UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

DÉPARTEMENT DES SCIENCES ADMINISTRATIVES

WORKING PAPER NO. 44

The Post-ChatGPT Computer Science Revolution

A Comprehensive Systematic Mapping, Bibliometric Analysis, and Knowledge Graph of Computer Science Research, 2022–2026

Simon-Pierre Boucher

Département des sciences administratives
Université du Québec en Outaouais

simon-pierre.boucher@uqo.ca

Gatineau – Pavillon Alexandre-Taché
283, boulevard Alexandre-Taché
Gatineau, Québec, Canada J9A 1L8

November 2026

Version 0.1

Abstract

Keywords: systematic mapping, bibliometrics, computer science, large language models, structural change, knowledge graph, emerging topics, OpenAlex

JEL Classification: C55, C88, O33, L86, I23

Contents

I Introduction and Scientific Motivation **10**

II The Evolution of Computer Science **10**

1 The disciplinary structure of computer science **10**

1.1 Classification schemes and what they encode 10
1.2 The forty-category ontology of this study 12
1.3 Where the classifications disagree 13
1.4 Why the growth of a field is hard to measure 15

2 Three prior transformations and how they propagated **16**

2.1 The internet and web era 16
2.2 The deep-learning wave, 2012–2017 17
2.3 The transformer and pretraining era, 2017–2022 19
2.4 A comparative reading of the three waves 22
2.5 The publication system as it stood in 2022 23

3 The non-AI areas at the baseline, 2018–2022 **24**

3.1 Systems, cloud and distributed computing 24
3.2 Hardware and computer architecture 25
3.3 Theory of computation 26
3.4 Programming languages, compilers and formal verification 26
3.5 Security and cryptography 27
3.6 Privacy-preserving computation and federated learning 28
3.7 Data systems, information retrieval and networks 28
3.8 Human–computer interaction and computer graphics 29
3.9 Perception, language, speech and robotics 30
3.10 Quantum computing 30
3.11 Scientific computing, computational biology and sustainable computing 31

4 November 2022 as a reference date **32**

4.1 What ChatGPT was and was not 32
4.2 The releases before and after 32

4.3 Why the date is used descriptively and not causally . . . 34

5 What the literature already claims about the post-2022 period 35
5.1 The growth of AI research . . . 35
5.2 LLM-assisted writing . . . 36
5.3 Illusions of understanding and the epistemology of AI in science . . . 37
5.4 Benchmark saturation and the reproducibility of evaluation . . . 37
5.5 Concentration, disruption and the pace of science . . . 37
5.6 Why the claims need a corpus-level test . . . 38

III Data and Systematic Review Protocol 38

6 Information sources and conditions of access 38
6.1 Why a multi-source design . . . 39
6.2 OpenAlex . . . 39
6.3 DBLP . . . 40
6.4 arXiv . . . 41
6.5 ACL Anthology . . . 41
6.6 Crossref, Semantic Scholar and other sources . . . 42

7 Identification: what each source contributes 42
7.1 Scope filters . . . 43
7.2 The OpenAlex topic classifier as a scope instrument . . . 43
7.3 Verification of the scale against the mandate . . . 44

8 Record linkage across sources 44
8.1 Keys and matching order . . . 45
8.2 The second snapshot pass . . . 46
8.3 Precision of the linkage . . . 46

9 Deduplication of preprint and published versions 47
9.1 Pairing rule . . . 47
9.2 What is kept from each member of a pair . . . 47
9.3 Outcome and residual duplication . . . 48

10 Eligibility criteria and exclusions 48
10.1 Reason codes . . . 48

10.2 Amendments A-001 and A-002 and their evidence . . . 50

11 The computer-science core, the penumbra and the strict core 51
11.1 Counts . . . 52
11.2 The strict core . . . 52
11.3 What the design does not do . . . 52

12 Dates, date precision, indexing lag and right truncation 54
12.1 Which date . . . 54
12.2 Date precision and the January artefact . . . 54
12.3 Indexing lag and why `created_date` cannot measure it . . . 55
12.4 Right truncation . . . 56

13 Quality indicators and known biases 57
13.1 Metadata completeness . . . 57
13.2 Coverage overlap between sources . . . 57
13.3 Known biases and their expected direction . . . 58

14 Research-integrity procedures 60
14.1 Preregistration and the amendments log . . . 60
14.2 Immutable raw snapshots and provenance . . . 61
14.3 Numbers that cannot be typed . . . 61
14.4 Verified references . . . 62
14.5 Agents, artefacts and the register . . . 62
14.6 Human-review checkpoints . . . 62
14.7 What the procedures do not guarantee . . . 63

IV Computational Methodology 63

15 Notation and the corpus as a measurable object 63
15.1 Works, dates and periods . . . 63
15.2 Categories, areas and the measurable structure . . . 65
15.3 Storage and computation . . . 65

16 Record linkage, deduplication and eligibility 66
16.1 Normalised keys and the linkage cascade . . . 66
16.2 Preprint–publication deduplication . . . 68

16.3 Eligibility predicate . . . 68
16.4 Design of the linkage-precision estimate . . . 70

17 Multi-label classification into the ontology **70**
17.1 Signals and the score . . . 71
17.2 Primary category . . . 72
17.3 Validation design and metrics . . . 72
17.4 Known failure modes . . . 73

18 Bibliometric indicators **74**
18.1 Volume, composition and growth . . . 74
18.2 Fixed citation windows . . . 75
18.3 Field normalisation . . . 75
18.4 Inequality and concentration . . . 76
18.5 Collaboration, countries and institutions . . . 77
18.6 Diversity indices . . . 77

19 Citation and collaboration networks **78**
19.1 The citation graph and its boundary . . . 78
19.2 Category flow matrices . . . 79
19.3 Modularity of the yearly flow network . . . 80
19.4 Co-authorship, institution and country networks . . . 80

20 Semantic mapping **81**
20.1 Document embeddings . . . 81
20.2 Atlas: projection and clustering . . . 81
20.3 Stability . . . 82
20.4 Category centroids, drift and novelty . . . 83

21 Structural change in monthly series **83**
21.1 Series construction and truncation . . . 84
21.2 Interrupted time series with seasonality and HAC errors . . . 85
21.3 Placebo distribution . . . 85
21.4 Multiple breaks: dynamic programming with information criteria . . . 86
21.5 PELT . . . 87
21.6 Bayesian online changepoint detection . . . 87
21.7 Alternative event dates . . . 89

22 Emerging-topic detection **89**
22.1 Units and panel . . . 89
22.2 Features at a cutoff . . . 90
22.3 Label at horizon h . . . 91
22.4 Cutoffs, models and metrics . . . 91
22.5 Lead time and the preregistered known-emergence set . . . 92
22.6 Look-ahead safeguards and what they do not cover . . . 92

23 Knowledge-graph extraction **93**
23.1 Structural and extracted layers . . . 93
23.2 High-priority and landmark sets . . . 93
23.3 Schema, provenance and machine verification . . . 93

