From Static Repositories to Agentic Knowledge Webs: Implementing the S-index via Bimodal Glial-Neural Optimization for Human-AI Research Partnerships

Martin Frasch[1]

[1]Independent Researcher, martin@researchtwin.net

February 2026

Abstract

The exponential growth of scientific literature, datasets, and code repositories has created a *discovery bottleneck* that impedes knowledge synthesis, reproducibility, and cross-disciplinary collaboration. Traditional dissemination formats—static PDFs, siloed code hosting, and fragmented data repositories—fail to represent the interconnected narrative of modern research. Conventional impact metrics such as the H-index measure citation counts alone, neglecting the substantial contributions of reusable code and shared datasets. We introduce **ResearchTwin**, an open-source federated platform that transforms a researcher's complete scholarly output into a conversational digital twin. The system is built on a novel *Bimodal Glial-Neural Optimization* (BGNO) architecture comprising a Multi-Modal Connector Layer, a Glial Layer for caching and rate management, and a Neural Layer implementing Retrieval-Augmented Generation with Claude. We formalize the **S-index**, a composite metric grounded in the Quality–Impact–Collaboration (QIC) framework that extends FAIR principles to quantify the multimodal impact of research artifacts. ResearchTwin exposes an inter-agentic discovery API using Schema.org typed responses, enabling autonomous AI agents to navigate researcher profiles via HATEOAS links and discover cross-lab synergies. A three-tier federated architecture—Local Nodes, Hubs, and Hosted Edges—preserves data sovereignty while enabling global discoverability. The platform is released as open-source software at https://github.com/martinfrasch/ResearchTwin.

Keywords: digital twin, research impact metrics, FAIR principles, retrieval-augmented generation, federated architecture, scholarly communication, inter-agentic discovery

1 Introduction

The pace of scientific publication has accelerated to an unprecedented scale. In 2024 alone, an estimated 3.5 million peer-reviewed articles were indexed by major databases, accompanied by millions of datasets deposited in repositories such as Figshare and Zenodo, and hundreds of thousands of research-related code repositories created on GitHub. While this growth reflects vibrant research activity, it simultaneously creates a *discovery bottleneck*: individual researchers cannot keep pace with the literature relevant to their own sub-fields, let alone identify cross-disciplinary opportunities for data reuse or code integration.

Current dissemination practices exacerbate this problem. Research narratives remain fragmented across static PDF documents, isolated code repositories, and metadata-sparse data archives. A

given project's full contribution—the paper describing the method, the code implementing it, and the dataset validating it—is scattered across platforms with no machine-readable links connecting these artifacts. Discovering that a particular dataset pairs naturally with a codebase from another lab requires serendipity rather than systematic retrieval.

Impact measurement compounds the issue. The H-index [**?**] has served as the dominant measure of research productivity for two decades, yet it captures only citation-based influence among publications. As research increasingly produces reusable code (via GitHub) and shared data (via Figshare, Zenodo, or Dryad), a significant portion of scholarly impact goes unmeasured. A highly-cited paper whose accompanying code has been forked thousands of times and whose dataset has been downloaded by hundreds of labs has a fundamentally different impact profile from one with equivalent citations but no reusable artifacts—yet the H-index treats them identically.

We argue that the next generation of scholarly infrastructure must satisfy three requirements: (1) **multi-modal integration**, unifying papers, code, and data into a single queryable representation; (2) **conversational access**, enabling both human researchers and AI agents to explore research artifacts through natural language; and (3) **composite impact measurement**, quantifying the full spectrum of contributions including code utility and data reuse.

This paper presents **ResearchTwin**, a federated platform that addresses all three requirements. The system is built on a *Bimodal Glial-Neural Optimization* (BGNO) architecture inspired by the separation of metabolic support and signal processing in biological neural tissue. We formalize the **S-index**, a composite metric that extends FAIR principles [**?**] into a quantitative Quality–Impact–Collaboration (QIC) framework. We describe an inter-agentic discovery API using Schema.org types that enables autonomous AI agents to traverse researcher profiles and discover cross-lab synergies. The platform is released as open-source software under the MIT license.

