spatial patterns without necessitating the retention of detailed individual records. Aggregation can occur at different levels of spatial granularity. For instance, real-time fire data can be aggregated at fine spatial scales (e.g., neighborhood or street levels) to facilitate targeted response efforts and resource allocation within localized areas. Aggregating the same fire data at higher spatial levels (e.g., city or county levels) helps in understanding broader phenomena such as air pollution or regional fire trends. Summaries computed at a lower level can be aggregated upward through the S2 hierarchy to compute values for larger spatial regions. Summaries can also be decomposed downward within the hierarchy, provided the feature in consideration is either *sfEqual* or *sfWithin* the S2 cell at the top level. This hierarchical approach supports adaptive analysis, where users can dynamically adjust the level of decomposition to focus on specific spatial features or patterns of interest.

Both data decomposition and aggregation through discretization facilitate multi-scale analysis for both spatial and non-spatial dimensions of the data. The specific choice of method and selected grid resolution should reflect both the purpose and scale of the analysis. The hierarchical grid structure of S2 supports seamless transitions between different scales — from local to global — allowing for detailed representation and the discovery of patterns that may not be evident at a single scale. Furthermore, this quantization method improves data management by supporting feature simplification. It also supports advanced geovisualization techniques, allowing for dynamic scaling and context-aware rendering. This ensures that large-scale patterns and localized details are preserved and accurately represented, enhancing the ability to uncover meaningful spatial relationships and insights.

Here we describe two examples of data discretization in KWG: one for a vector dataset and another for a raster dataset.

4.2.1 | Vector Data to S2 Cell Discretization

The Soil Survey Geographic Database[16] (SSURGO) contains information on hundreds of soil attributes, such as chemical and physical properties, and their derived interpretations. The dataset consists of over 36 million soil map units distributed across more than 70 shapefiles, with varying scales depending on the region. For instance, some states like Nevada and Utah have large polygons at a scale of 1:63,360, while states with higher agricultural interest, such as Indiana and Iowa, have smaller polygons at a scale of 1:12,000. Due to its size and complexity, this dataset is often impractical for use in an integrated analytics environment. In KWG, this dataset is decomposed onto the S2 level 13 grid. Figure 12 illustrates the pattern for this modeling. Overlap areas between individual soil polygons and an S2 cell (in square kilometers) are precomputed and

[16] https://websoilsurvey.nrcs.usda.gov/app/WebSoilSurvey.aspx

materialized as observations in the *SoilOverlapAreaObservation* class, with the S2 cell as the feature of interest. The geometry of the original soil map units, denoted as *SoilMapUnit*, is also included in the graph along with the soil map units' detailed soil attributes, represented as *SoilMapUnitObservableProperty*. These geometries facilitate more accurate visualizations, as a cell can contain disconnected geometries from one soil multipolygon. Through the relation composition *overlapObservedWith* ∘ *isFeatureOfInterestOf*, users can quickly explore all soil properties associated with an S2 cell without needing to access the soil polygon geometries. Ultimately, discretizing soil data over a high-resolution S2 grid allows for improving its level of detail and accuracy by integrating it with other thematic data, such as landforms, floodplains, crop types, and high-resolution digital elevation models (DEMs).

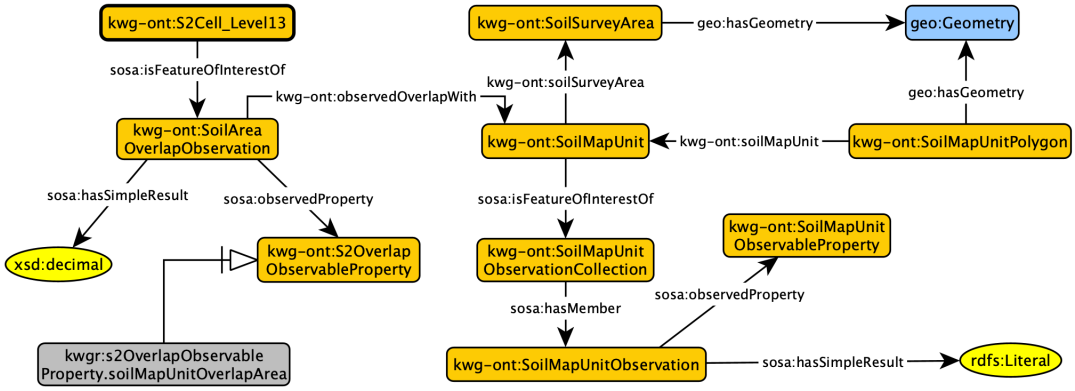


FIGURE 12 Ontology pattern for discretizing vector soil data to S2 reference cells.

To discretize SSURGO data, we build upon the method for computing topological relationships between vector geometries and S2 cells, as described in Section 4.1. Vector *SoilMapUnitPolygon* geometries are approximated as `S2Polygon` objects, and their cell coverings are generated using S2 reference cells. The overlaps between the cells in the covering and the `S2Polygon` are computed using the `S2Polygon.GetOverlapFractions` method, which returns the fraction of overlap for each cell relative to the polygon. This fraction is then multiplied by the cell's area to compute the actual overlapping area on the Earths surface. This quantifiable measure provides insights into the soil properties of reference S2 cells

4.2.2 | Raster Data to S2 Cell Discretization

The USDA National Agricultural Statistics Service Cropland Data Layer[17] (CDL) is a raster crop-related land cover layer produced annually using satellite imagery and extensive agricultural ground reference data. The dataset has a ground resolution ranging from 30 to 56 meters, depending on the state and year. It quantifies changes in forest extent and height, cropland, built-up lands, surface water, perennial snow and ice extent, and over 134 crop types, making it valuable for monitoring agricultural land use for KWG needs. To handle this in KWG, pixel-based information is transformed into data values corresponding to the area of each crop type within an S2 level 13 cell. Figure 10a illustrates the pattern for this materialization in the graph. The areal extent (in percentage) of each crop type, denoted as *CroplandObservableProperty*, in an S2 level 13 cell is computed and materialized as a temporally-scoped observation (*CroplandS2OverlapObservation*). To ensure accuracy, we verify that the sum of all land cover observations within each cell, indexed (or featured as an interest) over a specific timestamp, is 100%. This discretization process compresses the heavy volume of raster CDL data and represents it in the graph, simplifying and scaling computation. Consequently, queries are no longer spatial but attribute-based, enhancing efficiency and manageability.

