The Missing Piece: Standardising for AI-ready Earth Observation Datasets

Anonymous Authors[1]

Abstract

Geospatial communities have long relied on standardised formats for raster and vector data, enabling interoperability, stable tool development, and long-term data preservation. In contrast, Artificial Intelligence (AI)-ready datasets, particularly those derived from Earth Observation (EO), lack equivalent conventions. As a result, data producers often adopt ad hoc file structures, loosely defined formats, and inconsistent semantic encodings. This fragmentation hinders interoperability, complicates reuse, and undermines reproducibility. We argue that the lack of a standard format represents a structural bottleneck to scalable scientific progress, especially in the era of foundation models, where diverse datasets must be combined for effective training and performance evaluation in downstream tasks. To address this, we introduce TACO: a comprehensive specification that defines a formal data model, a cloud-optimized on-disk layout, and an API for creating and accessing AI-ready EO datasets.

1. Introduction

The rapid increase in Earth Observation (EO) data, combined with advances in AI and cloud computing, has unlocked new opportunities for scientific discovery and operational monitoring (Montillet et al., 2024; Eyring et al., 2024; Hagos et al., 2022). Modern applications range from methane superemitter detection (Vaughan et al., 2024) and burned area estimation (Ribeiro et al., 2023) to biodiversity tracking (Yeh et al., 2021) and global-scale weather forecasting (Rasp et al., 2020; Bi et al., 2023). These efforts increasingly rely on data-driven models, which require large volumes of curated, structured, and accessible EO data (Reichstein et al., 2019). However, preparing AI-ready EO datasets continues to be a significant challenge (Sambasivan et al., 2021; Francis & Czerkawski, 2024). Most datasets require extensive preprocessing and reformatting before they can be integrated into AI pipelines, and only a small fraction are usable "out of the box". Although the number of AI-ready EO datasets has grown substantially, with more than 500 now cataloged (Schmitt et al., 2023), they still lack a unified structure and consistent metadata conventions. This fragmentation hinders reproducibility, limits interoperability, and slows the development of AI (Dimitrovski et al., 2023; Long et al., 2021). These issues are especially critical for training foundation models, which rely on combining diverse sources (Marsocci et al., 2024).

Insights from scientific communities can guide the development of standardised, AI-ready EO datasets. Fields such as climate science and geographic information systems (GIS) have long struggled with data standardisation and provide valuable lessons through widely adopted formats like NetCDF (Treinish & Gough, 1987; Rew & Davis, 1990) and GeoTIFF (Ritter & and, 1997; Devys et al., 2019). NetCDF was initially created as a binary format for scientific data. However, as its use has grown within the climate science community, it became evident that the existing specification did not sufficiently capture the complexity of domain-specific metadata. This realization led to the development of several *metadata conventions*, most notably the CF (Climate and Forecast) Conventions (Eaton et al., 2024), which aimed to standardise the description of scientific variables, coordinates, and attributes. Although these conventions significantly improved interoperability, their *text-based* definitions introduced ambiguities and made consistent implementation difficult. To address this, formal *data models*, such as the CF data model (Hassell et al., 2017), were introduced years later, offering a structured and unambiguous interpretation of what CF-compliant data means. GeoTIFF, in contrast, took a more pragmatic approach. Designed to facilitate the exchange of raster data between GIS applications (Ritter & and, 1997), GeoTIFF embeds minimal but critical metadata, specifically the coordinate reference system (CRS) and geotransform, directly within the file (Devys et al., 2019). GeoTIFF, unlike NetCDF, was not developed with a comprehensive semantic model in mind. However, its simplicity and user-friendly design have led to widespread adoption.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