24 Uncertainty, provenance and reproducibility **95**
24.1 What is uncertain and how it is quantified . . . 95
24.2 Seeds and determinism . . . 95
24.3 Ledgers, hashes and the chain of custody . . . 96
24.4 Human checkpoints and the status of conclusions . . . 96

V The Artificial Intelligence Transformation **96**

VI Systems and Computing Infrastructure **97**

VII Theory, Algorithms, and Security **97**

VIII Scientific Knowledge Networks **97**

IX Empirical Results **97**

X The Next Research Frontiers **97**

XI Discussion and Conclusion **97**

Appendices **141**

A The Ontology of Computer-Science Research **141**
A.1 Structure: areas, leaves and edges . . . 142
A.2 The forty categories . . . 144
A.3 Signal types and the assignment rule . . . 150
A.4 Coverage of the OpenAlex topic vocabulary . . . 153
A.5 Coverage of the DBLP venue vocabulary . . . 154
A.6 Unmapped field-17 topics and the reasons . . . 156
A.7 Known ambiguities and the decisions taken . . . 159
A.8 Limitations of the mapping and the validation design . . . 161

B Search strategies, sources and acquisition record **161**
B.1 Acquisition at a glance . . . 162
B.2 OpenAlex works snapshot . . . 162
B.3 Second snapshot pass by identifier set . . . 165
B.4 OpenAlex API increment . . . 166
B.5 DBLP . . . 167
B.6 arXiv . . . 168
B.7 ACL Anthology . . . 169
B.8 What was sought and not obtained . . . 169
B.9 PRISMA-S search record . . . 170
B.10 Exclusion codes and their predicates . . . 170
B.11 Record-linkage cascade and thresholds . . . 173
B.12 Reproducing the acquisition . . . 174

C Mathematical derivations **175**
C.1 The interrupted time series as a linear regression . . . 175
C.2 Bai–Perron multiple breaks by dynamic programming . . . 177
C.3 PELT: cost, penalty and pruning . . . 179
C.4 Bayesian online changepoint detection . . . 180

D Data Dictionary **182**
D.1 Conventions . . . 182
D.2 Raw-source tables . . . 183
D.3 Normalised OpenAlex tables . . . 186
D.4 Key tables and linkage tables . . . 190

D.5 The corpus table . . . 191
D.6 Citation windows . . . 193
D.7 Citation-graph edge tables . . . 194
D.8 Classification tables . . . 195
D.9 Embedding shards . . . 196

E Reproducibility **197**
E.1 Environment . . . 197
E.2 Steps, commands and measured times . . . 199
E.3 Provenance mechanisms . . . 201
E.4 Incident log and its implications . . . 202
E.5 Protocol amendments . . . 204
E.6 Sources, licences and what is redistributed . . . 205
E.7 Minimal reproduction sequence . . . 206

Part I

Introduction and Scientific Motivation

Part II

The Evolution of Computer Science

1. The disciplinary structure of computer science

Any claim that computer science has been transformed since late 2022 presupposes a map of the discipline on which the transformation can be located. That map is not given by nature. Computer science is organised simultaneously by its objects (processors, programs, data, networks, users), by its methods (proof, construction, measurement, learning), and by the communities and venues through which results circulate; the three organisations overlap only partially, and every bibliometric classification is a compromise among them. This section describes the compromise adopted in the present study and the alternatives against which it must be read. It serves two purposes for the remainder of the manuscript: it fixes the vocabulary of *areas* and *categories* used in Parts V–IX, and it makes explicit the sense in which a "shift of research toward artificial intelligence" is a measurable proposition rather than a slogan.

1.1. Classification schemes and what they encode

Four classification systems are relevant to a corpus of the kind assembled here, and they encode the structure of the discipline in different ways.

The *ACM Computing Classification System* (CCS) is the only one produced by a professional society as an explicit ontology. Its 2012 revision replaced the 1998 four-level scheme with a poly-hierarchy of roughly two thousand concepts organised under about a dozen top-level nodes, among them "Theory of computation", "Computing methodologies" (which houses artificial intelligence, machine learning, computer graphics and simulation side by side), "Computer systems organization", "Software and its engineering", "Security and privacy", "Human-centered computing" and "Networks" (Rous, 2012). The CCS is a statement of how the ACM understood the field in 2012. It is applied by authors at submission time, is used almost exclusively in ACM venues, and is therefore sparse in any corpus that spans IEEE, Springer, Elsevier, open-access publishers and preprint servers. Its value for this study is conceptual rather than operational: it provides an independent, expert-designed reference against which the forty categories of CLAUDE.md, §4 can

be positioned, and it documents a pre-transformer view in which large language models, foundation models, AI agents and AI safety do not exist as concepts at all.

The *arXiv subject classes* are the second system. arXiv was created in 1991 for high-energy physics and its taxonomy reflects that origin; the computer science archive (`cs`) is organised into forty subject classes largely inherited from the ACM scheme of the 1990s (Ginsparg, 2011). Authors choose a primary class and may add cross-lists, so the classification is self-reported, multi-label, and coarse. Its distinctive property is temporal: a preprint is classified at the moment of submission, before peer review, so the arXiv class records how authors positioned their work at the time, not how a later indexer reconstructed it. For the questions of this paper that property matters: when a paper on, say, program verification with language models is submitted to `cs.PL` with a cross-list to `cs.LG`, the pair of classes is a small but genuine piece of evidence about disciplinary boundaries as practitioners perceive them.

The *DBLP computer science bibliography* is the third. DBLP does not classify papers at all; it classifies *venues*. Begun in 1993 as a database and logic programming bibliography and extended to the whole of computer science, it records the tables of contents of conferences and journals with curated author identities (Ley, 2002). Membership in DBLP is itself a judgement that a venue belongs to computer science, and the venue stream to which a paper belongs is a strong signal of community: a paper in the proceedings of OSDI is an operating-systems paper in the sociological sense whatever its subject matter. DBLP therefore encodes the discipline as a set of communities rather than as a set of topics, and the two encodings disagree precisely where this paper expects change, namely where machine-learning methods enter venues that were not built for them.

The fourth system is the family of *algorithmic, citation-based classifications* that underlie modern open bibliographic indexes. OpenAlex, which succeeded the Microsoft Academic Graph after the latter was discontinued at the end of 2021 (Wang et al., 2020; Priem et al., 2022), assigns to every work a ranked list of *topics* from a fixed vocabulary of about 4,500 topics grouped into subfields, fields and domains; the topics were obtained by clustering the citation network and labelling the clusters with a language model, and are assigned to new works by a classifier over title, abstract and venue (OpenAlex, 2025). The approach descends from the publication-level clustering methodology of Waltman and van Eck (2012), refined by the Leiden community-detection algorithm (Traag et al., 2019), and it inherits the known properties of that family: clusters are well connected and reproducible, but their granularity is a free parameter with no natural value (Sjögårde and Ahlgren, 2018), and citation-based taxonomies, while more accurate than text-based or journal-based ones on average (Klavans and Boyack, 2017), lag behind the literature by construction because a cluster must accumulate citations before it can exist. The Semantic Scholar academic graph offers a comparable but differently constructed resource (Kinney et al., 2023a). Older, journal-based classifications of the kind proposed by Glänzel and Schubert (2003) remain in use in evaluative

scientometrics but are too coarse, and too journal-centric, for a conference-dominated discipline.

