Journal of Open Humanities Data

Making the Complete OpenAIRE Citation Graph Easily Accessible Through Compact Data Representation

DATA PAPER

JOAKIM SKARDING
PAVEL SANDA
*Author affiliations can be found in the back matter of this article

ubiquity press

ABSTRACT

The OpenAIRE graph contains a large citation graph dataset, with over 200 million publications and over two billion citations. The current graph is available as a dump with metadata which, when uncompressed, totals ~2.5 TB. This makes it hard to process on conventional computers. To make this network more accessible for the community, we provide a processed OpenAIRE graph which is downscaled to 16 GB RAM, while preserving the full graph structure. Apart from this we offer the processed data in a very simple format, which allows for further straightforward manipulation. We also provide (1) a Python pipeline, which can be used to process the next releases of the OpenAIRE graph, and (2) a larger version of the dataset including more publication fields such as the title and list of authors.

CORRESPONDING AUTHOR:
Pavel Sanda
Institute of Computer Science of the Czech Academy of Sciences, Prague, Czech Republic
sanda@cs.cas.cz

KEYWORDS:
citation network; dynamic network; OpenAIRE; large scale network

TO CITE THIS ARTICLE:
Skarding, J., & Sanda, P. (2026). Making the Complete OpenAIRE Citation Graph Easily Accessible Through Compact Data Representation. *Journal of Open Humanities Data*, 12: 63, pp. 1–8. DOI: https://doi.org/10.5334/johd.520

DOI: 10.5334/johd.520

1 OVERVIEW

REPOSITORY LOCATION

We use two repositories: Zenodo to share our data and fixed version of the processing pipeline, and git repository on Codeberg to facilitate sharing and further development of the processing pipeline.

DATA REPOSITORY

https://zenodo.org/records/19207803

CODE REPOSITORY

https://codeberg.org/Zmeos/OpenAIRE-citation-extraction

CONTEXT

The OpenAIRE graph (Manghi et al., 2025; Rettberg & Schmidt, 2012) is a large knowledge graph storing several kinds of research data. In this work we focus on extracting the citations and publications from that dataset, and distilling the information to make it more accessible.

There are several other citations graphs available – OpenAlex (Priem et al., 2022), Open Research Knowledge Graph (Jaradeh et al., 2019), Crossref (Hendricks et al., 2020) and OpenCitations (Peroni & Shotton, 2020). These datasets differ in many ways; some are only citation graphs, while others like OpenAlex and OpenAIRE are research knowledge graphs that record much more than just citations and publications. The datasets also differ in their publication coverage (Culbert et al., 2025; Martín-Martín et al., 2021) and how the works are indexed, e.g., how they determine taxonomy (Ciuciu-Kiss & Garijo, 2024b). Details of publication coverage between OpenAIRE and other open datasets are not well explored – a subset of OpenAIRE has been compared to OpenAlex (Ciuciu-Kiss & Garijo, 2024a), but to the best of our knowledge, no coverage analysis to the extent of previous works like (Martín-Martín et al., 2021) or (Culbert et al., 2025) has taken place. It is not sufficient to compare publication numbers since these datasets aggregate publications from multiple sources and a large node number might simply be due to poor de-duplication.

The OpenAIRE initiative offers several APIs, a search engine, cloud access through Google BigQuery, and a complete dataset dump (Manghi et al., 2025). The API and search engine serve as accessible avenues for any researcher, however, they are limited in the scope of data that can be efficiently accessed. A scan of the complete graph is outside the free tier of BigQuery, at the time of writing. If a researcher is interested in large portions of the graph, or the graph as a whole, then the dump is more suitable.

Processing the whole graph dump requires software coding skills, a large amount of memory, storage and computational power, resources not readily accessible to many scholars in the humanities (Dederke et al., 2024). Accessing the raw data also requires solid understanding of the OpenAIRE JSON schema. To fill this gap, we pre-process the dump, distilling it into a more manageable size, and distributing it as simply structured plain text and Parquet files, making the whole graph more accessible. Similar work has also been done for OpenAlex (Caetano Machado Lopes & Chacko, 2024).

2 METHOD

The OpenAIRE graph consists of many entities and many relations. These are stored in many files in the dump. We only need the publication files (nodes) and the relation files (citations/edges) to extract the citation network. The publication files themselves contain a lot of information about publications, only some of which is relevant to most researchers. The relation files include many types of relations. For this dataset we are only interested in citations (the type “Cites”). The following list shows the generic steps we followed to extract the citation network and minimize memory footprint. The list is organized by python files and shows the file responsible for each step (the full code is available in the Codeberg repository).