In hindsight, both cases underscore the importance of maintainability. Crucially, both NetCDF and GeoTIFF have survived because active communities emerged around them, building tools, libraries, and practices that reinforced and extended the specifications over time (Devys et al., 2019; Maso et al., 2023; Eaton et al., 2024). For CF-compliant NetCDF datasets, the experience highlighted the limitations of relying only on text-based definitions: as the authors of the CF data model argue in their conclusion, "*creating an explicit data model before the CF conventions were written would arguably have been preferable. A data model encourages coherent implementations, which could be file storage syntaxes or software codes*" (Hassell et al., 2017). In contrast, GeoTIFF illustrates how a well-defined minimal standard focused on a specific use case can achieve broad interoperability without necessitating a complex data model. These lessons highlight the need to balance formal rigor with practical simplicity. Given the inherent complexity of AI-ready EO datasets, a formal data model is essential; however, whenever possible, it should be designed around the tools and workflows practitioners use on a daily basis to facilitate smooth adoption.

The FAIR principles (Wilkinson et al., 2016), Findability, Accessibility, Interoperability, and Reusability, provide a useful framework to systematically address the challenges faced by the AI-ready EO datasets. Regarding **Findability**, web standardised metadata schemas (i.e., Schema.org, Guha et al. 2016) are rarely used to describe AI-ready EO datasets, limiting their visibility in search engines and data catalogs (Benjelloun et al., 2024). In terms of **Accessibility**, data access often depends on manual downloads or custom APIs rather than scalable, cloud-native formats that support partial or selective retrieval. With respect to **Interoperability**, the wide variety of formats, with differing conventions for byte layout, chunking strategies, compression, and explicit metadata, creates barriers to seamless integration across datasets. Finally, on **Reusability**, many datasets lack clear licenses, provenance, or documentation, making them difficult to audit, cite, or extend.

To close these gaps, we propose TACO (**T**ransparent **A**ccess to **C**loud **O**ptimized Datasets), a FAIR-compliant, cloud-optimized specification for organizing AI-ready EO datasets. TACO files are self-contained, portable, and complete, encapsulating all the information required for sample interpretation without relying on external files or software dependencies. Built on widely supported technologies like GDAL and Apache Parquet, TACO allows for seamless integration across multiple programming languages. The remainder of this paper presents the TACO specification in detail and outlines directions for future development.

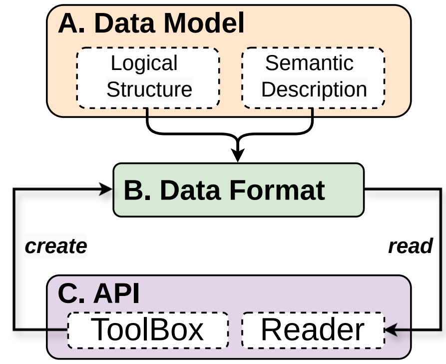

Figure 1. Conceptual organization of the TACO Specification. The Data Model (A) is composed of two layers: Logical Structure (describing the relationships between data and metadata) and Semantic Description (standardised metadata definitions). These layers collectively define the Data Format (B), specifying how data is stored, which can be created and accessed through a dedicated API (C) consisting of the ToolBox (for creation) and the Reader (for reading).

2. Specification

The TACO specification defines the data model, file format, and API (Figure 1). Here, the "data model" refers to an abstract representation of a dataset that defines the rules, constraints, and relationships connecting metadata to the associated data assets (Figure 2). The "data format" defines the physical representation of the dataset, specifying how data and metadata are encoded, stored, and organized. Finally, the API specifies the programmatic methods and conventions by which users and applications can interact with TACO-compliant datasets. By providing a unique and well-structured interface, the API abstracts the underlying complexity of the data format and data model, allowing data users to query, modify, and even integrate multiple TACO datasets. The specifications presented here correspond to version 0.2.0; future versions must remain backward-compatible with this standard.

2.1. Data Model

The logical structure of the TACO data model is illustrated in the UML diagram in Figure 2. At its core, a TACO dataset is defined as a structured collection of minimal self-contained data units, called SAMPLEs, organized within a container, called TORTILLA, and enriched by dataset-level metadata.