Each of these systems is a model of the discipline with its own inductive bias. The CCS encodes expert judgement frozen in 2012; arXiv classes encode author self-positioning at submission; DBLP streams encode venue communities; OpenAlex topics encode citation neighbourhoods as of a recent snapshot. None of them is ground truth, and this study treats their disagreements as data rather than as noise.

1.2. The forty-category ontology of this study

The classification used throughout the manuscript is the hierarchical ontology that the project mandate specifies (CLAUDE.md, §4) and that Phase 2 built (Appendix A). It comprises forty leaf categories grouped into eight areas: artificial intelligence and machine learning (nine leaves, from large language models to AI safety); perception, language, robotics and interaction (six); software, programming languages and formal methods (four); theory of computation (two); data, information and distributed systems (six); security, privacy and trust (five); hardware, architecture and emerging computing (five); and computational science and discovery (three). A small number of *specialisation* edges record that one leaf refines another: large language models and multimodal AI refine foundation models; compilers refine programming languages; cloud and edge computing refine distributed computing; federated learning refines privacy-preserving computation; AI accelerators refine computer architecture. The areas are a reporting device; classification operates on leaves.

The assignment rule is deliberately simple so that it can be audited. Let $\mathscr{W}$ be the set of works and $\mathscr{C}$ the set of forty leaves. For each leaf c the ontology lists weighted signals of five types $k \in \{\text{topic, subfield, arXiv, stream, keyword}\}$, with signal set $S_k(c)$ and weights $\omega_{kj} \in (0,1]$ for $j \in S_k(c)$. A work w matches a signal j if the corresponding attribute of w (an OpenAlex topic identifier, a subfield, an arXiv class, a DBLP stream, or a regular expression over title and abstract) contains it. The score of w for c is

$$s(w,c) \;=\; \sum_{k} \min\Big\{1,\; \sum_{j \in S_k(c)} \omega_{kj}\, \mathbf{1}[w \text{ matches } j]\Big\} \;-\; \infty \cdot \mathbf{1}[w \text{ matches an exclusion of } c], \quad (1)$$

so that no single signal type can contribute more than one unit, a weight of one means "this signal alone suffices", and an exclusion pattern vetoes the leaf outright. The work belongs to every leaf with $s(w,c) \geq 1$; its *primary* leaf is $\arg\max_c s(w,c)$, with ties broken in favour of a child over its parent and then by category number. Multi-label membership is not an accident of the rule but its purpose: an instruction-tuning paper is expected to score in both large language models and natural language processing, and the share of works with more than one leaf is itself one of the quantities tracked in Part IX.

Two design choices deserve comment because they shape every downstream result. First,

the ontology is not a relabelling of OpenAlex topics. Six AI categories that the citation-based topics cannot separate (large language models, foundation models, multimodal AI, generative AI, AI agents, AI safety) and seven small categories with no dedicated topic or venue (federated learning, sustainable computing, autonomous scientific discovery, privacy-preserving computation, blockchain, quantum computing, edge computing) rely on keyword patterns of weight one; the build fails if such a pattern is absent. The reason is historical and is the subject of Section 2: the citation clusters that OpenAlex labels were formed by a literature in which "large language model" was not yet a stable term, and a topic such as "Topic Modeling" in fact hosts most BERT-era and retrieval-augmented work. Second, the ontology is positioned relative to the CCS rather than derived from it. The AI area corresponds to the CCS branch *Computing methodologies* → *Artificial intelligence / Machine learning*, but the leaves inside it are drawn from the post-2017 literature; theory, systems, networks, security and human-centred computing map onto the corresponding CCS top-level nodes almost one-to-one; computational biology, scientific computing and sustainable computing sit in CCS under *Applied computing* and *Social and professional topics*, which the present ontology promotes to first-class areas because the questions of this paper concern precisely the traffic between computing and the sciences it serves.

The ontology's limitations are stated in Appendix A and are not repeated here, except for one that bears on interpretation. A category defined partly by keywords is defined partly by vocabulary, and vocabulary changes faster than research practice. If the phrase "large language model" spreads into papers whose methods are unchanged, a keyword-driven category will register growth that is partly nominal. Section 5 returns to this problem under the heading of LLM-assisted writing; Part IV describes the embedding-based checks designed to detect it.

1.3. Where the classifications disagree

The disagreements among the four systems are systematic, and they fall into three kinds.

The first is *coverage*. Comparisons of bibliographic databases consistently show that no source indexes the whole of a discipline and that the gaps are not random: conference proceedings, preprints, and non-English venues are the most unevenly covered (Visser et al., 2021; Martín-Martín et al., 2021). OpenAlex, the backbone of this corpus, inherits the broad but uneven coverage of Microsoft Academic and Crossref; its reference lists are less complete than those of the commercial indexes for part of the period, which affects citation-based measures more than counts (a preprint finding; Culbert et al., 2024. The linking phase of this study illustrates the scale of the problem at the level of record identity. The OpenAlex snapshot contains 15,490,159 works classified in computer science between 2018 and 2026; DBLP contributes 4,347,365 records over the same years, of which 2,736,903 could be matched to an OpenAlex work by DOI or by title and year in the first linking pass and 1,610,462 could not; the arXiv `cs` harvest contributes 915,115 preprints, of which

744,680 were matched, overwhelmingly through the arXiv identifier that OpenAlex stores. These are linkage statistics, not findings about research, but they establish that a classification applied to one source cannot simply be transferred to another: a non-trivial fraction of the venue-defined (DBLP) universe has no citation-defined (OpenAlex) counterpart, and vice versa. Part III describes the second linking pass and the treatment of unmatched records.

The second kind of disagreement is *granularity and boundary*. The arXiv class `cs.LG` is one node among forty; in OpenAlex, machine learning spreads across dozens of topics in several fields, including statistics, engineering and medicine, because the citation network does not respect the boundary of the computer-science field. Conversely, OpenAlex's field 17 ("Computer Science") contains clusters on educational technology, media studies and physics that no computer scientist would claim, and the ontology of this study deliberately leaves most of them unmapped (Appendix A). The practical consequence is that the apparent size of "AI" depends on which boundary is used: a venue-based definition (DBLP streams of NeurIPS, ICML, ICLR, AAAI and the vision and language conferences) gives a different set from a topic-based definition, which gives a different set again from an arXiv-class definition, and the three sets diverge most in the applied sciences, where machine-learning methods are used without the paper belonging to any machine-learning community. The multi-label design of the ontology, and the use of all three signal types with capped weights in equation (1), is the response to this divergence: a work that is an AI paper by every criterion scores high on all of them, while a work that is an AI paper by vocabulary alone is held to a higher evidentiary standard.