The remainder of this paper is organized as follows. Section **??** surveys related work. Section **??** presents the BGNO architecture. Section **??** formalizes the S-index. Section **??** describes the inter-agentic discovery protocol. Section **??** details the federated architecture. Section **??** covers implementation. Section **??** discusses advantages and limitations. Section **??** outlines future work, and Section **??** concludes.

2 Related Work

2.1 Digital Twins in Manufacturing and Beyond

The Digital Twin concept originated in manufacturing, where Grieves [**?**] proposed maintaining a virtual replica of a physical product throughout its lifecycle. A Digital Twin mirrors the state, behavior, and context of its physical counterpart, enabling simulation, monitoring, and optimization. This paradigm has since expanded to healthcare, urban planning, and infrastructure management. We adapt the concept to the research domain: a *Research Digital Twin* mirrors a researcher's scholarly identity by integrating their publications, code, datasets, and impact metrics into a dynamic, queryable representation.

2.2 Research Impact Metrics

Hirsch's H-index [**?**] remains the most widely used measure of individual research productivity, defined as the largest number h such that at least h papers have each been cited at least h times. While elegant, the H-index has well-documented limitations: it ignores citation context, penalizes

early-career researchers, and—crucially for our purposes—captures only publication-based impact. The i10-index (papers with ≥ 10 citations) shares these limitations. Field-normalized alternatives such as the Field-Weighted Citation Impact improve cross-disciplinary comparability but remain confined to citation counts. No widely adopted metric integrates code reuse (e.g., GitHub stars and forks) or dataset downloads into a unified impact score.

2.3 FAIR Data Principles

Wilkinson et al. [**?**] introduced the FAIR principles—Findability, Accessibility, Interoperability, and Reusability—as guiding criteria for scientific data management. FAIR has become a cornerstone of open science policy, influencing repository design, funder requirements, and metadata standards. However, FAIR compliance is typically assessed qualitatively or through binary checklists. Our QIC framework operationalizes FAIR into continuous numerical scores that feed directly into the S-index computation.

2.4 Retrieval-Augmented Generation

Lewis et al. [**?**] introduced Retrieval-Augmented Generation (RAG), combining parametric language models with non-parametric retrieval to ground generated text in external knowledge. RAG architectures have since been applied to question answering, code generation, and domain-specific chatbots. ResearchTwin employs RAG to synthesize answers from a researcher's multi-modal context, positioning the language model as a conversational proxy for the researcher's body of work.

2.5 Scholarly Knowledge Graphs and Communication Standards

Schema.org provides a shared vocabulary for structured data markup, including types relevant to scholarly communication: `Person`, `ScholarlyArticle`, `Dataset`, and `SoftwareSourceCode`. Projects such as OpenAIRE, Semantic Scholar, and the Microsoft Academic Graph have built large-scale knowledge graphs over scholarly metadata. ORCID provides persistent researcher identifiers. Our contribution is to combine these elements into a live, conversational system with a formal impact metric and an agent-navigable API layer.

2.6 Federated Systems

Federated architectures distribute control across autonomous nodes while enabling global interoperability. ActivityPub powers decentralized social networks (e.g., Mastodon), demonstrating that federation can scale to millions of users while preserving data sovereignty. In the commercial domain, Discord's server model allows communities to operate independently while sharing a common protocol. ResearchTwin adopts a three-tier federation model inspired by these systems, enabling researchers to host their own nodes while remaining discoverable through a shared protocol.

3 Architecture: Bimodal Glial-Neural Optimization

The BGNO architecture separates concerns into three layers, inspired by the functional division between glial cells (metabolic support, homeostasis) and neurons (signal processing, computation) in biological neural tissue. Figure **??** provides an overview.