A SAMPLE represents the minimal self-contained and smallest indivisible unit for AI training and evaluation. Each SAMPLE encapsulates the actual data and metadata (Figure 4). Importantly, each SAMPLE contains a pointer to a DataSource that specifies how to access the underlying data. TACO supports three primary DataSource types: (i)

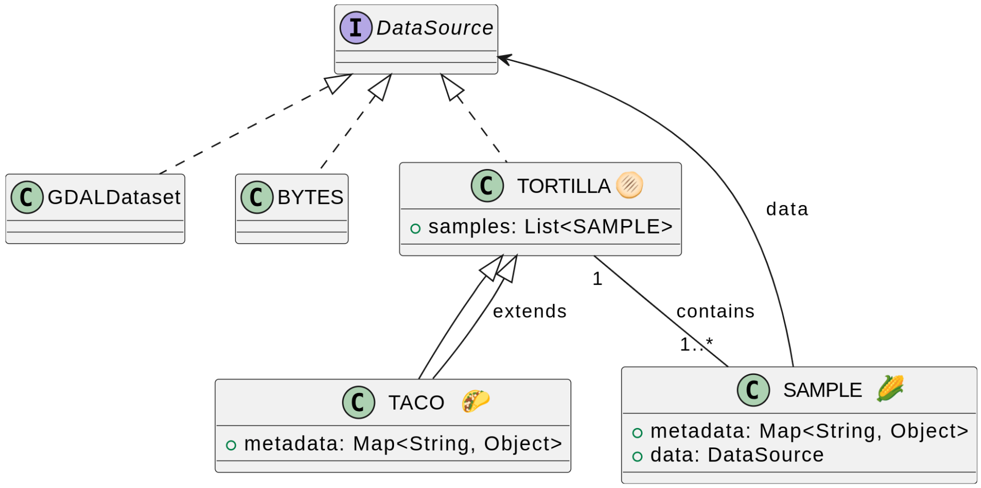

Figure 2. TACO logical structure. A SAMPLE encapsulates raw data and metadata, with a pointer to a DataSource. Supported data sources include GDALDataset, BYTES, and TORTILLA. TACO extends TORTILLA by adding high-level dataset metadata.

GDALDataset, for raster or vector data readable by the GDAL library; (ii) BYTES, representing raw byte streams for unsupported or custom formats; and (iii) TORTILLA. While the BYTES option is available, GDALDataset is recommended for partial read support.

The TORTILLA serves as a container that manages multiple SAMPLE instances. All SAMPLEs within a TORTILLA share a uniform metadata schema, enabling the combined metadata to be represented as a dataframe. Since TORTILLA implements the DataSource interface, it can be referenced within a SAMPLE, enabling recursive nesting of TORTILLA containers. This design supports the representation of hierarchical datasets while preserving the modularity and self-contained nature of individual SAMPLEs.

Building upon TORTILLA, the TACO class extends this container structure by adding comprehensive dataset-level metadata (Figure 5). This additional metadata provides a semantic overview of the collection, supporting dataset management, discovery, and interoperability.

2.2. Data Format

The TORTILLA and TACO file formats are designed for efficient storage of large-scale datasets using a binary serialization scheme (Figure 3). Each TORTILLA file requires a consistent schema and metadata structure in all its samples. Metadata is stored in the FOOTER using Apache Parquet, while the corresponding sample data is stored as a Binary Large Object (BLOB). Each row in the Apache Parquet file corresponds to a different SAMPLE object. The BLOB and the FOOTER are combined within a single file, constituting the TORTILLA format (see Figure 3). Notably, the format allows for partial reads of the BLOB during sample-level access, while the FOOTER is read in full only once during the loading process. A TACO file extends TORTILLA by incorporating additional dataset-level metadata (the COLLECTION), encoded in JSON at the end of the file. This design ensures that both TORTILLA and TACO files are self-contained, portable, and complete, encapsulating all the information required for sample interpretation without relying on external files or software dependencies.