The third kind is *temporal*. The CCS was fixed in 2012 and the arXiv classes are older still; OpenAlex topics were generated from a citation network observed in the early 2020s and are reassigned when the snapshot is rebuilt; DBLP streams are stable but new venues appear and old ones are renamed. A classification that does not change cannot register the emergence of a field except as the growth of an existing node; a classification that is regenerated cannot be compared across snapshots without care. Neither property is a defect, but they pull in opposite directions, and a study of structural change must choose. This study freezes all classifications at a single snapshot (Part III), so that any growth it reports is growth of a fixed category, and it treats the emergence of *new* categories as a separate detection problem (Part IV, Section 5).

A final, sociological disagreement should be named explicitly because it affects what counts as a "paper". Computer science is unusual among the sciences in treating selective conferences as the primary venue of record, a practice that has been defended on grounds of speed and criticised on grounds of reviewing quality and the fragmentation of the literature (Vardi, 2009; Fortnow, 2009). The convention has two consequences for bibliometrics. Conference papers are indexed less uniformly than journal articles, so coverage disagreements are concentrated in exactly the venues that the discipline values most; and the conference calendar imposes a strong seasonality

on publication dates, so that monthly series must be deseasonalised before any break can be read off them. The preprint culture adds a third consequence. Computer scientists came to arXiv late relative to physicists and mathematicians, and a large share of computer-science preprints are never matched to a published version, either because they are not published or because the match fails (Lin et al., 2020). A corpus that counts preprints and published versions as separate works will over-count the preprint-heavy subfields; a corpus that links them will under-count whenever linkage fails; and the subfields where linkage fails most often are, again, the fast-moving ones.

1.4. Why the growth of a field is hard to measure

The questions of this paper are ultimately about rates: of publication, of citation, of movement between categories. Rates presuppose denominators, and the denominators of science are themselves contested quantities.

The literature on the growth of science offers two sobering lessons. The first is that measured growth depends heavily on the database. Using cited references as a proxy for the literature that scientists actually use, Bornmann and Mutz (2015) estimated the long-run growth of science at roughly three per cent per year with a doubling time of about a quarter of a century, while estimates based on the publication counts of individual databases vary widely with the database's coverage policy and its own expansion. A later analysis with segmented growth curves across several databases found that the growth rate has itself changed over time, with faster growth in recent decades, but that the location of the segments depends on the source (Bornmann et al., 2021). For the present study the lesson is that a change in the volume of computer-science publications between 2018 and 2026 is the sum of a change in research activity and a change in indexing, and that only the first is of scientific interest. OpenAlex grew substantially during the period as it absorbed new sources and improved its matching; the indexing lag (the delay between a work's publication and its appearance in the snapshot) is longest for the most recent months, so that any series ending in 2026 slopes downward at its end for reasons that have nothing to do with research. These are among the pre-period diagnostics and sensitivity checks required by CLAUDE.md, §6.5 and specified in Part IV.

The second lesson concerns the "science of science" as a field in its own right. The programme described by Fortunato et al. (2018b) and developed at book length by Wang and Barabási (2021) has established robust regularities: citation distributions are heavy-tailed within every field; team size has grown steadily; novelty in the sense of atypical combinations of prior work is associated with higher impact at the top of the distribution; and the productivity of individual scientists is approximately stable over careers. Those regularities are the background against which any field-specific claim must be read. A rise in the citation inequality of AI papers after 2022, for instance, is interesting only if it exceeds the inequality that heavy-tailed citation distributions produce in

every growing field; the hypotheses H1–H8 of the preregistered protocol (Part III) are formulated as comparisons between the post-2022 period and the baseline for that reason.

A further difficulty is specific to computer science and concerns what the discipline takes itself to be. Denning (2005) argued that computer science is a science of information processes and not merely an engineering of artefacts, and Wing (2006) that "computational thinking" is a general intellectual skill rather than a professional one; both arguments were written at a moment when the discipline was defending its boundaries. The post-2022 period raises the converse question: whether the discipline's methods have diffused so widely that the boundary is no longer where the classifications place it. Papers that apply learned models to protein structure, materials, weather or law are classified as computer science by some systems and not others; a transformation that consists partly of such diffusion is partly invisible to a classification that keeps the boundary fixed. The ontology's eighth area, computational science and discovery, and the "penumbra" set defined in Part III (works with a computer-science topic but without venue or arXiv support) exist to make that diffusion observable without inflating the core.

2. Three prior transformations and how they propagated

The post-2022 period is not the first time computer science has been reorganised around a new technology. Three earlier transformations are directly relevant because they established the publication practices, the infrastructure, and the conceptual vocabulary within which the present one unfolds: the internet and web era, which created the systems and the industrial research laboratories; the deep-learning wave of 2012–2017, which created the benchmark culture and the compute economy; and the transformer and pretraining era of 2017–2022, which created the models themselves. For each, the account below is deliberately narrow. It does not attempt a history of computing; it asks what changed in how research was done and published, and how the change propagated from the subfield in which it began to the others. The answers suggest a way of reading the post-2022 data that the empirical parts of this paper then test.

2.1. The internet and web era

The World-Wide Web was presented to the computing community in 1994 as an information system built on three modest conventions, a naming scheme, a transfer protocol and a markup language, that happened to compose (Berners-Lee et al., 1994). Within five years the problem of finding things on the web had produced two results that still structure information retrieval and network science: the use of the hyperlink graph as a source of authority, through PageRank in the Google prototype (Brin and Page, 1998) and through the hubs-and-authorities formulation of Kleinberg (1999). Within another five, the engineering problem of running such systems at scale had produced the Google

File System and MapReduce (Ghemawat et al., 2003; Dean and Ghemawat, 2008), and within a decade the commoditisation of that infrastructure as "cloud computing" had been analysed as a shift in the economics of computing rather than in its technology (Armbrust et al., 2010). Apache Spark, which generalised MapReduce into a unified engine for iterative and interactive workloads, marks the maturation of the era (Zaharia et al., 2016).

Three features of this transformation matter for what followed. First, the decisive results came from industrial laboratories or from academic groups that immediately founded companies, and they were published in the discipline's selective conferences (SOSP, OSDI, WWW, SIGCOMM) rather than in journals; the practice by which an industrial systems paper describes a deployed artefact, with measurements but without a released implementation, dates from this period and persists in the large-model reports discussed below. Second, the web made open-source software the ordinary form of a systems research artefact: Hadoop, the open re-implementation of MapReduce, propagated the method more effectively than the paper did, and the later machine-learning frameworks follow the same pattern. Third, the era created the data. Web-scale text, images, logs and graphs were the precondition for statistical methods in language, vision and recommendation, and the subfields that had access to such data (information retrieval, speech, machine translation) were the first to adopt statistical learning, a decade before the deep-learning wave made it general.