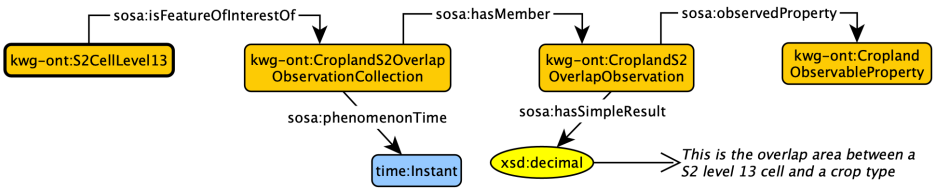


FIGURE 13 Ontology pattern for discretizing raster cropland data to S2 reference cells.

[17] `https://croplandcros.scinet.usda.gov/`

The discretization of cropland data with S2 cells involves transforming the raster representations into S2-compatible geometries for analyzing spatial relationships.

1. **Defining spatial coverage:** Raster data often comes with its own coordinate reference system, which needs to be aligned with the S2 library's spherical geometry before generating S2 cell covering and overlap calculations. The first step involved converting the rasters spatial extent into an S2-compatible format. To do, the spatial bounds of the raster dataset is extracted from its metadata. This determines the geographical coordinates of the rasters corners. These bounds are transformed into an `S2Polygon`, which represents the geographical area covered by the raster. This conversion ensures that the rasters spatial footprint is accurately mapped onto the spherical surface.
2. **Generating covering:** The next step is to cover the `S2Polygon` with S2 reference cells using the `S2RegionCoverer` method.
3. **Discretizing data:** For each S2 cell, aggregate the raster values to compute statistical summaries such as the mean or sum. This provides a quantifiable measure of raster data distribution within each S2 cell.

5 | BENEFITS OF S2 IN ADVANCING ANALYTICAL CAPABILITIES OF KWG

This section explores four use cases to illustrate how the two quantization strategies improve the analytical pipeline in KWG to help optimize storage and computation, enhance spatial query capabilities, and facilitate seamless integration and analysis of heterogeneous datasets.

5.1 | Conflating S2 Cells with Thematic Observation Data via Object Property Paths

Observation data, such as hurricane severity and impacts, population demographics, air quality indices, and public health observations, are collected through explicit or implicit sampling strategies. These datasets contain regional identifiers (e.g., ZIP code, FIPS code, NWZ code, statistical area code) but are not explicitly connected to region geometries. In KWG, these georeferenced observation data are linked to their corresponding geographic entities or regions, such as ZIP code tabulation areas, administrative region boundaries, national weather zones, and statistical areas, via the *sosa*:*hasFeatureOfInterest* relation as shown in Figures 14 and 15a.

From a query performance perspective, using S2 cells to topologically enrich geographic entities with explicit geometries is highly beneficial. This approach consequentially conflates S2 cells with various observation data via object property paths, thereby thematically contextualizing the cells. Figure 14 shows an example of the property path from various observations to S2 cells via connected geographic regions. The result of this process is a set of mappings from observations to geographic entities to sets of S2 cells at multiple levels of the S2 grid. This thematic enrichment of S2 cells entails a level of abstraction determined by the chosen geographic region and the resolution of the cells. However, it plays a crucial role in uncovering latent patterns, for example identifying that regions with high social vulnerability scores have an increased exposure to air pollution.

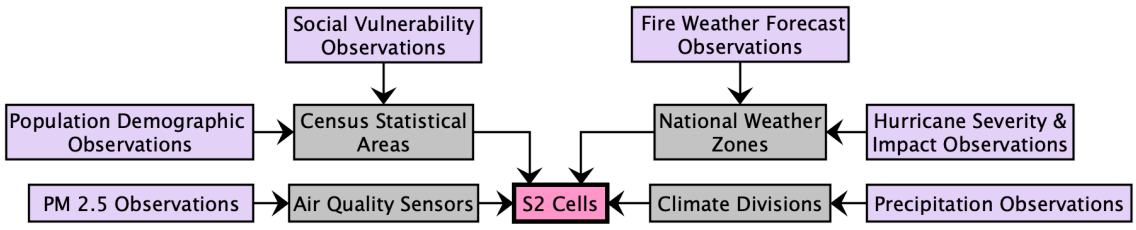
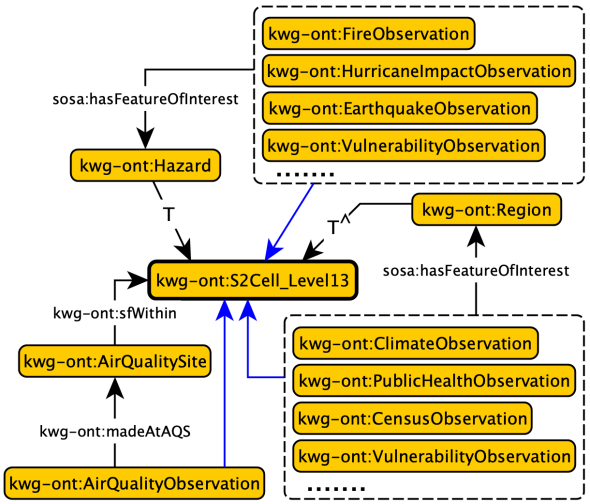


FIGURE 14 Contextual enrichment of S2 cells with various observation data via geographic entities in KWG. Purple boxes indicate observation data and grey boxes indicate entities with explicit geometries.

Figure 2 illustrates the ontology pattern that explicitly links observations to geographic regions via the *sosa*:*hasFeatureOfInterest* relation. The path between observations and S2 cells is defined through OWL 2 property chains (Potoniec, 2022), specifically through the composition of *sosa-hasFeatureOfInterest* and *kwg – ont*:*sfOverlaps*. In KWG, the values for these object paths are computed on-demand via SPARQL queries.

Thematic contextualization of S2 cells is highly beneficial for use cases that require granular spatial summarization, cross-theme data exploration, and spatial interpolation. Conflation links different observation themes to S2 cells, allowing users to geographically navigate through thematic spaces. This is particularly useful for searching related locations within the knowledge graph. For instance, an analyst from a humanitarian aid organization might search for places based on specific thematic criteria, such as identifying S2 cells associated with fire forecasts, weather conditions favorable to fires, and socioeconomically vulnerable neighborhoods. This integration of data helps the analyst formulate effective emergency plans. Figure 3a illustrates the SPARQL query used to retrieve this information from KWG. Additionally, this thematic conflation supports advanced geo-visualization techniques, enabling dynamic representation of hotspots or areas of interest at varying zoom levels, which is critical for detailed spatial analysis and decision-making.