Each file begins with a fixed 200-byte HEADER that includes a 2-byte magic number, an 8-byte offset and length for the FOOTER, and an 8-byte data partition count indicating how many segments the dataset contains. This count allows the TACO API to verify completeness and reconstruct the dataset correctly. TACO files add two more 8-byte fields for the COLLECTION offset and length. Both formats reserve space in the header for future use: 174 bytes in TORTILLA and 158 bytes in TACO.

The TACO API (Section 2.3) automatically generates some fields based on the input data. For example, it records sample-level offsets and lengths in the FOOTER as columns, allowing efficient random access to individual samples (illustrated by the red dotted line in Figure 3). To support multiple programming languages and partial reads, TACO depends on GDAL's Virtual File System (VFS), particularly the /vsisubfile/ handler, which treats byte ranges within a TACO file as standalone GDALDataset objects. This enables random access without reading the BLOB region. TACO also supports cloud-optimized access, adding other GDAL VFS handlers, such as /vsicurl/, /vsis3/, /vsiaz/, /vsigs/, /vsioss/, and /vsiswift/, ensuring high-performance reads across diverse cloud storage platforms.

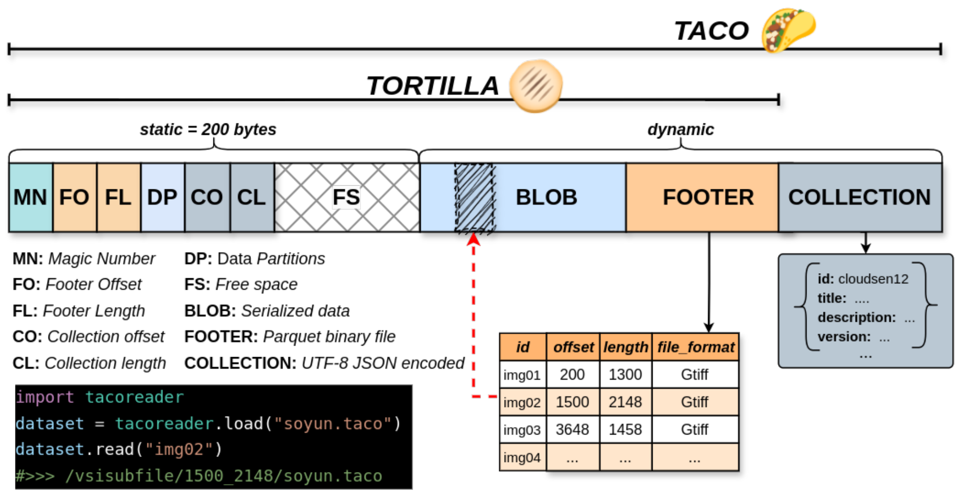

Figure 3. Structure of the TACO and TORTILLA file format, used as the underlying container for SAMPLEs. The static section encodes file-level metadata including a magic number (MN), FOOTER offset (FO) and length (FL), data partition (DP), and pointers to the COLLECTION (CO and CL, only for TACO). The black box illustrates the current API for reading a TACO file: if the SAMPLE is GDAL-readable, the API returns a GDAL virtual file system (VFS) string snippet.

2.3. API

The TACO API consists of two main components: the Toolbox and the Reader. The Toolbox provides constructs for core data classes, SAMPLE, TORTILLA, and TACO, enabling users to define and modify the dataset structure entirely through code. It includes a `create` method, which serializes both data and metadata into fully compliant TACO or TORTILLA files. Additionally, an `edit` method allows users to update existing files, whether they need to adjust the COLLECTION or the FOOTER.

The Reader component provides a simple interface to load and interact with TACO and TORTILLA files. It includes a `load` function that retrieves the FOOTER and, if called with `collection=True`, also returns the COLLECTION. It must also provide a `compile` function that creates smaller subsets of existing TACO or TORTILLA files. The Reader is designed to work with a DataFrame interface in the target programming language (e.g., R, Python, or Julia), where the FOOTER is mapped to a DataFrame object. In addition, a `read` method must be implemented on the DataFrame interface to expose GDAL Virtual File System (VFS) access. For instance, consider the black-box Python code in Figure 3. When 'load' is called, the API converts the FOOTER into a Pandas DataFrame. In the following line, 'read' is invoked. Since the SAMPLEs (each row in the orange table of Figure 3) are in GeoTIFF format, the TACO API generates a GDAL VFS string, which can be interpreted by the GDALOpen class.