STEPS

0. `step0_download_data_and_extract.py`

   Download the complete OpenAIRE graph dump (Manghi et al., 2025), including all publication and relation files. When calling the pipeline, this is an optional step.

1. `step1_extract_raw.py`

   Partially extract the compressed dump files.

2. `step2_publications.py`

   (a) Filter records, obtaining publication-type only.

   (b) Flatten nested JSON structures to obtain a tabular representation.

   (c) Generate new, more memory-efficient(`int32`) nodeIDs, and build a hashtable for translating between OpenAIRE IDs and our nodeIDs.

   (d) For the TSV files, replace any tabs or newlines with whitespace.

   (e) Export TSV and Parquet publication files.

3. `step3_citations.py`

   (a) Process relation files using PySpark (Zaharia et al., 2016), retaining only relations of type `Cites` and extracting source and target identifiers.

   (b) Use hashtable produced by `step2_publications.py` to replace OpenAIRE IDs. Update all citation relations to reference the new identifiers.

   (c) Export TSV and Parquet citation files.

4. `step4_distribution.py`

   Compresses the TSV files to `xz`.

This achieves a distilled dataset where the relations are efficiently stored using pairs of short integers.

QUALITY CONTROL

Several validation scripts are run after processing to control for mistakes in the processing. The scripts check whether any entries from the original dump were lost and report any missing values.

- `full_id_coverage.py` Verifies that every publication in the raw source data is present in the final output.
- `distinct_constraints.py` Checks that there are no duplicate publications or duplicate citation edges in the output.
- `format_checks.py` Verifies that publication node IDs form a complete, gap-free sequence starting from zero.

Additionally the file `run_all_validations.py` runs all validation scripts, collects their results, and writes a consolidated summary report. The entire validation pipeline is automatically run after the processing pipeline completes. Output from the validation files are shown in Table 1.

Table 1 Combined validation script output. This is the automated output log of the validation scripts.

SCRIPT	METRIC	VALUE
full_id_coverage (pass)		
	raw_count	205841448
	parquet_count	205841448
	missing_count	0
	extra_count	0

(Contd.)

SCRIPT	METRIC	VALUE
distinct_constraints (pass)		
	publications total	205841448
	publications distinct	205841448
	publications duplicates	0
	citations total	2184347684
	citations distinct	2184347684
	citations duplicates	0
	citations null_rows	0
format_checks (pass)		
	nodeid gaps	0
	nodeid contiguous	true

OUTPUT

The processing pipeline takes the full OpenAIRE dump as input and transforms it into the TSV (tab-separated values) and equivalent Parquet files described below. TSV is a variant of a more common CSV. We chose TSV because several of the text fields in the publications_large.tsv routinely contained commas, and our attempts at quoting the commas turned out to be error prone. Parquet is a binary format for high-performance data processing.

- `citations.tsv.xz` and `citations.parquet` – a simple edge list of the graph (all the citations). Each row has a simple form of two connected nodeIDs, e.g.: `159486578 118392581`
- `publications.tsv.xz` and `publications.parquet` – all graph nodes (publications), it contains only the nodeID and, when available, the DOI. Each row has a simple form, e.g.: `14209 10.3931/e-rara-45685`
- `publications_large.tsv.xz` and `publications_large.parquet` – includes the same number of nodes as the publication files, but includes additional fields, e.g. title, authors, description, etc. For a full overview of fields, see Table 3.
- `pipeline.tar.xz` – processing pipeline which transforms full OpenAIRE dump into the dataset files above. Full details are documented in included `README.md`. The source code is also present online at https://codeberg.org/Zmeos/OpenAIRE-citation-extraction.

The `pid_*` columns (included in publications_large, see Table 3) are only extracted schemes from the `pids` field in the dump which OpenAIRE populates with identifiers collected from authoritative sources. The instances field (a nested field of additional metadata, Table 5) also includes identifiers, those were not collected. An exception to this is the `primary_doi` field in the publications file, in which the doi is drawn from the identifiers field if it is not found in the pids field.