(a)

```
PREFIX sosa: <http://www.w3.org/ns/sosa/>
PREFIX kwg-ont: <http://stko-kwg.geog.ucsb.edu/lod/ontology/>
PREFIX time: <http://www.w3.org/2006/time#>
SELECT ?h where {
    ?f a kwg-ont:FireForecastObservation;
        sosa:hasFeatureOfInterest/kwg-ont:sfOverlaps ?s;
        kwg-ont:hasTemporalScope/time:inXSDDate ?day.
    ?t a kwg-ont:MaxTemperatureObservation;
        sosa:hasFeatureOfInterest/kwg-ont:sfOverlaps ?s;
        sosa:hasSimpleResult ?temp;
        kwg-ont:hasTemporalScope/time:inXSDDate ?day.
    ?t a kwg-ont:SocialVulnerabilityObservation;
        sosa:hasFeatureOfInterest/kwg-ont:sfOverlaps ?s;
        sosa:hasSimpleResult ?svi.
    ?s a kwg-ont:S2Cell_Level13.
    FILTER (?day > "2010-01-10"^^xsd:date)
    FILTER (?day > "2010-01-10"^^xsd:date)
    FILTER (?day > "2010-01-10"^^xsd:date)}
```

(b)

FIGURE 15 (a) Pattern for conflating observation data on S2 cells via object property paths. The blue arrow denotes the shortcut that can be computed on demand with dedicated SPARQL queries or materialized via ontology rules. (b) Sample SPARQL query to identify S2 level 13 cells linked to fire risk on 10th January 2010.

5.2 | Multi-scale Representation of Quantitative Geospatial Data

Quantitative datasets encompass various observation datasets typically represented as numerical values associated with specific locations. Examples include the Social Vulnerability Index (SVI) linked to counties or the PM 2.5 concentration measured at point-based air quality monitoring sites. While these datasets can be queried through conflation at specific S2 grid levels, some statistical queries may not benefit from a multi-scale spatial representation. This principally depends on the kind of quantity (amounts vs. measurements as discussed in (Top, 2024)). For instance, while a county-level SVI measurement can be mapped to all S2 level 13 cells within the county, these values cannot be simply summed or averaged to obtain a value for a higher-level spatial region, such as a state. Conversely, spatial overlap areas of a crop type from multiple S2 cells can be summed to determine the overlap area of a larger S2 cell that contains them. Quantities that can be aggregated (mereotopological quantities) versus those that cannot (arithmetic quantities) are further defined and axiomatized in (Top, 2024). Discussing the implications of discretizing both types of quantities and their aggregations over multiple resolutions is beyond the scope of this paper. However, understanding these distinctions is crucial for accurate geospatial analysis and data integration within S2 Geometry frameworks.

Certain quantity measurements, when aggregated over a larger region, should incorporate weighting measures to adjust for spatial biases or the uneven distribution of data points within the region. For example, an averaging query, which requests the average of data values within a specified spatial polygon, can benefit from knowing the s2 cell containing the data point, its spatial relation concerning the spatial polygon, and the size of the s2 cell. These details can help determine the placement of data points within the polygon and the appropriate weights to allocate. By incorporating these weighting factors, the aggregated results may better approximate the true characteristics of the entire area, rather than being skewed by the concentration or

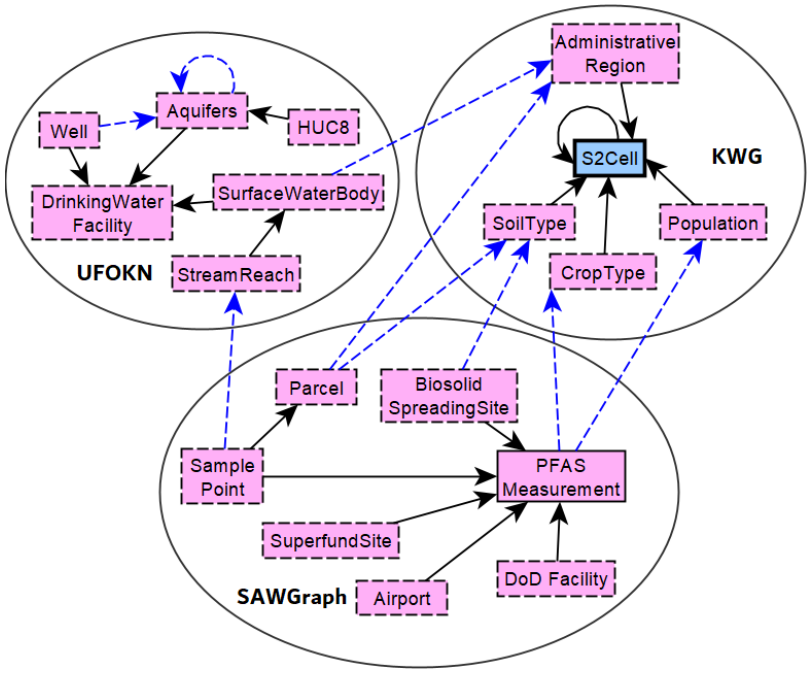


FIGURE 16 Synoptic illustration of cross-integration between KnowWhereGraph, SAWGraph, and the Urban Flooding Open Knowledge Network (UFOKN). Concepts in pink boxes are linked to S2 cells. Blue dashed arrows indicate object properties that can be queried via S2 integration.

distribution of the data points. An important goal in defining such a representation is to enable queries at the data's coarse scale without significant loss in accuracy.

Consider the air quality observation dataset in KWG, which contains concentration measurements for pollutants such as Ozone and PM 2.5. The PM 2.5 concentration is a crucial metric in air quality assessment systems. Typically, air quality monitoring sites are associated with counties, but their distribution within these counties is neither uniform nor evenly spaced. This uneven distribution complicates the derivation of accurate county-level estimates since PM 2.5 concentration exhibits nonlinear spatial characteristics. For instance, urban areas require data with a spatial resolution of 200–400 meters to improve pollution exposure modeling accuracy (Stroh, Harrie, & Gustafsson, 2007). In KWG, individual PM 2.5 measurements are mapped onto 1 x 1 km^2 grid cells at S2 level 13. These measurements are then hierarchically contained within cells of lower resolution that are also topologically connected to county polygon boundaries through quantization. This hierarchical structuring and enrichment enables one to determine the spatial distribution of monitoring sites within counties, facilitating the calculation of appropriate weights for more accurate regional estimates. Furthermore, the fine-grained overlap of monitoring sites with other data layers achieved through S2 enrichment and discretization enhances our capacity to identify other controlling quantities or measures that may disproportionately influence PM 2.5 concentrations at specific sites. Thus, quantization and leveraging the hierarchical and topological relationships within the S2 grid framework can significantly improve the precision and reliability of aggregating quantities over a spatial dimension, particularly in regions with complex spatial characteristics.