3. Discussion and Future work

Several further directions are planned to enhance TACO's usability, performance, and interoperability. One major area of focus is optimizing support for streaming datasets. While TACO already enables partial reads, this approach can be inefficient in nested datasets, since inspecting each sample often results in a separate Parquet read operation, which in cloud environments translates to an additional HTTP GET request per sample. This not only increases latency but also adds operational costs. To mitigate this, future versions will revise the FOOTER layout to consolidate all sample metadata upfront. While this will increase the size of the FOOTER, it will enable data users to have all metadata locally, nested or not, eliminating the need for repeated remote fetches. Another major direction is the introduction of metadata conventions tailored to common EO downstream tasks such as land cover classification, change detection, methane detection, or flood mapping. These conventions enhance consistency and interoperability, and TACO will provide constructors and utilities to create compliant extension metadata.

Finally, we envision developing a shared C/C++ core TACO API designed for interoperability across multiple programming languages, including Python, R, Julia, MATLAB, and JavaScript. This architecture ensures consistent behavior and high performance regardless of the language used. JavaScript API will further support in-browser visualization, enabling users to explore large datasets efficiently. By offering the same API interface across programming languages, TACO can be seamlessly integrated into diverse

AI pipelines, fostering broader adoption and ease of use.

References

Benjelloun, O., Simperl, E., Marcenac, P., Ruyssen, P., Conforti, C., Kuchnik, M., van der Velde, J., Oala, L., Vogler, S., Akthar, M., Jain, N., and Tykhonov, S. Croissant format specification, version 1.0. Technical report, MLCommons, 2024. URL https://docs.mlcommons.org/croissant/docs/croissant-spec.html. Based on Schema.org/Dataset; supported by Hugging Face Datasets.

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., Yan, J., and Tian, Q. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619:533–538, 2023. doi: 10.1038/s41586-023-06185-3.

Devys, E., Habermann, T., Heazel, C., Lott, R., and Rouault, E. OGC geotiff standard, version 1.1. OGC® Implementation Standard 19-008r4, Open Geospatial Consortium, Wayland, MA, USA, September 2019. URL https://docs.ogc.org/is/19-008r4/19-008r4.html. Submission: 2019-06-05; Approval: 2019-09-10. External ID: http://www.opengis.net/doc/IS/GeoTIFF/1.1. Licensed CC0.

Dimitrovski, I., Kitanovski, I., Kocev, D., and Simidjievski, N. Current trends in deep learning for earth observation: An open-source benchmark arena for image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 197:18–35, 2023. doi: 10.1016/j.isprsjprs.2023.01.014.

Eaton, B., Gregory, J., Drach, B., Taylor, K., Hankin, S., Caron, J., Signell, R., Bentley, P., Rappa, G., Höck, H., Pamment, A., Juckes, M., Raspaud, M., Horne, R., Blower, J., Whiteaker, T., Blodgett, D., Zender, C., Lee, D., Hassell, D., Snow, A. D., Kölling, T., Allured, D., Jelenak, A., Soerensen, A. M., Gaultier, L., Herlédan, S., Manzano, F., Bärring, L., Barker, C., and Bartholomew, S. NetCDF climate and forecast (cf) metadata conventions, version 1.12. https://cfconventions.org/Data/cf-conventions/cf-conventions-1.12/cf-conventions.html, December 04 2024. Version 1.12; DOI: 10.5281/zenodo.14275599; CC0 1.0 Universal.