MEMORY REQUIREMENTS

Below, we summarize the hardware requirements for the transformed dataset and the original OpenAire full dataset dump. An overview of memory and disk requirements are shown in Table 2. For loading the citations.tsv into memory with `int32` using the Pandas library in Python, the following can be used:

```
df_refs = pd.read_csv(
    “citations.tsv.xz”, sep=“\t”,
    dtype={“source_nodeId”: “int32”, “target_nodeId”: “int32”}
)
```

The data formats (TSV and Parquet) are of course generic and can be loaded using any tool that supports these common formats. When loading the Parquet, `int32` will be used automatically.

```
df_cites = pd.read_parquet(
    “citations.parquet”,
    engine=“pyarrow”,
    dtype_backend=“pyarrow”,
)
```

The publications files benefit from being loaded using the PyArrow backend. This significantly reduces the memory usage of the doi field.

```
df_pubs = pd.read_parquet(
    “publications.parquet”,
    engine=“pyarrow”,
    dtype_backend=“pyarrow”,
)
```

The publications_large files are the most important to load efficiently as ~200GB of RAM can be saved. Be aware that even efficient loading still requires ~185GB of RAM. For users interested in a subset of the provided columns, we recommend selecting the columns on load. For loading the full dataset the following code can be used, and selected columns can be removed. This efficient loading is the approach used for the “Pandas Opt” column in Table 3. In the example below, the columns nodeId, title and pid_dois are selected.

Table 2 Comparison of storage size and memory requirements of the full OpenAIRE (OA) dataset (release 2025-12-01) and its compact versions. Memory usage corresponds to the amount of GB each dataset occupied when loaded into a Pandas dataframe (The pandas development team, 2026). Since the citations are loaded as int32, the memory size is much lower than the disk size.

DATASET	SIZE ON DISK (GB)	MEMORY USAGE (GB)
citations.tsv	39	17
publications.tsv	6	5
publications_large.tsv	187	185
citations.parquet	8	17
publications.parquet	2	5
publications_large.parquet	68	185
Full OA – edges	1820	NA
Full OA – nodes	700	NA

```
df_large = pd.read_parquet(
    “publications_large.parquet”,
    columns=[“nodeId”, “title”, “pid_dois”],
    engine=“pyarrow”,
    dtype_backend=“pyarrow”,
)
df_large[[“language”, “container”]] = (
    df_large[[“language”, “container”]].astype(“category”)
)
```

Table 3 Memory size of columns with Python types and short descriptions. Arrow is the memory size loaded using PyArrow; Pandas is the size if the data is loaded straight into a default Pandas dataframe. The Pandas Optimized column uses PyArrow as a backend, and utilizes “categorical” on the container and language fields to further lower their footprint. The MAG IDs are Microsoft Academic Graph IDs.

COLUMN	PYTHON TYPE	DESCRIPTION	ARROW	PANDAS	PANDAS OPT
nodeId	int32	Node ID	0.767 GB	0.767 GB	0.767 GB
openaireId	str	OpenAIRE unique ID	9.585 GB	19.746 GB	9.586 GB
title	str	Paper title	16.486 GB	29.707 GB	16.486 GB
authors	list[str]	List of authors	11.037 GB	23.005 GB	11.038 GB
description	str	Main text/abstract	131.248 GB	193.583 GB	131.248 GB
date	datetime	Publication date	0.767 GB	7.587 GB	0.768 GB
container	str	Journal/conference name	5.471 GB	13.953 GB	2.181 GB
citations	int	Citation count	1.558 GB	1.534 GB	1.558 GB
language	str	Language	1.394 GB	11.555 GB	0.197 GB
pid_dois	list[str]	DOI identifiers	5.639 GB	19.453 GB	5.639 GB
pid_mag_ids	list[str]	MAG IDs	2.004 GB	12.748 GB	2.005 GB

(Contd.)

COLUMN	PYTHON TYPE	DESCRIPTION	ARROW	PANDAS	PANDAS OPT
pid_pmids	list[str]	PubMed IDs	1.202 GB	7.948 GB	1.202 GB
pid_handles	list[str]	Persistent handles	1.149 GB	6.134 GB	1.149 GB
pid_pmcs	list[str]	PubMed Central IDs	0.921 GB	5.480 GB	0.921 GB
pid_arxiv_ids	list[str]	ArXiv IDs	0.885 GB	4.856 GB	0.886 GB
TOTAL			190.113 GB	358.055 GB	185.631 GB

3 DATASET DESCRIPTION

REPOSITORY NAME

Zenodo

OBJECT NAME

Compact representation of the OpenAIRE citation graph

FORMAT NAMES AND VERSIONS

TSV (Tab Separated Values) and Apache Parquet

CREATION DATES

Based on the 2025-12-01 OpenAIRE dump.

DATASET CREATORS

Joakim Skarding wrote the pipeline that processed the dump, Pavel Sanda supervised the work. The OpenAIRE initiative, with which this project is unaffiliated, created the OpenAIRE dump.