5.3 | Cross-Graph Integration

S2 serves as the *spatial fabric* for cross-discipline graph integration and interoperability in the OKN framework. The SAWGraph integrates data on water, soil, plant and animal tissue, feed, and agricultural food products tested for per- and polyfluoroalkyl substances (PFAS). This integration includes test results, testing sites and facilities, and potential contamination sources, meeting the analytical needs for PFAS monitoring. SAWGraph connects with KWG and UFOKN to identify and trace pathways of PFAS transport and accumulation within the US's food and water systems. It also helps prioritize additional sampling in specific locations, such as wells, agricultural lands, or crops. This is achieved through federated querying using S2 cells as points of interest. To enable this functionality, individual graphs must implement S2 integration with specific datasets.

Figure **??** demonstrates a sample query that illustrates how spatial data is utilized in this context. Figure 16 depicts key spatial features within each graph that are integrated to S2 cells via topological enrichment. This integration allows for the contextualization and conflation of data by leveraging complementary information from different graphs. For instance, inferred links (denoted using dashed blue arrows) enable queries such as: "*Which towns in the state of Maine have higher than average contamination of PFAS compared to towns in the state of Illinois?*"

5.4 | Graph Sharding

KWG is a continuously expanding graph, with the number of geometry nodes having increased by ??% from version 1.0 to version 3.0. As this growth is expected to continue, scaling considerations for maintenance and performance become crucial. Managing large volumes of data poses challenges not only for search but also for ingestion and indexing, especially with the highly dynamic environmental data incorporated into the graph. To address size-related challenges, the graph can be partitioned into smaller sub-graphs and distributed as shards across a cluster of servers for further processing. Federated SPARQL queries can then aggregate query results from data distributed across these shards (W3C, 2021).

Several sharding approaches exist, and the appropriate method depends on various factors (Saleem et al., 2023). For KWG, where the primary focus is on extracting location insights, queries often target features that are co-located or spatially interacting. Thus, sharding based on locations is a viable approach. This method assumes a pattern of spatial affinity between queries and locations (Ren F & Thomson D, 2018). For instance, identifying population health and available public health facilities in a hurricane-affected region, or retrieving crops grown in areas impacted by wildfire smoke. However, in a rich GeoKG like KWG, data often interact across defined geographical boundaries (e.g., natural phenomena are not circumscribed by administrative boundaries). Therefore, sharding based solely on regional identifiers may not be effective. The key to location-based sharding is to identify data within specific geographies that do not frequently interact with other geographies, thereby reducing the number of distributed joins during federated querying over the shards.

Conversely, geospatial data are naturally divided on a geographic grid. Through topological indexing, conflation, and quantization, KWG data are already clustered on the S2 grid based on locality, providing an ideal key for partitioning by localization. Data belonging to different cells (at an appropriate level) can be hosted on different servers, forming a cluster of shards that can be further indexed (e.g., using Elastic Search). For instance, if the entire KWG data for North America is sharded on the S2 grid at level 2, there would be eight servers corresponding to the eight level 2 cells overlapping North America. ElasticSearch's shard routing can then direct federated queries to the appropriate server.

On the other hand, geospatial data are more naturally divided/segmented on the geographic grid. Through topological indexing, conflating, and quantization, KWG data is already clustered on the S2 grid based on their locality, therefore providing the ideal key for partitioning based on localization. Data belonging to different cells (at an appropriate level) can now be hosted in different servers forming a cluster of shards that can be further indexed (e.g., using Elasticsearch). Assuming the entire KWG data for the U.S. is sharded on S2 Geometry at level 2, then there will be 8 servers corresponding to the eight cells overlapping data in this region – see Figure 17. Shart routing algorithms (such as Elasticsearch's) can then be used to direct a federated query to the appropriate server. It is also necessary to evaluate if representation learning complexity increases when applying embedding models to such grid-based shards separately before merging the outcomes to produce the overall embedding.

6 | CONCLUSION

The implementation of S2 using GeoSPARQL's vector data model as an RDF graph establishes the DGGS as a versatile and adaptable spatial data reference framework that can be seamlessly integrated into any GeoKG infrastructure. This framework eliminates the need for specialized geographic extensions that support representing and querying raster and gridded data within an RDF graph. This approach simplifies the heterogeneous data integration process by utilizing well-supported Simple Feature geometry types and geometry serializations like WKT and GML, which already have optimized indexing and caching mechanisms in graph databases. Connectivity and hierarchy relations between cells are established using GeoSPARQL's topological relations, thereby increasing the qualitative spatial perception of the S2 RDF graph. It also eliminates the need for complex combined vector/raster querying, using cross-geometric operations and raster functions (e.g., raster algebra operations as constructed in GeoSPARQL+ (Homburg, Staab, & Janke, 2020)). The current approach in KWG streamlines all aspects of semantically describing, integrating, and querying heterogeneous geospatial data in RDF by using a simple query pattern based

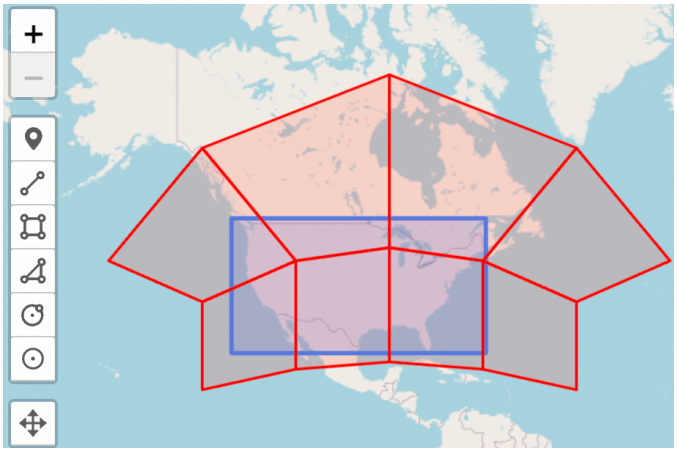


FIGURE 17 Eight S2 cells at level 2, drawn in red, overlap the continental U.S.

on GeoSPARQL. Spatial operations such as geometric operations, spatial statistics, map algebra, and network analysis can now be implemented uniformly across all data using operators and filter functions from SPARQL and GeoSPARQL.