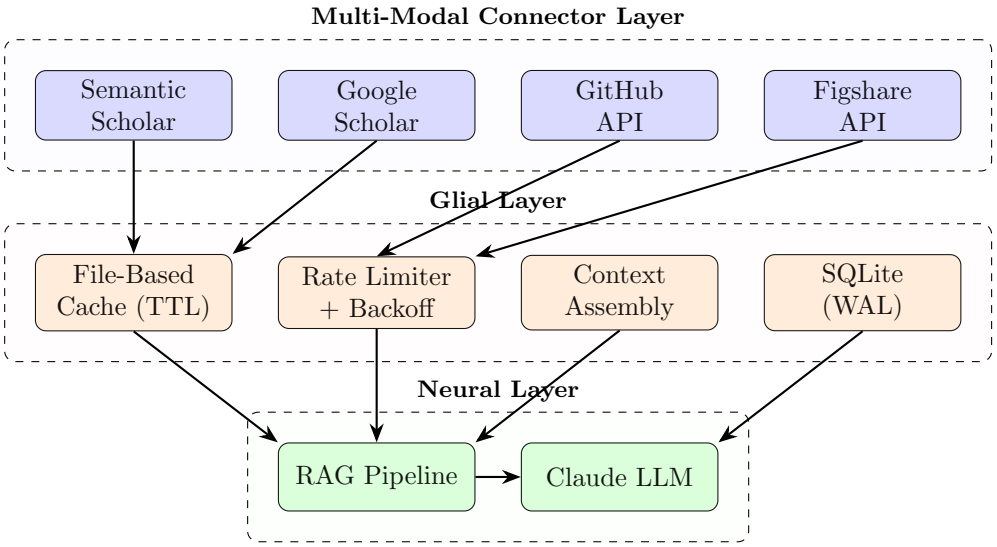

Figure 1: BGNO architecture overview. The Multi-Modal Connector Layer fetches data from four scholarly APIs in parallel. The Glial Layer manages caching, rate limiting, context assembly, and persistent storage. The Neural Layer implements RAG with an LLM to synthesize conversational responses.

3.1 Multi-Modal Connector Layer

The Connector Layer integrates four external data sources, each accessed through a dedicated asynchronous connector:

(i) **Semantic Scholar API**: Retrieves author profiles, publication lists with citation counts, and H-index values via the Semantic Scholar Academic Graph API.

(ii) **Google Scholar**: Accesses author profiles, the i10-index, and publication lists through the `scholarly` Python library with proxy rotation for rate-limit mitigation.

(iii) **GitHub API**: Fetches public repository metadata including star counts, fork counts, language distributions, and license information.

(iv) **Figshare Search API**: Queries datasets and software artifacts associated with a researcher, retrieving download counts, view counts, DOIs, and file metadata.

All four connectors execute in parallel using `asyncio.gather`, and the system gracefully degrades when individual sources are unavailable. The Semantic Scholar and Google Scholar results are merged via a deduplication procedure: paper titles from both sources are normalized (lowercased, punctuation-stripped, whitespace-collapsed), and pairwise similarity is computed using Python's `SequenceMatcher`. Papers with a normalized similarity ratio exceeding 0.85 are considered duplicates; in such cases, the citation count is set to $\max(c_{\text{S2}}, c_{\text{GS}})$ and author-level metrics (H-index, total citations) similarly take the maximum across sources:

$$h = \max(h_{\text{S2}},\, h_{\text{GS}}), \qquad c_{\text{total}} = \max(c_{\text{S2}},\, c_{\text{GS}}). \tag{1}$$

Papers present in Google Scholar but absent from Semantic Scholar (similarity below 0.85 to all S2 titles) are appended to the merged list, providing broader coverage.