The propagation pattern was *infrastructural*. The web changed the objects of study in networks, databases, distributed systems and security, and it created new subfields (web search, social computing, cloud computing), but it did not change the methods of theory, programming languages, graphics or human-computer interaction except by supplying them with new problems. The discipline's map was extended rather than redrawn: the ACM classification of 2012 could still accommodate every web-era subfield under its existing top-level nodes (Rous, 2012).

2.2. The deep-learning wave, 2012–2017

The second transformation is usually dated to 2012, when a convolutional network trained on two GPUs reduced the top-5 error on the ImageNet Large Scale Visual Recognition Challenge by a margin that no incremental improvement in hand-designed features could match (Krizhevsky et al., 2017; Deng et al., 2009; Russakovsky et al., 2015), and when deep networks replaced Gaussian mixtures in the acoustic models of the major speech laboratories at roughly the same time (Hinton et al., 2012). The methods were not new: the long short-term memory cell dates from 1997 (Hochreiter and Schmidhuber, 1997), convolutional networks from the 1980s, and the retrospective of LeCun et al. (2015) presents the decade's results as the delayed payoff of a thirty-year programme. What was new was the conjunction of labelled data at the scale of ImageNet, general-purpose graphics processors, and a benchmark with a public leaderboard.

The subsequent five years produced the architectural vocabulary that the transformer era would

inherit. Distributed word representations trained by simple predictive objectives (Mikolov et al., 2013) made it natural to treat language as a sequence of dense vectors; the variational autoencoder (Kingma and Welling, 2013) and the generative adversarial network (Goodfellow et al., 2014) established deep generative modelling as a research programme; sequence-to-sequence learning with recurrent encoders and decoders (Sutskever et al., 2014) and, crucially, the attention mechanism that let a decoder consult every encoder state (Bahdanau et al., 2014) reframed machine translation and then every conditional generation task; residual connections (He et al., 2016) removed the practical ceiling on depth. In reinforcement learning, deep Q-networks learned Atari games from pixels (Mnih et al., 2015), and the AlphaGo lineage combined deep networks with tree search to defeat professional Go players, first with human games as training data and then without them (Silver et al., 2016, 2017), generalising to chess and shogi with a single algorithm (Silver et al., 2018) and, in a different formulation, to real-time strategy games (Vinyals et al., 2019).

The changes in publication practice were as consequential as the methods. Four deserve to be stated because the post-2022 data will be read against them.

The first is the *benchmark as the unit of progress*. ImageNet established the pattern in which a public dataset, a fixed metric and a leaderboard define a subfield's agenda; the pattern was reproduced in language understanding, question answering and reinforcement learning, and it made progress legible to outsiders at the cost of concentrating effort on whatever the benchmark measured. The critiques of that culture, which argue that benchmarks are chosen for convenience rather than validity and that leaderboard competition rewards narrow optimisation (Dehghani et al., 2021; Raji et al., 2021), were written in the transformer era but describe a practice born in this one.

The second is the *preprint-first, conference-second* publication cycle. Machine-learning results began to appear on arXiv months before, or instead of, their conference versions; the practice spread from a minority to a norm within the decade, and it is the reason the arXiv `cs` harvest is indispensable to this corpus. The accompanying costs were catalogued at the end of the period: speculation presented as explanation, mathematics as decoration, suggestive language and the conflation of explanation with description (Lipton and Steinhardt, 2019). Concerns about reproducibility were raised at the same time, both in audits of what published papers actually report (Gundersen and Kjensmo, 2018) and in the organised reproducibility programmes that the major conferences introduced in response (Pineau et al., 2020).

The third is the *open-source framework as a shared instrument*. TensorFlow and PyTorch (Abadi et al., 2016b; Paszke et al., 2019), released by Google and Facebook respectively, lowered the cost of entry to deep learning to the point where a method published on arXiv could be reproduced and extended within days; they also standardised the computational substrate on NVIDIA GPUs and, later, on tensor processing units (Jouppi et al., 2017), which made compute a measurable input to research. The retrospective analysis of Sevilla et al. (2022) identifies this period as the start of a

"deep learning era" in which the training compute of notable systems grew with a doubling time of about six months, far faster than before 2010; Thompson et al. (2020) argued that such growth is economically and environmentally unsustainable, a theme to which the sustainable-computing category of this study returns.

The fourth is the *entry of industry as a producer rather than a consumer of research*. The laboratories of Google, DeepMind, Facebook, Microsoft and Baidu published the decisive results of the period in the academic venues and on arXiv, and their access to data and compute began to separate them from universities. Ahmed and Wahed (2020) documented the growing presence of large technology firms and elite universities at the major AI conferences and the relative decline of other institutions; the later and broader analysis of Ahmed et al. (2023) found that industry produced an increasing share of the largest models, the most-cited papers and the leading benchmarks. Whether that concentration intensified after 2022 is hypothesis H6 of the protocol.

The propagation pattern of this wave was *methodological* and sequential. Deep learning entered computer vision and speech first, natural language processing second (roughly 2014–2016, with the sequence-to-sequence results), reinforcement learning and robotics third, and then, more slowly and with more resistance, the subfields whose objects are not perceptual: program analysis (Allamanis et al., 2018), systems (learned index structures and learned scheduling), databases, and scientific computing (Raissi et al., 2019). In each case the receiving subfield first treated learning as a tool for a pre-existing problem, and only later, if at all, reorganised its problems around it. That two-stage pattern, tool adoption followed by problem redefinition, is the template against which Part IX reads the post-2022 movement of methods across the forty categories.

2.3. The transformer and pretraining era, 2017–2022

The third transformation began with an architecture and ended with a change in what a model is for. The Transformer of Vaswani et al. (2017) replaced recurrence with self-attention in the encoder–decoder framework of neural machine translation; its immediate merit was parallelism in training, and its lasting one was that the same block could be stacked to any depth and applied to any sequence of tokens. Within eighteen months it had been combined with the idea, already present in ELMo's contextual word vectors (Peters et al., 2018), that a language model trained on unlabelled text could be *pretrained* once and *fine-tuned* on many tasks: BERT (Devlin et al., 2019), with its masked-language objective, set new results on every task of the GLUE benchmark (Wang et al., 2018) and on SQuAD (Rajpurkar et al., 2016), and its successors showed that most of its design choices were less important than the amount of data and training (Liu et al., 2019; Raffel et al., 2019). The generative branch of the family, the GPT series of decoder-only models, reached its third member in 2020 with 175 billion parameters and the demonstration that a sufficiently large language model could perform new tasks from a few examples placed in its prompt, without any

gradient update (Brown et al., 2020). In-context learning, more than any architectural novelty, is what separates the transformer era from the deep-learning wave: the unit of research shifted from the task-specific model to the general model and the prompt.