LANGUAGE

English

LICENSE

CC BY 4.0

PUBLICATION DATE

2026-02-12

4 REUSE POTENTIAL

Citation networks are routinely used in a wide range of scientific fields. This includes among others: research in historical trends of science (Drivas, 2024; Frank et al., 2019; González-Márquez et al., 2024; Kitajima & Okamura, 2025), sociology of scientific knowledge (Carradore, 2022; Crothers et al., 2020), network science (Costa & Frigori, 2024; Xiao et al., 2025), and as training sets for graph neural networks (Kipf, 2016; Leskovec & Sosič, 2016). Since the dataset is an evolving network, it can also be used to train temporal models, such as dynamic graph neural networks (Skarding et al., 2021).

Compared to working directly with the original OpenAIRE data dump, this dataset significantly reduces the time and effort required to begin analysis. The data is provided as flat TSV and Parquet files, removing the need to parse and process the raw JSON. The citation graph is already structured as an edge list with integer node IDs, the format expected by most graph libraries, meaning graph algorithms (e.g., community detection (Fortunato, 2010) and centrality measures (Bloch et al., 2023)) can be applied directly without further transformation. The reduced file size allows the full dataset to be downloaded and explored locally. Together, these properties make the dataset a convenient starting point for bibliometric studies, network analysis, and graph learning research.

We provide the source code used to produce the data, allowing researchers to run the pipeline when new versions of the OpenAIRE graph is released, as well as make their own customized distilled dataset with the fields they desire.

ADDITIONAL FILE

The additional file for this article can be found as follows:

- **Supplementary file.** Supplementary information. DOI: https://doi.org/10.5334/johd.520.s1

FUNDING STATEMENT

This work has been funded by a grant from the Programme Johannes Amos Comenius under the Ministry of Education, Youth and Sports of the Czech Republic, CZ.02.01.01/00/23_025/0008711.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Joakim Skarding orcid.org/0000-0001-8509-658X
Institute of Computer Science of the Czech Academy of Sciences, Prague, Czech Republic
Pavel Sanda orcid.org/0000-0002-2554-0180
Institute of Computer Science of the Czech Academy of Sciences, Prague, Czech Republic

REFERENCES

Bloch, F., Jackson, M. O., & **Tebaldi, P.** (2023). Centrality measures in networks. *Social Choice and Welfare*, *61*(2), 413–453. https://doi.org/10.1007/s00355-023-01456-4

Caetano Machado Lopes, L., & **Chacko, G.** (2024). *A Citation Graph from OpenAlex (Works)*. University of Illinois Urbana-Champaign. https://doi.org/10.13012/B2IDB-7362697_V1

Carradore, M. (2022). Academic research output on social capital: a bibliometric and visualization analysis. *International Journal of Sociology and Social Policy*, *42*(13/14), 113–134. https://doi.org/10.1108/IJSSP-11-2022-0281

Ciuciu-Kiss, J. T., & **Garijo, D.** (2024a). Assessing the overlap of science knowledge graphs: A quantitative analysis. In *International workshop on natural scientific language processing and research knowledge graphs* (pp. 171–185). https://doi.org/10.1007/978-3-031-65794-8_11

Ciuciu-Kiss, J. T., & **Garijo, D.** (2024b). Assessing the overlap of science knowledge graphs: A quantitative analysis. In G. Rehm, S. Dietze, S. Schimmler, & F. Krüger (Eds.), *Natural scientific language processing and research knowledge graphs* (pp. 171–185). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-65794-8_11

Costa, A. A., & **Frigori, R. B.** (2024). Complexity and phase transitions in citation networks: insights from artificial intelligence research. *Frontiers in Research Metrics and Analytics*, *9*, 1456978. https://doi.org/10.3389/frma.2024.1456978

Crothers, C., Bornmann, L., & **Haunschild, R.** (2020). Citation concept analysis (CCA) of Robert K. Merton's book Social Theory and Social Structure: How often are certain concepts from the book cited in subsequent publications? *Quantitative Science Studies*, *1*(2), 675–690. https://doi.org/10.1162/qss_a_00029

Culbert, J. H., Hobert, A., Jahn, N., Haupka, N., Schmidt, M., Donner, P., & **Mayr, P.** (2025). Reference coverage analysis of OpenAlex compared to web of science and scopus. *Scientometrics*, *130*(4), 2475–2492. https://doi.org/10.1007/s11192-025-05293-3