Eyring, V., Collins, W. D., Gentine, P., Barnes, E. A., Barreiro, M., Beucler, T., Bocquet, M., Bretherton, C. S., Christensen, H. M., Dagon, K., et al. Pushing the frontiers in climate modelling and analysis with machine learning. *Nature Climate Change*, pp. 1–13, 2024.

Francis, A. and Czerkawski, M. Major tom: Expandable datasets for earth observation. pp. 2935–2940, 2024. doi: 10.1109/IGARSS53475.2024.10640760.

Guha, R. V., Brickley, D., and Macbeth, S. Schema.org: evolution of structured data on the web. *Communications of the ACM*, 59(2):44–51, 2016.

Hagos, D. H., Kakantousis, T., Sheikholeslami, S., Wang, T., Vlassov, V., Payberah, A. H., Meister, M., Andersson, R., and Dowling, J. Scalable artificial intelligence for earth observation data using hopsworks. *Remote Sensing*, 14(8):1889, 2022. doi: 10.3390/rs14081889.

Hassell, D., Gregory, J., Blower, J., Lawrence, B. N., and Taylor, K. E. A data model of the climate and forecast metadata conventions (CF-1.6) with a software implementation (cf-python v2.1). *Geoscientific Model Development*, 10:4619–4646, 2017. doi: 10.5194/gmd-10-4619-2017.

Long, Y., Xia, G.-S., Li, S., Yang, W., Yang, M., Zhu, X. X., Zhang, L., and Li, D. On creating benchmark dataset for aerial image interpretation: Reviews, guidances and million-aid. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:4205–4230, 2021. doi: 10.1109/JSTARS.2021.3070368.

Marsocci, V., Jia, Y., Bellier, G. L., Kerekes, D., Zeng, L., Hafner, S., Gerard, S., Brune, E., Yadav, R., Shibli, A., et al. Pangaea: A global and inclusive benchmark for geospatial foundation models. *arXiv preprint arXiv:2412.04204*, 2024.

Maso, J., Scripter, Q., Rouault, E., and Sarago, V. Ogc cloud optimized geotiff standard, version 1.0. OGC® Implementation Standard 21-026, Open Geospatial Consortium, Wayland, MA, USA, July 2023. URL https://docs.ogc.org/is/21-026/21-026.html. Submission: 2022-06-29; Approval: 2023-05-08; Copyright © 2023 Open Geospatial Consortium.

Montillet, J.-P., Kermarrec, G., Forootan, E., Haberreiter, M., He, X., Finsterle, W., Fernandes, R., and Shum, C. How big data can help to monitor the environment and to mitigate risks due to climate change: a review. *IEEE Geoscience and Remote Sensing Magazine*, 2024.

Rasp, S., Dueben, P. D., Scher, S., Weyn, J. A., Mouatadid, S., and Thuerey, N. Weatherbench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat, F. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.

Rew, R. and Davis, G. Netcdf: an interface for scientific data access. *IEEE Computer Graphics and Applications*, 10(4):76–82, 1990. doi: 10.1109/38.56302.

Ribeiro, T. F., Silva, F., Moreira, J., and Costa, R. L. d. C. Burned area semantic segmentation: A novel dataset and evaluation using convolutional networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 202:565–580, 2023.

Ritter, N. and and, M. R. The geotiff data interchange standard for raster geographic images. *International Journal of Remote Sensing*, 18(7):1637–1647, 1997. doi: 10.1080/014311697218340. URL https://doi.org/10.1080/014311697218340.

Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., and Aroyo, L. M. "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2021.

Schmitt, M., Ahmadi, S. A., Xu, Y., Taşkin, G., Verma, U., Sica, F., and Hänsch, R. There are no data like more data: Datasets for deep learning in earth observation. *IEEE Geoscience and Remote Sensing Magazine*, 11(3):63–97, 2023.

Treinish, L. A. and Gough, M. L. A software package for the data-independent management of multidimensional data. *Eos, Transactions American Geophysical Union*, 68(28):633–635, 1987.