The era had a quantitative theory of its own growth. Kaplan et al. (2020) reported that the test loss of autoregressive language models falls as a power law in parameters, data and compute over several orders of magnitude, with the implication that larger models are more sample-efficient; Hoffmann et al. (2022) corrected the allocation, showing that models of the GPT-3 generation were undertrained relative to their size and that compute-optimal training requires data and parameters to scale together. Between the two papers, laboratories built models of increasing size on the strength of the first law (Gopher, PaLM, OPT and the open multilingual BLOOM (Rae et al., 2021; Chowdhery et al., 2022; Zhang et al., 2022; BigScience Workshop, 2022)), and the systems research needed to train them, model parallelism, memory-partitioned optimisers and IO-aware attention kernels (Shoeybi et al., 2019; Rajbhandari et al., 2019; Dao et al., 2022), became a research area in its own right, as did sparse mixtures of experts that decouple parameter count from per-token compute (Fedus et al., 2021). The claim that certain abilities appear discontinuously with scale (Wei et al., 2022b) became one of the era's most cited and most contested propositions; the companion observation that prompting a model to write out intermediate reasoning improves its answers on arithmetic and symbolic tasks (Wei et al., 2022a) established prompting as a method. New benchmarks were built at the scale of the models: a fifty-seven-subject examination of knowledge (Hendrycks et al., 2020), a harder successor to GLUE (Wang et al., 2019), and a collaborative suite of more than two hundred tasks explicitly designed to resist saturation (Srivastava et al., 2022).

The transformer did not remain a language architecture. Vision transformers showed that image patches could be treated as tokens and that, with enough data, the inductive bias of convolution was dispensable (Dosovitskiy et al., 2020); self-supervised objectives, contrastive (Chen et al., 2020), self-distilled (Caron et al., 2021) and masked (He et al., 2022), removed the dependence on labels. Contrastive language–image pretraining aligned the two modalities in a shared embedding space and produced zero-shot classifiers from natural-language prompts (Radford et al., 2021); in the same months, autoregressive text-to-image generation (Ramesh et al., 2021) and then diffusion models, a decade-old idea (Sohl-Dickstein et al., 2015) made practical by denoising objectives (Ho et al., 2020), score-based formulations (Song et al., 2020) and architectural refinements that overtook adversarial models (Dhariwal and Nichol, 2021), converged on text-conditioned image synthesis at photographic quality (Ramesh et al., 2022; Saharia et al., 2022). The latent-diffusion formulation (Rombach et al., 2022), released with open weights in the summer of 2022 under the name Stable Diffusion, is the generative-AI counterpart of ChatGPT: a system whose public availability, more than its technical novelty, changed who could use it. Speech followed the self-supervised path (Baevski et al., 2020); few-shot visual language models (Alayrac et al., 2022) and a single network

trained across text, images, games and robot control (Reed et al., 2022) prefigured the multimodal and agentic systems of the following period. In structural biology, a transformer-based system solved, for most single-chain proteins, a problem that had resisted fifty years of effort (Jumper et al., 2021), and was quickly joined by an independent three-track network (Baek et al., 2021).

Two further lines of work prepared the specific object that appeared in November 2022. Models trained on public code could synthesise functions from docstrings (Chen et al., 2021), which turned software engineering into an early adopter of large models and created the first commercial product built on one. And the problem of making a language model do what a user asks, rather than what the pretraining distribution predicts, was addressed by reinforcement learning from human preferences, a method developed for control tasks (Christiano et al., 2017), applied to summarisation (Stiennon et al., 2020), and then used to fine-tune GPT-3 into InstructGPT (Ouyang et al., 2022), whose 1.3-billion-parameter version was preferred by raters to the 175-billion-parameter base model. The same method, framed around helpfulness and harmlessness, was being developed in parallel elsewhere (Bai et al., 2022). Retrieval-augmented generation, which conditions a generator on documents retrieved from an external index, was proposed in the same period (Lewis et al., 2020) and would become the standard answer to the models' factual unreliability.

The era also produced its own critique. The report that coined the term "foundation model" argued that the homogenisation of research around a few pretrained models was a change in the structure of the field, with risks that were social and institutional as much as technical (Bommasani et al., 2021); the "stochastic parrots" paper questioned whether ever-larger language models were a sound research direction at all, on environmental, documentary and epistemic grounds (Bender et al., 2021). The environmental argument had been quantified for NLP models by Strubell et al. (2019), contested and refined for the largest models by Patterson et al. (2021), and turned into a research agenda by Schwartz et al. (2020); the carbon footprint of a 176-billion-parameter model was measured end to end by Luccioni et al. (2022).

What changed in publication practice during this era can be summarised in three observations. The leading models were described in *technical reports* rather than in peer-reviewed papers, often with hundreds of authors, sometimes without the training data, the code, or the weights; the preprint became the version of record for the most consequential work in the discipline. The infrastructure for sharing models became an institution in its own right: the Transformers library (Wolf et al., 2020) and the model hub built around it made the pretrained model, not the paper, the thing that propagated. And the compute required to participate at the frontier grew beyond the reach of universities, so that the concentration documented by Ahmed et al. (2023) is, in this era, a concentration of capability rather than merely of attention.

2.4. A comparative reading of the three waves

Table 1 summarises the three transformations along the dimensions that matter for the empirical design of this study. The comparison is qualitative; the corpus-level counterpart is the subject of Part IX.

Table 1. Three prior transformations of computer science compared on the dimensions relevant to the post-2022 analysis. Qualitative characterisation by the authors from the works cited in Sections 2.1–2.3; no corpus quantities.

	Internet and web (c. 1994–2010)	Deep learning (2012–2017)	Transformer and pretraining (2017–2022)
Originating subfields	Networks, information retrieval, distributed systems	Computer vision, speech	Natural language processing
Core artefact	Protocols, search engines, warehouse-scale systems	Trained task-specific networks; frameworks	Pretrained general models; prompts; weights
Locus of production	Industry and start-ups; academic systems groups	Academic groups with industry laboratories entering	Industry laboratories, with academic participation declining at the frontier
Publication practice	Selective conferences; deployed-artefact papers; open-source re-implementations	arXiv-first; leaderboards; shared frameworks; reproducibility programmes	Technical reports; model hubs; very large author lists; preprint as version of record
Propagation pattern	Infrastructural: new objects and subfields, methods of other areas unchanged	Methodological and sequential: tool adoption then problem redefinition, perceptual fields first	Substitutive: one architecture and one pretraining recipe across modalities; general models replace task-specific ones
Visibility in classifications	Accommodated by existing top-level nodes	New venues and arXiv classes gain weight; topics split	Vocabulary outruns citation clusters; keyword-defined categories required

Three conclusions follow from the comparison and shape the hypotheses tested later. First, the mechanisms of propagation differ in kind. The web changed the problems of other subfields; deep learning changed their methods; the transformer era changed the *models* they use, which can be adopted without changing either problems or methods. A subfield can therefore register the third transformation at low cost, by using a pretrained model as a component, and the resulting growth in the vocabulary of the subfield need not indicate a reorganisation of its research. Distinguishing component adoption from reorganisation is the purpose of the citation-flow and embedding analyses of Parts IV and VIII.