3.2 Glial Layer

The Glial Layer provides the metabolic infrastructure that sustains the system's operation:

- **File-based caching**: Each connector's responses are cached as JSON files with configurable time-to-live (TTL). Google Scholar data, which changes slowly and whose access is rate-limited, uses an aggressive 48-hour TTL. Other connectors use a 24-hour default. Cache keys are SHA-256 hashes of the request parameters.
- **Rate limiting with exponential backoff**: External API calls are throttled to respect rate limits. When a rate limit is encountered, the connector retries with exponential backoff, preventing cascading failures.
- **Context preparation**: Fetched data is assembled into structured Markdown documents organized by section (publications, repositories, datasets, QIC scores). This Markdown context serves as the retrieval corpus for the Neural Layer.
- **Persistent storage**: Researcher profiles, registration data, and configuration are stored in SQLite with Write-Ahead Logging (WAL) mode enabled for concurrent read performance. Foreign key constraints ensure referential integrity.

3.3 Neural Layer

The Neural Layer implements Retrieval-Augmented Generation using Anthropic's Claude (model `claude-sonnet-4-5-20250929`) as the generative backbone. The system prompt positions the LLM as a "digital twin" representing the researcher:

> *"You are ResearchTwin, a digital twin representing researcher [Name]. You answer questions about their research, publications, code, datasets, and impact metrics. Use the provided context to give accurate, specific answers. Cite specific papers, repositories, or datasets when relevant."*

The context window is populated with the structured Markdown assembled by the Glial Layer, including publication lists with citation counts, repository descriptions with star/fork metrics, dataset metadata with download statistics, and per-object QIC scores. Responses are bounded to 1024 tokens to ensure conciseness and reduce latency.

This architecture intentionally avoids maintaining a persistent vector store or embedding index. Instead, each query triggers a fresh context assembly from cached connector data, ensuring that the LLM always operates on current information without the staleness risks inherent in pre-computed embeddings.

4 The S-Index: A Quality–Impact–Collaboration Framework

The S-index extends traditional citation-based metrics by quantifying the multi-modal impact of a researcher's complete scholarly output: publications, datasets, and code. The formal specification is maintained at https://github.com/martinfrasch/S-index.

4.1 Per-Object Quality Score (FAIR-Based)

For each research artifact j (dataset or code repository), we compute a Quality score Q_j grounded in the FAIR principles:

Definition 1 (Quality Score)**.** *Let $F_j, A_j, I_{q,j}, R_j \in [0, 10]$ denote the Findability, Accessibility, Interoperability, and Reusability scores of artifact j, respectively. The Quality score is:*

$$Q_j = 0.3\,F_j + 0.3\,A_j + 0.2\,I_{q,j} + 0.2\,R_j. \tag{2}$$

Each FAIR dimension is operationalized through metadata-derived indicators:

Table 1: FAIR-based Quality Score operationalization. Points are additive within each dimension, capped at 10.

Dimension	**Indicator**	**Points**
3*Findability (F)	Has DOI	+6
	Has title	+2
	Has description or categories	+2
3*Accessibility (A)	Has public URL	+5
	Flagged as public	+3
	Files present	+2
3*Interoperability (I_q)	Typed (dataset/software/code)	+4
	Has DOI (standard identifier)	+3
	Has categories	+3
3*Reusability (R)	Has license	+5
	Description $>$ 50 characters	+3
	Has README or multiple files	+2

The weighting in Equation **??** assigns equal importance to Findability and Accessibility (0.3 each), reflecting that artifacts must first be discoverable and accessible before interoperability and reusability become relevant.

4.2 Impact Score

The Impact score captures actual reuse of the artifact, measured through platform-specific engagement metrics:

Definition 2 (Impact Score)**.** *Let r_j denote the number of reuse events for artifact j. The Impact score is:*

$$I_j = 1 + \ln(1 + r_j). \tag{3}$$

The logarithmic transformation ensures diminishing returns at scale, preventing a single viral artifact from dominating the metric. The additive constant 1 guarantees $I_j \geq 1$ for all artifacts, including those with zero observed reuse. Reuse events are source-specific:

- **Figshare datasets**: $r_j = d_j + \lfloor v_j/10 \rfloor$, where d_j is the download count and v_j is the view count. Views are discounted by a factor of 10 relative to downloads, as downloads represent a stronger signal of intent to reuse.
- **GitHub repositories**: $r_j = s_j + 3f_j$, where s_j is the star count and f_j is the fork count. Forks are weighted 3× relative to stars, as forking implies active code reuse rather than passive bookmarking.