Quantization methods adopted in KWG consider S2 cells as vector bins by which different qualitative and quantitative geospatial information is encapsulated. Statements for this binning are pre-materialized in the graph as RDF triples. By leveraging materialized views based on topologically enriched S2 cells, computationally expensive queries that involve spatial predicates or spatial join operations are optimized for performance. Instead of scanning the entire graph, such queries can now quickly access relevant materialized views corresponding to specific S2 cells, leading to significant improvements in query response times. Discretization transforms spatial data from its original resolution into a finer or more granular format that is directly tagged to cells. This method is particularly suited for representing and querying environmental observation data that are largely remotely sensed raster data, which graph databases are not fully fledged to handle. The geometric and semantic flexibility of the S2 RDF graph makes this data amenable for spatial graph analytics through cell-based raster-to-vector conflation. Further, S2 grid levels at different resolutions combined with explicit relationships between cells in different levels allow for consistent multi-resolution data representation and aggregation but also support the integration and analysis of time-series geospatial data. However, discretization should be approached with prudence, as the quality of the results is highly dependent on the original data accuracy/spatial uncertainty, area distortion accompanying each cell, and the required accuracy of the spatial query. Both through topological enrichment and discretization, various thematic datasets in KWG are cross-linked via stable, portable S2 cell identifiers, which in turn supports compressed data representation by making geometries redundant or obsolete. Furthermore, the modular data model inherent in S2 enables flexible and efficient data management, promoting hardware scalability and sustainable graph maintenance. Their regular, hierarchical grid structure allows for straightforward partitioning of spatial data, making it ideal for distributed computing environments. This capability is particularly beneficial for handling large-scale geospatial datasets, where parallel processing can significantly accelerate data analysis and processing tasks. Grid cells serve as analytic tiles for client-side analytics such as machine learning model development and geovisualization.

Transforming and representing non-DGGS spatial data on a DGGS framework within a GeoKG requires a thorough understanding of the specific DGGS software library being utilized. Depending on the size of the datasets and the extent of the polygon geometries, quantization can demand substantial processing time and computational power, potentially resulting in significant geographic data simplification. The geospatial ecosystem's traditional focus on planar (Euclidean) geometry, in terms of data storage formats, standards, and open-source software libraries, poses challenges when working with spherical geometries like those in S2. Converting data between planar and spherical representations must be handled carefully to maintain accuracy, particularly over large areas. The process of converting from vector/raster formats to a discrete set of DGGS cells involves some data loss. Despite the flexible data model offered by the variable resolution nature of DGGS, S2 cells are not equal-area within each level, posing issues such as the Modifiable Areal Unit Problem. When discretizing information onto S2 cells, it is crucial to specify the spatial uncertainty associated with the geographic data to ensure accuracy and reliability in spatial analyses. Encoding information in DGGS cells also requires specifying spatial uncertainty, which, while beneficial, raises questions about its role in facilitating multiscale analysis. Determining the appropriate level of uncertainty can be challenging, and the impact of spatial uncertainty on decision-makers accustomed to raster and vector data representations remains

unclear. A possible approach to mitigate some of these issues is to use DGGS as a backend reference framework to enhance the representation and computational efficiency of GeoKGs while integrating standard data models for front-end geovisualization applications. This approach could help balance the benefits of DGGS with the practical requirements of traditional geospatial data representation.

This paper presents a detailed methodological approach for utilizing S2 Geometry as an integration framework to semantically reconcile and integrate diverse vector and raster data across various scales and reference systems, addressing the limitations posed by traditional graph databases. The quantization methods adopted in KWG are tailored to specific use cases, yet they outline a broader vision and potential for advanced analytics within the context of GeoKGs. While acknowledging the existing challenges, we aim to engage the broader GIScience research community by presenting open questions that need further investigation. This engagement is crucial for paving the way for more accurate and integrated geospatial analyses within GeoKGs.

AUTHOR CONTRIBUTIONS

This is an author contribution text. This is an author contribution text. This is an author contribution text. This is an author contribution text. This is an author contribution text.

ACKNOWLEDGMENTS

This research has been supported by the National Science Foundation under Grant No. 2033521: "KnowWhereGraph: Enriching and Linking Cross-Domain Knowledge Graphs using Spatially-Explicit AI Technologies". Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

ABBREVIATIONS

The following abbreviations are used in this manuscript:

DE-9IM	Dimensionally Extended 9-Intersection Model
DGG	Discrete Global Grid
DGGS	Discrete Global Grid System
FAIR	Findable, Accessible, Interoperable, Reusable
FIPS	Federal Information Processing Standard
GeoAI	Geospatial Artificial Intelligence
GeoKG	Geospatial Knowledge Graph
GML	Geography Markup Language
KWG	KnowWhereGraph
OGC	Open Geospatial Consortium
OKN	Open Knowledge Network
OWL	Web Ontology Language
NWZ	National Weather Zone
PFAS	Per- and polyfluoroalkyl substances
PM	Particulate Matter
RDF	Resource Description Framework
SAWGraph	Safe Agricultural Products and Water Graph
SSURGO	Soil Survey Geographic Database
SVI	Social Vulnerability Index
UF-OKN	Urban Flooding Open Knowledge Graph
USDA	United States Department of Agriculture
WKT	Well-known Text
ZIP	Zone Improvement Plan

??

SUPPORTING INFORMATION

Additional supporting information and code for this article may be found in the following Github repository: .

References

Abu-Salih, B. (2021). Domain-specific Knowledge Graphs: A survey. *Journal of Network and Computer Applications*, *185*, 103076.

Acharya, A. (2023). *How companies like Uber, Google and Airbnb disrupt industries with location intelligence?* Retrieved from https://medium.com/@abhirup.acharya009/how-companies-like-uber-google-and-airbnb-disrupt-industries-with-location-intelligence-f4fb6ddc3808

Adams, B. (2017). Wāhi, a discrete global grid gazetteer built using linked open data. *International journal of digital earth*, *10*(5), 490–503.

Balla, D., Zichar, M., Tóth, R., Kiss, E., Karancsi, G., & Mester, T. (2020). Geovisualization techniques of spatial environmental data using different visualization tools. *Applied Sciences*, *10*(19), 6701.

Bastrakova, I., & Crossman, S. (2020). Enabling communities to integrate earth, space and environmental data - Australian Location Index. *Earth and Space Science Open Archive ESSOAr*.

Birch, C. P., Oom, S. P., & Beecham, J. A. (2007). Rectangular and hexagonal grids used for observation, experiment and simulation in ecology. *Ecological Modelling*, *206*(3-4), 347–359.

Böckling, M., Paulheim, H., & Detzler, S. (2024). A planet scale spatial-temporal Knowledge Graph based On OpenStreetMap And H3 Grid. *arXiv preprint arXiv:2405.15375*.

Bondaruk, B., Roberts, S. A., & Robertson, C. (2020). Assessing the state of the art in Discrete Global Grid Systems: OGC criteria and present functionality. *Geomatica*, *74*(1), 9–30.