Vaughan, A., Mateo-García, G., Gómez-Chova, L., Růžička, V., Guanter, L., and Irakulis-Loitxate, I. CH4Net: a deep learning model for monitoring methane super-emitters with Sentinel-2 imagery. *Atmospheric Measurement Techniques*, 17(9):2583–2593, May 2024. ISSN 1867-1381. doi: 10.5194/amt-17-2583-2024. URL https://amt.copernicus.org/articles/17/2583/2024/. Publisher: Copernicus GmbH.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9, 2016.

Yeh, C., Meng, C., Wang, S., Driscoll, A., Rozi, E., Liu, P., Lee, J., Burke, M., Lobell, D., and Ermon, S. Sustainbench: Benchmarks for monitoring the sustainable development goals with machine learning. In *Thirty-fifth Conference on Neural Information Processing Systems, Datasets and Benchmarks Track (Round 2)*, 12 2021. URL https://openreview.net/forum?id=5HR3vCylqD.

A. Semantic description of the SAMPLE metadata

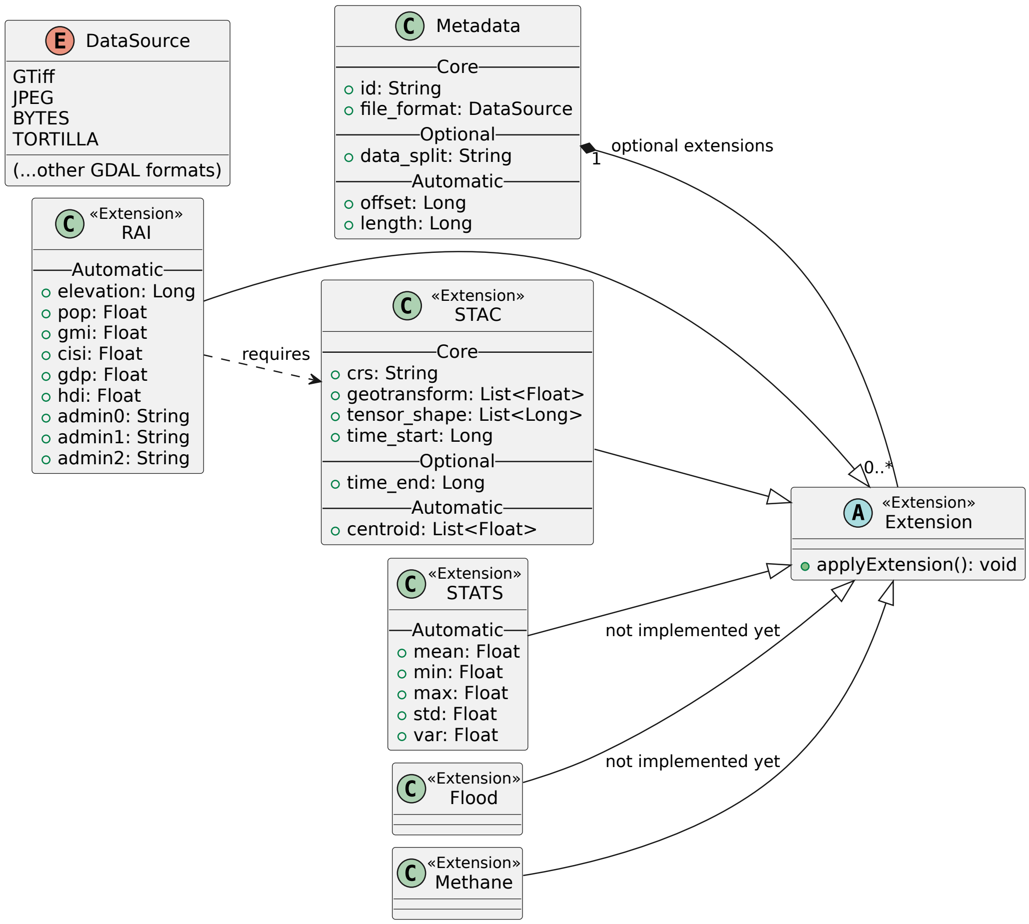

Figure 4. Semantic description of the SAMPLE metadata. The Metadata class contains essential fields for file identification and storage. An abstract Extension class defines the interface for optional metadata, allowing for expansion. Core fields are required, optional fields are user-defined, and automatic fields are generated by the TACO API.