4.3 Collaboration Score

The Collaboration score rewards artifacts produced through multi-author, multi-institutional effort:
Definition 3 (Collaboration Score)**.** *Let N_a denote the number of authors and N_i the number of distinct institutions contributing to artifact j. The Collaboration score is:*

$$C_j = \big(1 + \ln(N_a)\big) \times \big(1 + 0.5\ \ln(N_i)\big). \tag{4}$$

The logarithmic scaling of both terms reflects the observation that collaboration value grows sub-linearly with team size. The institutional diversity term is weighted at 0.5 relative to author count, acknowledging that inter-institutional collaboration carries additional coordination costs that signal higher-impact work.

4.4 Per-Object Score

Definition 4 (Per-Object QIC Score)**.** *The score for artifact j is the product of its three components:*

$$s_j = Q_j \times I_j \times C_j. \tag{5}$$

The multiplicative structure ensures that all three dimensions must be non-trivially satisfied: high quality with zero impact, or high impact with poor quality, both yield low scores.

4.5 Paper Impact Term

Publications are assessed through a dedicated term that leverages traditional bibliometric data:
Definition 5 (Paper Impact)**.** *Let h denote the researcher's H-index and c their total citation count, each taken as the maximum across available sources (Equation **??**). The Paper Impact is:*

$$P = h \times \big(1 + \log_{10}(c + 1)\big). \tag{6}$$

The $\log_{10}$ scaling of total citations moderates the influence of a small number of highly-cited papers, while the H-index base rewards consistent productivity.

4.6 Researcher S-Index

Definition 6 (S-Index)**.** *The S-index for researcher i aggregates the Paper Impact term with per-object scores across all datasets and code repositories:*

$$S_i = P + \sum_{j \in \mathcal{D}_i} s_j^{(data)} + \sum_{k \in \mathcal{C}_i} s_k^{(code)}, \tag{7}$$

where $\mathcal{D}_i$ is the set of researcher i's datasets and $\mathcal{C}_i$ is their set of code repositories.
Proposition 1 (Non-negativity and Monotonicity)**.** *The S-index satisfies $S_i \geq 0$ for all researchers i. Furthermore, S_i is monotonically non-decreasing in each of its component metrics: improving any FAIR score, gaining additional reuse events, increasing collaboration breadth, or accumulating citations can only increase S_i.*

Proof. Each component score $Q_j, I_j, C_j \geq 0$ by construction (sums of non-negative terms with $\ln(1 + x) \geq 0$ for $x \geq 0$), hence $s_j \geq 0$. The Paper Impact term $P \geq 0$ since $h \geq 0$ and $\log_{10}(c + 1) \geq 0$. As S_i is a sum of non-negative terms, $S_i \geq 0$. Monotonicity follows from the fact that each sub-expression is non-decreasing in its respective input variables. □

5 Inter-Agentic Discovery Protocol

A central design goal of ResearchTwin is to enable *autonomous AI agents* to discover and traverse research artifacts across researchers without human mediation. We achieve this through a structured API that uses Schema.org types and HATEOAS (Hypermedia as the Engine of Application State) navigation.

5.1 Endpoint Taxonomy

The discovery API exposes five resource types, each annotated with a Schema.org `@type`:

Table 2: Inter-agentic discovery API endpoint taxonomy.

Endpoint	@type	Description
`/api/researcher/{slug}/profile`	`Person`	Researcher identity, ORCID, S-index, and HATEOAS links to sub-resources
`/api/researcher/{slug}/papers`	`ItemList` of `ScholarlyArticle`	Publications with title, year, citation count, and source URL
`/api/researcher/{slug}/datasets`	`ItemList` of `Dataset`	Datasets with DOI, download/view counts, and QIC scores
`/api/researcher/{slug}/repos`	`ItemList` of `SoftwareSourceCode`	Repositories with stars, forks, language, and QIC scores
`/api/discover?q={query}`	`SearchResultSet`	Cross-researcher search with optional type filter

5.2 HATEOAS Navigation

Each `Person` profile response includes a `resources` object containing relative URIs to the researcher's papers, datasets, and repositories. An AI agent can begin at any researcher profile, follow links to enumerate their artifacts, and then use the cross-researcher `/api/discover` endpoint to find related work by other researchers. This self-describing structure allows agents to navigate the knowledge web without prior knowledge of the API schema.