Bousquin, J. (2021). Discrete Global Grid Systems as scalable geospatial frameworks for characterizing coastal environments. *Environmental Modelling & Software*, *146*, 105210.

Bowater, D., & Stefanakis, E. (2019). On the isolatitude property of the rHEALPix Discrete Global Grid System. *Big Earth Data*, *3*(4), 362–377.

Car, N. J., & Homburg, T. (2022). GeoSPARQL 1.1: Motivations, details and applications of the decadal update to the most important geospatial LOD standard. *ISPRS International Journal of Geo-Information*, *11*(2), 117.

Car, N. J. C., Homburg, T., Perry, M., Knibbe, F., Cox, S. J., Abhayaratna, J., ... Janowicz, K. (2023). *OGC GeoSPARQL – A geographic query language for RDF data.* Retrieved from http://www.opengis.net/doc/IS/geosparql/1.1

David Habgood. (n.d.). *RDFLib GeoSPARQL Functions for DGGS.* Retrieved from https://github.com/rdflib/geosparql-dggs

Davis, F., Quattrochi, D., Ridd, M., Lam, N., Walsh, S. J., Michaelsen, J. C., ... Johnston, C. A. (1991). Environmental analysis using integrated GIS and remotely sensed data – some research needs and priorities. *Photogrammetric Engineering and Remote Sensing*, *57*(6), 689–697.

Dinkins, P. (2023). *Quad-tree geospatial data structure: functionality, benefits, and limitations.* Retrieved from https://opensourcegisdata.com/quad-tree-geospatial-data-structure-functionality-benefits-and-limitations.html

Egenhofer, M. J., Mark, D. M., & Herring, J. (1994). The 9-intersection: Formalism and its use for natural-language spatial predicates (94-1). *NCGIA Technical Reports*.

Gangemi, A., & Presutti, V. (2009). Ontology design patterns. In *Handbook on Ontologies* (pp. 221–243). Springer.

Gibb, R. (2016). The rHEALPix Discrete Global Grid System. In *IOP Conference Series: Earth and Environmental Science* (Vol. 34, p. 012012).

Goodchild, M. (1994). Geographical grid models for environmental monitoring and analysis across the globe (panel session). In *Proceedings of GIS/LIS94 Conference.*

Goodchild, M. F. (2000). Discrete global grids for digital earth. In *International Conference on Discrete Global Grids* (pp. 26–28).

Goodchild, M. F. (2018). Reimagining the history of GIS. *Annals of GIS*, *24*(1), 1–8.

Google. (n.d.). *S2 Geometry.* (Date accessed: September 2024). Retrieved from http://s2geometry.io/

Gundeti, V. (2023). *Introducing the Foursquare graph, a pioneer in location intelligence.* Foursquare. Retrieved from https://location.foursquare.com/resources/blog/capabilities/knowledge-graph-technology/

Habgood, D., Homburg, T., Car, N. J., & Jovanovik, M. (2022). Implementation and compliance benchmarking of a DGGS-enabled, GeoSPARQL-aware triplestore. In *Proceedings of the 5th International Workshop on Geospatial Linked Data co-located with ESWC, May 30 2022, Hersonissos, Greece.*

Haller, A., Janowicz, K., Cox, S. J., Lefrançois, M., Taylor, K., Le Phuoc, D., ... Stadler, C. (2019). The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation. *Semantic Web*, *10*(1), 9–32.

Han, B., Qu, T., Huang, Z., Wang, Q., & Pan, X. (2022). Emergency airport site selection using global subdivision grids. *Big Earth Data*, *6*(3), 276–293.

Han, B., Qu, T., Tong, X., Wang, H., Liu, H., Huo, Y., & Cheng, C. (2023). AugGKG: A grid-augmented Geographic Knowledge Graph representation and spatio-temporal query model. *International Journal of Digital Earth*, *16*(2), 4934–4957.

Hitzler, P. (2021). A review of the Semantic Web field. *Commun. ACM*, *64*(2), 76–83. Retrieved from `https://doi.org/10.1145/3397512`

Hitzler, P., Krötzsch, M., & Rudolph, S. (2010). *Foundations of Semantic Web technologies*. Chapman and Hall/CRC Press. Retrieved from `http://www.semantic-web-book.org/`

Hojati, M., Robertson, C., Roberts, S., & Chaudhuri, C. (2022). GIScience research challenges for realizing discrete global grid systems as a Digital Earth. *Big Earth Data*, *6*(3), 358–379.

Homburg, T., Staab, S., & Janke, D. (2020). Geosparql+: Syntax, semantics and system for integrated querying of graph, raster and vector data. In *The Semantic Web–ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part I 19* (pp. 258–275).

Huang, W., Raza, S. A., Mirzov, O., & Harrie, L. (2019). Assessment and benchmarking of spatially enabled RDF stores for the next generation of spatial data infrastructure. *ISPRS International Journal of Geo-Information*, *8*(7), 310.

Ibragimov, D., Hose, K., Pedersen, T. B., & Zimányi, E. (2016). Optimizing aggregate SPARQL queries using materialized RDF views. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I 15* (pp. 341–359).

Iliakis, M. K. (2022). Geospatial Query Answering Using Knowledge Graph Embeddings.

Ioannidis, T., Garbis, G., Kyzirakos, K., Bereta, K., & Koubarakis, M. (2021). Evaluating geospatial RDF stores using the benchmark Geographica 2. *Journal on Data Semantics*, *10*(3), 189–228.

Janowicz, K. (2012). Observation-driven geo-ontology engineering. *Transactions in GIS*, *16*(3), 351–374.

Janowicz, K., Currier, K., Shimizu, C., Zhu, R., Shi, M., Fisher, C. K., ... Stephen, S. (2023). Fast forward from data to insight: (Geographic) Knowledge Graphs and their applications. In *Handbook of Geospatial Artificial Intelligence* (pp. 411–426). CRC Press.

Janowicz, K., Haller, A., Cox, S. J., Le Phuoc, D., & Lefrançois, M. (2019). SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*, *56*, 1–10.

Janowicz, K., Hitzler, P., Li, W., Rehberger, D., Schildhauer, M., Zhu, R., ... others (2022). Know, Know Where, KnowWhereGraph: A densely connected, cross-domain knowledge graph and geo-enrichment service stack for applications in environmental intelligence. *AI Magazine*, *43*(1), 30–39.

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, *33*(2), 494–514.

Jiang, B., You, X., Li, K., Li, T., Zhou, X., & Tan, L. (2020). Interactive analysis of epidemic situations based on a spatiotemporal information knowledge graph of COVID-19. *IEEE Access*, *10*, 46782–46795.