Dederke, J., Koch, M., & **Willemin, S.** (2024). The representation of Swiss higher education institutions in five bibliometric databases. *Qualität in der Wissenschaft*, *2024*(4), 117–124. https://doi.org/10.3929/ethz-b-000726102

Drivas, K. (2024). The evolution of order of authorship based on researchers' age. *Scientometrics*, *129*(9), 5615–5633. https://doi.org/10.1007/s11192-024-05124-x

Fortunato, S. (2010). Community detection in graphs. *Physics reports*, *486*(3-5), 75–174. https://doi.org/10.1016/j.physrep.2009.11.002

Frank, M. R., Wang, D., Cebrian, M., & **Rahwan, I.** (2019). The evolution of citation graphs in artificial intelligence research. *Nature Machine Intelligence*, *1*(2), 79–85. https://doi.org/10.1038/s42256-019-0024-5

González-Márquez, R., Schmidt, L., Schmidt, B. M., Berens, P., & **Kobak, D.** (2024). The landscape of biomedical research. *Patterns*, *5*(6). https://doi.org/10.1016/j.patter.2024.100968

Hendricks, G., Tkaczyk, D., Lin, J., & **Feeney, P.** (2020). Crossref: The sustainable source of community-owned scholarly metadata. *Quantitative Science Studies*, *1*(1), 414–427. https://doi.org/10.1162/qss_a_00022

Jaradeh, M. Y., Oelen, A., Farfar, K. E., Prinz, M., D'Souza, J., Kismihók, G., … Auer, S. (2019). Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. In *Proceedings of the 10th international conference on knowledge capture* (pp. 243–246). https://doi.org/10.1145/3360901.3364435

Kipf, T. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*. https://doi.org/10.48550/arXiv.1609.02907

Kitajima, K., & **Okamura, K.** (2025). The altering landscape of us–china science collaboration: from convergence to divergence. *Humanities and Social Sciences Communications*, *12*(1), 1–11. https://doi.org/10.1057/s41599-025-04550-3

Leskovec, J., & **Sosič, R.** (2016). Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8*(1), 1–20. https://doi.org/10.1145/2898361

Manghi, P., Atzori, C., Bardi, A., Baglioni, M., Dimitropoulos, H., La Bruzzo, S., … Chatzopoulos, S. (2025, September). *OpenAIRE graph dataset.* OpenAIRE. https://doi.org/10.5281/zenodo.17098012

Martín-Martín, A., Thelwall, M., Orduna-Malea, E., & **Delgado López-Cózar, E.** (2021). Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations' COCI: a multidisciplinary comparison of coverage via citations. *Scientometrics*, *126*(1), 871–906. https://doi.org/10.1007/s11192-020-03690-4

Peroni, S., & **Shotton, D.** (2020). Opencitations, an infrastructure organization for open scholarship. *Quantitative Science Studies*, *1*(1), 428–444. https://doi.org/10.1162/qss_a_00023

Priem, J., Piwowar, H., & **Orr, R.** (2022). OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. *arXiv preprint arXiv:2205.01833*. https://doi.org/10.48550/arXiv.2205.01833

Rettberg, N., & **Schmidt, B.** (2012). OpenAIRE-Building a collaborative open access infrastructure for european researchers. *LIBER Quarterly: The Journal of the Association of European Research Libraries*, *22*(3), 160–175. https://doi.org/10.18352/lq.8110

Skarding, J., Gabrys, B., & **Musial, K.** (2021). Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *IEEE Access*, *9*, 79143–79168. DOI: 10.1109/ACCESS.2021.3082932

The pandas development team. (2026, January). *pandas-dev/pandas: Pandas.* Zenodo. https://doi.org/10.5281/zenodo.18328522

Xiao, Z., Fan, L., Yu, Z., & **Liu, X.** (2025). Characterizing pandemic-related publications: a retrospective study using spatial citation network analysis. *Computational Urban Science*, *5*(1), 25. https://doi.org/10.1007/s43762-025-00184-y

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., … others (2016). Apache spark: a unified engine for big data processing. *Communications of the ACM*, *59*(11), 56–65. https://doi.org/10.1145/2934664

TO CITE THIS ARTICLE:
Skarding, J., & Sanda, P. (2026). Making the Complete OpenAIRE Citation Graph Easily Accessible Through Compact Data Representation. *Journal of Open Humanities Data*, 12: 63, pp. 1–8. DOI: https://doi.org/10.5334/johd.520

Submitted: 13 February 2026
Accepted: 10 April 2026
Published: 30 April 2026

Journal of Open Humanities Data is a peer-reviewed open access journal published by Ubiquity Press.