5.3 Cross-Researcher Discovery

The `/api/discover` endpoint accepts a text query q and an optional type filter (`dataset`, `repo`, or `paper`). For each registered researcher, the system searches titles and descriptions of their artifacts, returning results annotated with Schema.org types, QIC scores, and provenance metadata. Results are ranked by QIC score (for datasets and repositories) or citation count (for papers), enabling agents to prioritize high-impact discoveries.

This protocol enables a workflow in which an AI agent, given a research question, autonomously:

1. Queries `/api/discover` to identify relevant artifacts across all registered researchers.
2. Follows HATEOAS links to retrieve full artifact metadata and QIC scores.
3. Synthesizes findings into a research brief, identifying potential collaborations or data reuse opportunities.

6 Federated Architecture

ResearchTwin adopts a three-tier federated architecture that balances data sovereignty with global discoverability:

Tier 1 — Local Nodes. Individual researchers or small labs operate their own ResearchTwin instances via `run_node.py`. A Local Node is a complete, self-contained deployment with its own SQLite database, full API surface, and optional chat functionality. Researchers control their data entirely and can operate offline. Local Nodes may optionally register with a hub for discoverability.

Tier 2 — Hubs. Laboratory or departmental aggregators federate multiple Local Nodes, providing cross-node search and discovery within an institutional context. Hubs maintain an index of registered nodes and proxy discovery queries. This tier is currently specified but not yet deployed.

Tier 3 — Hosted Edges. Cloud-hosted instances (e.g., `https://researchtwin.net`) provide the full platform with advanced analytics, the D3.js knowledge graph visualization, and global cross-researcher discovery. Hosted Edges serve as the entry point for researchers who prefer not to self-host.

This model is inspired by Discord's server federation, where communities operate autonomously within a shared protocol layer. Data sovereignty is preserved at each tier: a Tier 1 node's data remains on the researcher's infrastructure, and registration with higher tiers involves metadata exchange only (researcher name, external API identifiers), not transfer of research artifacts themselves.

7 Implementation

ResearchTwin is implemented in Python 3.12 using FastAPI as the web framework, with SQLite in WAL mode for persistent storage. The system is containerized via Docker Compose for deployment portability. Table **??** summarizes the technology stack.

Table 3: Implementation technology stack.

Component	Technology
Web framework	FastAPI 0.100+, Uvicorn ASGI
Language model	Anthropic Claude (`claude-sonnet-4-5-20250929`)
Database	SQLite 3.x with WAL mode
Caching	JSON file-based, SHA-256 keyed, configurable TTL
Containerization	Docker Compose
Frontend	Vanilla HTML/CSS/JavaScript, D3.js (force-directed graph)
Reverse proxy	Nginx with TLS termination
Bot integration	discord.py with application commands

7.1 Security

The platform implements defense-in-depth security measures:

- **Content Security Policy and headers**: All responses include `X-Content-Type-Options: nosniff`, `X-Frame-Options: DENY`, `X-XSS-Protection`, and strict referrer policies via middleware.

- **CORS**: Origins are restricted to the production domain and localhost for development.
- **Input validation**: All user-facing inputs are validated via Pydantic models with regex constraints (e.g., slug format: `/^[a-z0-9][a-z0-9_-]{0,126}[a-z0-9]$/`).
- **Anti-spam registration**: Self-registration uses a honeypot field (`website`) that legitimate users never see but automated bots fill, combined with email uniqueness enforcement and input length constraints.
- **Rate limiting**: External API calls are rate-limited with exponential backoff; the Google Scholar connector uses aggressive caching (48-hour TTL) to minimize requests to a rate-sensitive source.