Johnson, J. M., Narock, T., Singh-Mohudpur, J., Fils, D., Clarke, K., Saksena, S., ... Yeghiazarian, L. (2022). Knowledge graphs to support real-time flood impact evaluation. *AI Magazine*, *43*(1), 40–45.

Jovanovik, M., Homburg, T., & Spasić, M. (2021). A GeoSPARQL compliance benchmark. *ISPRS International Journal of Geo-Information*, *10*(7), 487.

Kmoch, A., Vasilyev, I., Virro, H., & Uuemaa, E. (2022). Area and shape distortions in open-source discrete global grid systems. *Big Earth Data*, *6*(3), 256–275.

Kothuri, R. K. V., Ravada, S., & Abugov, D. (2002). Quadtree and R-tree indexes in oracle spatial: a comparison using GIS data. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data* (pp. 546–557).

Kraft, R., Birk, F., Reichert, M., Deshpande, A., Schlee, W., Langguth, B., ... Pryss, R. (2019). Design and implementation of a scalable crowdsensing platform for geospatial data of tinnitus patients. In *2019 IEEE 32nd International Symposium*

on Computer-Based Medical Systems (CBMS) (pp. 294–299).

Kranstauber, B., Weinzierl, R., Wikelski, M., & Safi, K. (2015). Global aerial flyways allow efficient travelling. *Ecology letters*, *18*(12), 1338–1345.

Lan, M., Wu, X., & Theodoratos, D. (2022). Answering Graph Pattern Queries using Compact Materialized Views. In *DOLAP* (pp. 51–60).

Li, M., McGrath, H., & Stefanakis, E. (2021). Integration of heterogeneous terrain data into Discrete Global Grid Systems. *Cartography and Geographic Information Science*, *48*(6), 546–564.

Li, M., McGrath, H., & Stefanakis, E. (2022). Geovisualization of hydrological flow in hexagonal grid systems. *Geographies*, *2*(2), 227–244.

Li, M., & Stefanakis, E. (2020). Geospatial operations of discrete global grid systemsA comparison with traditional GIS. *Journal of Geovisualization and Spatial Analysis*, *4*(2), 26.

Li, W., Wang, S., Chen, X., Tian, Y., Gu, Z., Lopez-Carr, A., ... Zhu, R. (2023). Geographvis: a knowledge graph and geovisualization empowered cyberinfrastructure to support disaster response and humanitarian aid. *ISPRS International Journal of Geo-Information*, *12*(3), 112.

Li, W., Wang, S., Wu, S., Gu, Z., & Tian, Y. (2022). Performance benchmark on semantic web repositories for spatially explicit knowledge graph applications. *Computers, environment and urban systems*, *98*, 101884.

Lin, B., Zhou, L., Xu, D., Zhu, A.-X., & Lu, G. (2018). A discrete global grid system for earth system modeling. *International Journal of Geographical Information Science*, *32*(4), 711–737.

Mahdavi-Amiri, A., Alderson, T., & Samavati, F. (2015). A survey of digital earth. *Computers & Graphics*, *53*, 95–117.

Mahdavi Amiri, A., Alderson, T., & Samavati, F. (2019). Geospatial data organization methods with emphasis on aperture-3 hexagonal discrete global grid systems. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *54*(1), 30–50.

Mai, G., Hu, Y., Gao, S., Cai, L., Martins, B., Scholz, J., ... Janowicz, K. (2022). Symbolic and subsymbolic GeoAI: Geospatial knowledge graphs and spatially explicit machine learning. *Transactions in GIS*, *26*(8), 3118–3124.

Mai, G., Janowicz, K., Zhu, R., Cai, L., & Lao, N. (2021). Geographic question answering: challenges, uniqueness, classification, and future directions. *AGILE: GIScience series*, *2*, 8.

Manley, D. (2021). Scale, aggregation, and the modifiable areal unit problem. In *Handbook of regional science* (pp. 1711–1725). Springer.

Manola, F., Miller, E., McBride, B., et al. (2004). Rdf primer. *W3C recommendation*, *10*(1-107), 6.

Miao, S., Wang, S., Huang, C., Xia, X., Sang, L., Huang, J., ... others (2023). A Big Data Grided Organization and Management Method for Cropland Quality Evaluation. *Land*, *12*(10), 1916.

NSF. (November 2018). *OKN: Open Knowledge Network: Creating the Semantic Information Infrastructure for the Future (Summary of the Big Data IWG Workshop), October 45, 2017.* Retrieved from `https://www.nitrd.gov/news/Open-Knowledge-Network-Workshop-Report-2018.asp`

OGC. (2017). *Topic 21: Discrete Global Grid Systems abstract specification.* Retrieved from `http://docs.opengeospatial.org/as/15-104r5/15-104r5.html`

Papadias, D., Zhang, J., Mamoulis, N., & Tao, Y. (2003). Query processing in spatial network databases. In *Proceedings 2003 vldb conference* (pp. 802–813).

Pereira, R. H., Herszenhut, D., Saraiva, M., & Farber, S. (2024). Ride-hailing and transit accessibility considering the trade-off between time and money. *Cities*, *144*, 104663.

Potoniec, J. (2022). Inductive learning of OWL 2 property chains. *IEEE Access*, *10*, 25327–25340.

Qi, Y., Mai, G., Zhu, R., & Zhang, M. (2023). EVKG: An interlinked and interoperable electric vehicle knowledge graph for smart transportation system. *Transactions in GIS*, *27*(4), 949–974.

Raposo, P. (2019). Geovisualization of complex origin-destination flow maps using Discrete Global Grid Systems. *Abstracts of the ICA*, *1*, 1–3.

Rawson, A., Sabeur, Z., & Brito, M. (2022). Intelligent geospatial maritime risk analytics using the Discrete Global Grid System. *Big Earth Data*, *6*(3), 294–322.

Regalia, B., Janowicz, K., & Gao, S. (2016). VOLT: A provenance-producing, transparent SPARQL proxy for the on-demand computation of linked data and its application to spatiotemporally dependent data. In *The Semantic Web. Latest Advances and New Domains: 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29–June 2, 2016, Proceedings 13* (pp. 523–538).

Regalia, B., Janowicz, K., & McKenzie, G. (2017). Revisiting the Representation of and Need for Raw Geometries on the Linked Data Web. In *LDOW@ WWW*.

Regalia, B., Janowicz, K., & McKenzie, G. (2019). Computing and querying strict, approximate, and metrically refined topological relations in linked geographic data. *Transactions in GIS*, *23*(3), 601–619.