Second, each wave shortened the publication cycle and shifted the version of record toward the preprint. The consequence for this study is that the arXiv harvest and the OpenAlex index measure different things: the former records when a result was first made public, the latter when it entered the citation system. A break detected in one series and not in the other is informative about the mechanism, not merely about measurement.

Third, the three waves were cumulative in infrastructure and in institutions. The industrial laboratories that produced the web systems trained the generation that produced deep learning, and the frameworks and compute of the deep-learning wave made the transformer era possible. The post-2022 period inherited all of it: the laboratories, the frameworks, the model hubs, the benchmark culture, and the preprint norm. Whatever is new about the period must be identified against that inheritance rather than attributed to a single product release, and the baseline window of 2018–2022 was chosen so that the inheritance is observed directly rather than assumed.

2.5. The publication system as it stood in 2022

The three waves left computer science, on the eve of the reference date, with a publication system unlike that of any other science, and the system's properties determine what the corpus can and cannot see. Four of them are load-bearing for the analyses that follow.

The first is the dominance of the preprint. By 2022 the arXiv was the first and often the only public version of most machine-learning results, and a growing share of systems, theory and security results as well; the matching study of Lin et al. (2020) found that a majority of computer-science preprints in their sample could not be matched to a published version, and that the matched ones had been published with a median delay of many months. The consequence is that the arXiv series lead the indexed series, and that a corpus which dates works by their indexed publication date will place a change later than it occurred and will smear it over the publication lag. Part III dates each work by the earliest of its preprint and publication dates when both are known, and Part IX reports the breaks on both clocks.

The second is the conference calendar. The major venues of every area have fixed annual deadlines, and preprint postings cluster in the weeks before and after them; monthly series of computer-science output are therefore strongly seasonal, with a pattern that differs by area because the calendars differ. Seasonality is not a nuisance to be removed once and forgotten; a change in the *amplitude* of seasonality, for instance because a venue grew or because preprints began to be posted without regard to deadlines, is itself a structural change, and the seasonal component $\mathbf{s}_c(t)$ of equation (2) is modelled separately for the baseline and the research period for that reason.

The third is the state of peer review. The volume of submissions to the leading machine-learning conferences had grown by an order of magnitude in a decade; reviewer pools had not. The debate about whether conferences remained an adequate venue of record (Vardi, 2009; Fortnow, 2009) was,

by 2022, less about prestige than about capacity, and the reproducibility programmes of Pineau et al. (2020) were one institutional response. The relevance to this study is twofold. Reviewing capacity constrains the number of works that can appear in venue-defined categories, so that growth under capacity constraints shows up as preprint growth and as the founding of new venues rather than as growth in existing streams; and the reviews themselves became, after 2022, one of the first places where the use of language models was measured (Liang et al., 2024b).

The fourth is the asymmetry between what is published and what is released. The model reports of Section 2.3 describe systems whose weights, data and code are variably available; the artefact-evaluation and badging practices adopted by systems and software-engineering venues during the period had no counterpart at the frontier of AI, where the most consequential systems were the least documented. The reproducibility indicators of RQ9, which the corpus measures through linked code repositories and data availability statements, are therefore expected to differ sharply by category, and a decline in the AI categories would be consistent with the literature rather than anomalous.

3. The non-AI areas at the baseline, 2018–2022

The hypotheses of this study concern not only the growth of AI but the condition of everything else. H2 predicts that at least one non-AI area declines in relative share while growing in absolute terms; H3 predicts that formal verification, computational complexity, operating systems and computer architecture show *no* structural break in the window around November 2022; H4 predicts that citations crossing from those areas into AI increase. Such predictions can be evaluated only if the state of each area at the baseline is described independently of the AI narrative. This section provides that description for the thirty-one non-AI leaves of the ontology, grouped by area, in a deliberately compressed form: for each, the problems that defined the 2018–2022 literature, the landmark results that the corpus's citation data should recover, and the points of contact with machine learning that existed before 2022. Where a result is known to the authors only from a report rather than from a peer-reviewed paper, that is stated.

3.1. Systems, cloud and distributed computing

The systems literature of the baseline period was organised around the consequences of the warehouse-scale computer (Barroso et al., 2018). Three agendas dominated. The first was the *serverless* abstraction, in which the unit of deployment is a stateless function and the provider manages everything else; the Berkeley view presented it as the next step in the simplification of cloud programming (Jonas et al., 2019), while a contemporaneous critique from the database community argued that its restrictions on state, communication and hardware access made it a

step backward for data-intensive work (Hellerstein et al., 2018). The disagreement was productive: much of the 2019–2022 systems literature consists of attempts to give serverless platforms the state and the locality they lacked. The second agenda was *resource disaggregation*, the separation of compute, memory and storage into independently scaled pools connected by fast networks, an idea with roots in blade-server memory expansion (Lim et al., 2009) that became a central topic of operating-systems research once network latency fell within an order of magnitude of memory latency. The third was *tail latency*: in a service composed of thousands of components, rare slow responses in any of them dominate the user-visible distribution, and the techniques of hedged requests, replication and tied requests described by Dean and Barroso (2013) became the standard vocabulary of large-scale service design.

Operating systems research in the period was marked by the consequences of the Spectre and Meltdown disclosures, discussed under security below, which forced kernel designers to reconsider the cost of isolation, and by the growth of in-kernel programmability. Distributed computing as a theory remained focused on consensus, fault tolerance and the coordination of replicated state, with blockchain systems supplying new adversarial models (Section 3.5). The contact with machine learning before 2022 was of two kinds: "systems for ML", the training and serving infrastructure of Section 2.3, which systems venues absorbed readily, and "ML for systems", the use of learned models for scheduling, configuration and resource management, which was an active but minority topic. The ontology treats cloud and edge computing as children of distributed computing precisely because the venue structure does: OSDI, SOSP, EuroSys and NSDI publish all three.

3.2. Hardware and computer architecture

The baseline period in computer architecture was dominated by the end of two scaling regimes. Dennard scaling, the observation that transistor power density stays constant as transistors shrink (Dennard et al., 1974), had ended in the mid-2000s; its consequence, that a growing fraction of a chip must be left dark at any time, was quantified by Esmaeilzadeh et al. (2011) and set the agenda for the following decade. The slowing of Moore's law completed the picture. The Turing lecture of Hennessy and Patterson (2019) drew the conclusion that general-purpose performance gains were over and that the future lay in domain-specific architectures, open instruction sets and agile hardware design; Leiserson et al. (2020) argued in parallel that performance would henceforth come from software, algorithms and hardware specialisation rather than from the "bottom" of the stack. The tensor processing unit (Jouppi et al., 2017) is the canonical example of the domain-specific architecture, and its publication in a mainstream architecture venue is a measure of how completely machine learning had become the architecture community's defining workload by 2017. AI accelerators are a child of computer architecture in the ontology for that reason, and the hypothesis that the architecture category shows no post-2022 break is, in effect, the hypothesis that

this absorption was complete before ChatGPT.