7.2 Frontend Visualization

The web frontend renders a force-directed knowledge graph using D3.js, displaying the researcher's artifacts as interconnected nodes. Publications, repositories, and datasets are represented as distinct node types with edges indicating shared authorship, topic similarity, or cross-references. The visualization provides an intuitive overview of a researcher's scholarly footprint and the relationships among their artifacts.

7.3 Discord Integration

A Discord bot built with `discord.py` provides conversational access to ResearchTwin within research group servers. Slash commands enable users to query a researcher's profile, retrieve S-index reports, and interact with the digital twin without leaving their communication platform.

8 Discussion

8.1 Advantages over Static Repositories

ResearchTwin offers several advantages over the status quo of static repositories and disconnected profiles:

1. **Unified multi-modal view**: A researcher's complete scholarly output—papers, code, and data—is accessible through a single conversational interface, eliminating the need to manually integrate information across platforms.
2. **Real-time impact measurement**: The S-index is computed on demand from live API data, providing current impact scores rather than periodic snapshots.
3. **Agent-navigable knowledge web**: The Schema.org-typed API enables AI agents to autonomously discover cross-researcher synergies, a capability absent from static profiles.
4. **Data sovereignty**: The federated architecture allows researchers to control their data while remaining globally discoverable.

8.2 Comparison to Existing Systems

Table **??** compares ResearchTwin to existing scholarly profile and discovery systems.

Table 4: Comparison of ResearchTwin with existing systems.

Feature	ResearchTwin	Google Scholar	ORCID	OpenAIRE	Semantic Scholar
Multi-modal (papers+code+data)	✓		✓	✓	
Conversational access	✓				
Composite impact metric	✓				
Code reuse in metric	✓				
Data reuse in metric	✓				
Agent-navigable API	✓			✓	✓
Federated self-hosting	✓				
Open source	✓			✓	

8.3 Limitations

Several limitations should be acknowledged:

- **No full-text indexing**: The current system operates on metadata and abstracts only. Full-text search over paper content would substantially improve retrieval quality but raises copyright and storage challenges.
- **Limited connector coverage**: The current four connectors (Semantic Scholar, Google Scholar, GitHub, Figshare) do not cover PubMed, arXiv, Zenodo, Dryad, or domain-specific repositories. This limits applicability in fields where these sources are primary.
- **Google Scholar access constraints**: Google Scholar does not provide an official API. The `scholarly` library relies on web scraping, which is subject to rate limiting and blocking. The 48-hour cache TTL mitigates but does not eliminate this fragility.
- **S-index calibration**: The QIC weights (Table **??**) and reuse event weightings (e.g., forks = 3× stars) are based on informed judgment rather than empirical calibration. Large-scale validation studies are needed to refine these parameters.
- **Hub tier not yet deployed**: The Tier 2 Hub layer is specified but not yet implemented, limiting cross-institutional federation to the Hosted Edge model.

8.4 Ethical Considerations

ResearchTwin operates exclusively on *public metadata*—author profiles, publication titles, abstracts, citation counts, repository descriptions, and dataset metadata. No full-text papers are scraped or stored. All data is sourced through official APIs or publicly available metadata endpoints. The system does not attempt to infer private information about researchers. Self-registration includes explicit consent, and researchers retain the ability to de-register and have their profiles removed.

9 Future Work

Several directions for future development are planned:

1. **Additional connectors**: PubMed and arXiv connectors would substantially improve coverage in biomedical and physical sciences. Zenodo and Dryad connectors would broaden dataset coverage.
2. **Hub federation protocol**: Implementation of the Tier 2 Hub layer with a defined federation protocol (potentially based on ActivityPub) for cross-institutional discovery.
3. **Full-text semantic search**: Integration of embedding-based retrieval over paper abstracts and README files, using a persistent vector store alongside the current metadata-only retrieval.
4. **Collaborative filtering**: Researcher recommendation based on artifact similarity, citation overlap, and complementary methodological expertise.
5. **S-index calibration**: Empirical studies correlating S-index scores with expert assessments of research impact, enabling data-driven refinement of QIC weights.
6. **Multi-language support**: Internationalization of the platform interface and support for non-English research metadata.
7. **Institutional dashboards**: Aggregated S-index analytics at the department, laboratory, or institutional level for research assessment and strategic planning.