Ren F, L. X., & Thomson D, G. D. (2018). *Geosharded Recommendations Part 1: Sharding Approach.* Tinder Tech Blog. Retrieved from https://tech.gotinder.com/geosharded-recommendations-part-1-sharding-approach-2/

Rigaux, P., Scholl, M., & Voisard, A. (2002). *Spatial databases: with application to GIS*. Morgan Kaufmann.

Riskaware Ltd. (2017). *OpenEAGGR: Open Equal Area Global GRid. Release: 2.0.* Retrieved from https://github.com/riskaware-ltd/open-eaggr

Robertson, C., Chaudhuri, C., Hojati, M., & Roberts, S. A. (2020). An integrated environmental analytics system (IDEAS) based on a DGGS. *ISPRS Journal of Photogrammetry and Remote Sensing*, *162*, 214–228.

Romanov, A. N., & Khvostov, I. V. (2018). Emissivity peculiarities of the inland salt marshes in the south of Western Siberia. *International journal of remote sensing*, *39*(2), 418–431.

Sahr, K., White, D., & Kimerling, A. J. (2003). Geodesic discrete global grid systems. *Cartography and Geographic Information Science*, *30*(2), 121–134.

Saksena, S. (2022). Urban Flooding Open Knowledge Network (UF-OKN): Current products and future functionalities. In *AGU Fall Meeting Abstracts* (Vol. 2022, pp. H46D–02).

Saleem, M., et al. (2023). Storage, indexing, query processing, and benchmarking in centralized and distributed RDF engines: a survey. *Authorea Preprints*.

Shimizu, C., Stephen, S., Zhu, R., Currier, K., Schildhauer, M., Rehberger, D., ... others (2023). The KnowWhereGraph ontology. *Under review*.

Shimizu, C., Zhu, R., Mai, G., Fisher, C., Cai, L., Schildhauer, M., ... Stephen, S. (2021). A Pattern for Features on a Hierarchical Spatial Grid. In *Proceedings of the 10th International Joint Conference on Knowledge Graphs* (pp. 108–114).

Sirdeshmukh, N., Verbree, E., Oosterom, P. v., Psomadaki, S., & Kodde, M. (2019). Utilizing a discrete global grid system for handling point clouds with varying locations, times, and levels of detail. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *54*(1), 4–15.

Strassburg, B. B., Kelly, A., Balmford, A., Davies, R. G., Gibbs, H. K., Lovett, A., ... others (2010). Global congruence of carbon storage and biodiversity in terrestrial ecosystems. *Conservation Letters*, *3*(2), 98–105.

Stroh, E., Harrie, L., & Gustafsson, S. (2007). A study of spatial resolution in pollution exposure modelling. *International journal of health geographics*, *6*, 1–13.

Theocharidis, K., Liagouris, J., Mamoulis, N., Bouros, P., & Terrovitis, M. (2019). Srx: efficient management of spatial rdf data. *The VLDB Journal*, *28*, 703–733.

Timpf, S., & Frank, A. U. (1997). Using hierarchical spatial data structures for hierarchical spatial reasoning. In *Spatial Information Theory A Theoretical Basis for GIS: International Conference COSIT'97 Laurel Highlands, Pennsylvania, USA, October 15–18, 1997 Proceedings 3* (pp. 69–83).

Top, E. J. (2024). Understanding quantities in geo-information in terms of amounts, magnitudes, extents, and intents. , 108–114.

Uber. (2023). *Uber H3: a hexagonal hierarchical geospatial indexing system. Release: 4.1.0.* Retrieved from https://github.com/uber/h3

Uher, V., Gajdoš, P., Snášel, V., Lai, Y.-C., & Radeckỳ, M. (2019). Hierarchical hexagonal clustering and indexing. *Symmetry*, *11*(6), 731.

Veach, E., Rosenstock, J., Engle, E., & Manshreck, T. (2017). S2 Geometry Library: Computational geometry and spatial indexing on the sphere. *Software*. Retrieved from http://s2geometry.io/

W3C. (2021). *Sparql 1.1 federated query.* World Wide Web Consortium. Retrieved from https://www.w3.org/TR/sparql11-federated-query/

Weinberger, G., Scholz, J., & Wandl-Vogt, E. (2022). Towards an intuitive User Interface and Geographic Question Answering for an existing spatial Linked Data Endpoint for Dialect Data. *Abstracts of the ICA*, *5*, 1–3.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., ... others (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data*, *3*(1), 1–9.

World Wide Web Consortium and others. (2013). *SPARQL 1.1 overview.* World Wide Web Consortium. Retrieved from https://www.w3.org/TR/sparql11-query/

Woźniak, S., & Szymański, P. (2021). Hex2vec: Context-aware embedding h3 hexagons with openstreetmap tags. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery* (pp. 61–71).

Yan, B., Mai, G., Hu, Y., & Janowicz, K. (2020). Harnessing Heterogeneous Big Geospatial Data. In *Handbook of Big Geospatial Data* (pp. 459–473). Springer.

Yao, X., Li, G., Xia, J., Ben, J., Cao, Q., Zhao, L., … Zhu, D. (2019). Enabling the big earth observation data via cloud computing and DGGS: Opportunities and challenges. *Remote Sensing*, *12*(1), 62.

Zalewski, J., Hitzler, P., & Janowicz, K. (2021). Semantic Compression with Region Calculi in Nested Hierarchical Grids. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems* (pp. 305–308).

Zhao, Z., Zhang, M., Chen, J., Qu, T., & Huang, G. Q. (2022). Digital twin-enabled dynamic spatial-temporal knowledge graph for production logistics resource allocation. *Computers & Industrial Engineering*, *171*, 108454.

Zhu, R., Cai, L., Mai, G., Shimizu, C., Fisher, C. K., Janowicz, K., … others (2021). Providing humanitarian relief support through knowledge graphs. In *Proceedings of the 11th Knowledge Capture Conference* (pp. 285–288).

Zhu, R., Janowicz, K., Cai, L., & Mai, G. (2022). Reasoning over higher-order qualitative spatial relations via spatially explicit neural networks. *International Journal of Geographical Information Science*, *36*(11), 2194–2225.

Zhu, R., Janowicz, K., Mai, G., Cai, L., & Shi, M. (2022). Covid-forecast-graph: An open Knowledge Graph for consolidating covid-19 forecasts and economic indicators via place and time. *AGILE: GIScience Series*, *3*, 21.

Zhu, R., Stephen, S., Zhou, L., Shimizu, C., Cai, L., Mai, G., … Schildhauer, M. (2021). Environmental Observations in Knowledge Graphs. In *DaMaLOS* (pp. 1–11).