B. Semantic description of the TACO dataset-level metadata.

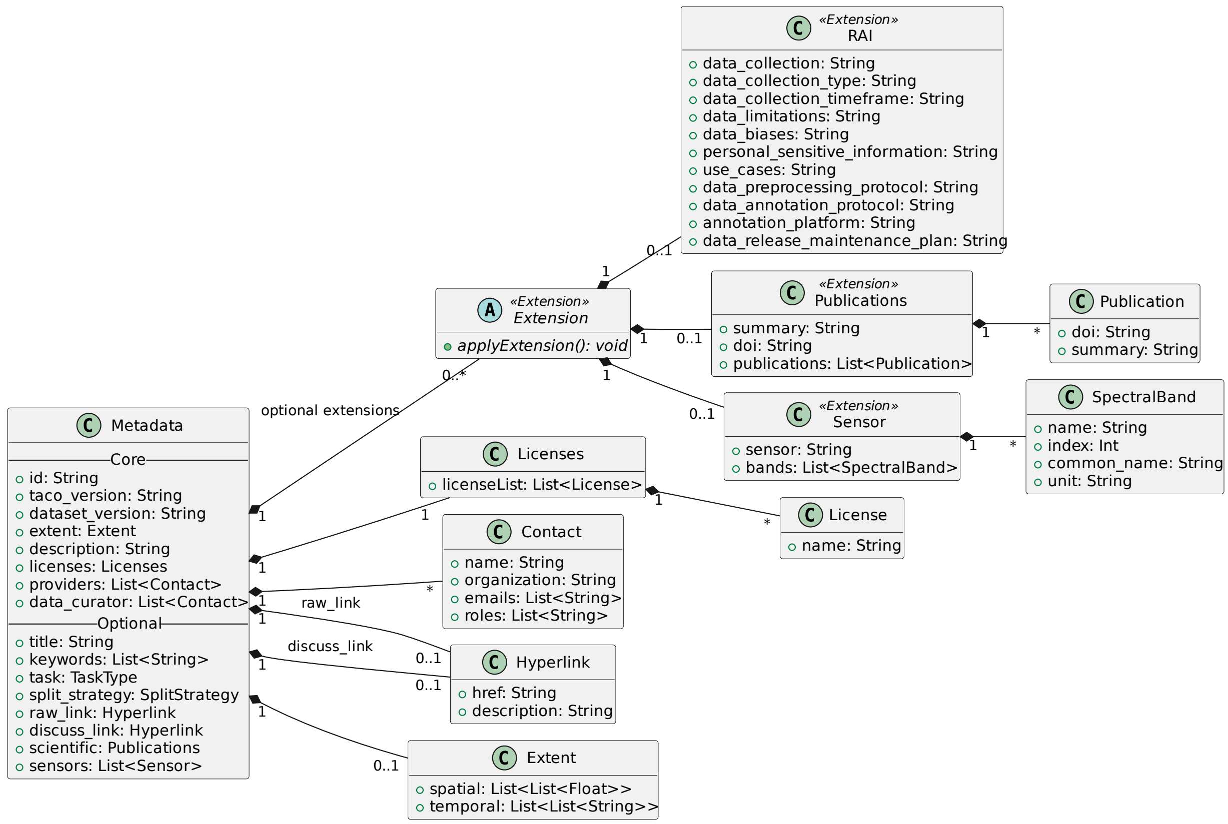

Figure 5. Semantic description of the TACO dataset-level metadata. Core dataset information is structured in the Metadata class, linking core and optional fields. Extensions, modeled through the abstract Extension class, allow modular inclusion of additional metadata.