10 Conclusion

We have presented ResearchTwin, an open-source federated platform that transforms a researcher's publications, datasets, and code into a conversational digital twin. The Bimodal Glial-Neural Optimization architecture cleanly separates data management concerns (caching, rate limiting, context assembly) from generative intelligence (RAG with a large language model), enabling scalable and cost-effective operation. The S-index provides a formally defined, FAIR-grounded composite metric that captures the full spectrum of modern research impact—publications, code reuse, and data sharing—addressing a significant gap in existing bibliometric tools.

The inter-agentic discovery API, built on Schema.org types and HATEOAS navigation, positions ResearchTwin as infrastructure for an emerging paradigm in which AI agents autonomously discover research synergies across institutional boundaries. The three-tier federated architecture ensures that this discoverability does not come at the cost of data sovereignty.

We believe that the transition from static repositories to agentic knowledge webs is not merely a technological upgrade but a fundamental shift in how scientific knowledge is organized, discovered, and reused. ResearchTwin represents a concrete step toward this vision, and we invite the research community to deploy, extend, and critique the platform.

The source code is available at `https://github.com/martinfrasch/ResearchTwin` under the MIT license. The S-index specification is maintained at `https://github.com/martinfrasch/S-index`. A hosted instance is accessible at `https://researchtwin.net`.

Acknowledgments

The author thanks the open-source community and early adopters for their feedback during the development of ResearchTwin. This work was conducted independently and received no external funding.

References

[1] J. E. Hirsch, "An index to quantify an individual's scientific research output," *Proceedings of the National Academy of Sciences*, vol. 102, no. 46, pp. 16569–16572, 2005. https://doi.org/10.1073/pnas.0507655102

[2] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, *et al.*, "The FAIR guiding principles for scientific data management and stewardship," *Scientific Data*, vol. 3, article 160018, 2016. https://doi.org/10.1038/sdata.2016.18

[3] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 9459–9474, 2020. https://arxiv.org/abs/2005.11401

[4] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*, pp. 85–113, Springer, 2017. (Concept originally proposed in 2002; formalized in M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," white paper, 2014.)

[5] Schema.org Community Group, "Schema.org vocabulary," https://schema.org/, accessed February 2026.

[6] Anthropic, "Claude API documentation," https://docs.anthropic.com/, accessed February 2026.

[7] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha, *et al.*, "Construction of the literature graph in Semantic Scholar," in *Proceedings of NAACL-HLT*, pp. 84–91, 2018. https://doi.org/10.18653/v1/N18-3011

[8] P. Manghi, L. Candela, and B. Lossau, "OpenAIRE: European open access infrastructure," *D-Lib Magazine*, vol. 18, no. 11/12, 2012. https://doi.org/10.1045/november2012-manghi

[9] C. Webber, J. Tallon, O. Shepherd, A. Guy, and E. Prodromou, "ActivityPub," W3C Recommendation, 23 January 2018. https://www.w3.org/TR/activitypub/

[10] L. L. Haak, M. Fenner, L. Paglione, E. Pentz, and H. Ratner, "ORCID: A system to uniquely identify researchers," *Learned Publishing*, vol. 25, no. 4, pp. 259–264, 2012. https://doi.org/10.1087/20120404

[11] M. Hahnel, "Referencing: The reuse factor," *Nature*, vol. 520, no. 7547, pp. S2–S3, 2015. https://doi.org/10.1038/520S2a

[12] S. Ramírez, "FastAPI: Modern, fast web framework for building APIs with Python 3.6+," https://fastapi.tiangolo.com/, accessed February 2026.