Compilers underwent a parallel reorganisation around machine-learning workloads. The multi-level intermediate representation MLIR (Lattner et al., 2021), designed to let one compiler infrastructure serve domain-specific languages, tensor programs and hardware targets alike, and the end-to-end tensor compiler TVM (Chen et al., 2018) represent the state of the art at the baseline. Both treat the deep-learning graph as a first-class compilation target; neither depends on large language models.

3.3. Theory of computation

Two developments characterise theoretical computer science at the baseline, one internal and one at the boundary with machine learning.

The internal development is *fine-grained complexity*: the programme of explaining why polynomial-time problems such as edit distance, all-pairs shortest paths and orthogonal vectors have resisted improvement for decades, by proving that faster algorithms for one would contradict a small number of hardness conjectures, above all the strong exponential-time hypothesis (Vassilevska Williams, 2015; WILLIAMS, 2019). The programme reorganised algorithms research around conditional lower bounds in the same way that NP-completeness had organised it around unconditional intractability, and its vocabulary is a reliable marker of the complexity category in this corpus.

The boundary development is the *theory of deep learning*. The empirical observation that networks with enough parameters to memorise random labels nevertheless generalise (Zhang et al., 2021) undermined the classical uniform-convergence account of generalisation and prompted new theory: the double-descent curve, which reconciles the classical bias–variance trade-off with the behaviour of overparameterised models (Belkin et al., 2019), and the neural tangent kernel, which characterises the training dynamics of infinitely wide networks (Jacot et al., 2018). The machine-learning-theory leaf of the ontology houses this literature, which belongs sociologically to the learning conferences and intellectually to the theory community. Its presence is a reminder that the theory area was in contact with AI long before 2022, through learning theory, optimisation and the complexity of sampling and inference.

3.4. Programming languages, compilers and formal verification

The programming-languages literature of the baseline is marked by the success of a language. Rust, whose ownership type system gives memory safety without garbage collection (Matsakis and Klock, 2014), became the first systems language to be adopted at scale on the strength of a safety guarantee; the RustBelt project supplied the semantic foundation by proving, in a separation-logic framework

mechanised in Coq, that the safe fragment is sound and that the unsafe library code on which it rests respects the invariants (Jung et al., 2018). The result is characteristic of the period's formal-methods culture: a verification effort that follows an engineering artefact rather than preceding it, and that uses a proof assistant as its medium.

Proof assistants themselves reached a new stage. The verified C compiler CompCert (Leroy, 2009) and the verified microkernel seL4 (Klein et al., 2009) had shown a decade earlier that production software could be proved correct; the Kepler conjecture was formally proved in 2017 after a twelve-year effort (HALES et al., 2017); and the Lean theorem prover (de Moura et al., 2015), re-implemented as a programming language in its fourth version (Moura and Ullrich, 2021), acquired a community-maintained mathematical library whose scale and organisation were new in formalised mathematics (mathlib Community, 2020). The relevance to the post-2022 period is direct: Lean and its library became the environment in which language models were later asked to prove theorems, so that formal verification, one of the categories predicted *not* to break in H3, is also one where the contact with AI is most visible in the recent literature. The hypothesis is therefore a sharp one.

Software engineering had already begun to absorb learned models before 2022. The survey of Allamanis et al. (2018) organised a decade of "big code" research around the naturalness hypothesis, that source code is as statistically regular as natural language; the CodeXGLUE benchmark collection (Lu et al., 2021) standardised evaluation; and the Codex models (Chen et al., 2021) made program synthesis from natural language a product. The software-engineering category is thus expected to show an early and strong AI signal, and its baseline is not an AI-free one.

3.5. Security and cryptography

Three events organise the security and cryptography literature of the baseline. The first is the disclosure, in January 2018, of Spectre and Meltdown, which showed that speculative execution in essentially all high-performance processors leaks data across isolation boundaries (Kocher et al., 2019; Lipp et al., 2020). The disclosures created a sustained literature on microarchitectural side channels and their mitigations that cuts across security, architecture and operating systems, and they illustrate the kind of cross-area citation flow that H4 measures.

The second is the standardisation of *post-quantum cryptography*. Shor's algorithm (Shor, 1997) had made the eventual obsolescence of RSA and elliptic-curve schemes a matter of engineering timelines, and the case for migrating before large quantum computers exist was made to a general scientific audience by Bernstein and Lange (2017). The process run by the United States National Institute of Standards and Technology from 2016 selected lattice-based and hash-based schemes, published as federal standards in August 2024 (National Institute of Standards and Technology , US,U,U). The applied-cryptography category of this corpus therefore contains a large and growing

body of work whose dynamics are driven by a standards calendar entirely independent of language models, which makes it a useful placebo field. Fully homomorphic encryption, shown to be possible by Gentry (2009), remained the subject of a steady literature on efficiency.

The third is the maturation of *adversarial machine learning*. The observation that imperceptible perturbations can change a network's output (Szegedy et al., 2013) and the attacks and evaluation standards that followed (Carlini and Wagner, 2017) created, by 2018, a field with ten years of history and a recognised research agenda (Biggio and Roli, 2018). The ontology scores adversarial work in both cybersecurity and AI safety, which reflects the baseline state: the problem belongs to both communities and neither owns it. Blockchain systems, finally, entered the period with a systematised research agenda (Bonneau et al., 2015) and left it as a mature if contested area whose publication dynamics follow cryptocurrency markets more than academic calendars.

3.6. Privacy-preserving computation and federated learning

Differential privacy, defined by Dwork et al. (2006) and systematised in the monograph of Dwork and Roth (2014), was by 2018 the accepted formal standard for statistical disclosure control, and its combination with stochastic gradient descent (Abadi et al., 2016a) had made private training of deep networks practical at modest cost in accuracy. Federated learning, proposed as a way to train models on data that never leaves the devices that generated it (McMahan et al., 2016), grew in five years into a field with its own open problems in optimisation, systems, privacy and incentives, catalogued in a survey of unusual breadth (Kairouz and McMahan, 2021). Both categories are keyword-driven in the ontology because neither has a venue of its own; both were already AI-adjacent at the baseline, so that any post-2022 growth they show cannot be read as a new contact with AI.

3.7. Data systems, information retrieval and networks

Database research at the baseline was absorbing the proposal that learned models could replace the data structures at the core of the system: the learned index (Kraska et al., 2018) argued that a model of the cumulative distribution of keys can outperform a B-tree, and the argument was extended to query optimisation, cardinality estimation and tuning. In parallel, approximate nearest-neighbour search over dense vectors, long a niche concern, acquired the data structures and the GPU implementations that would later make vector databases an industry: hierarchical navigable small-world graphs (Malkov and Yashunin, 2020) and billion-scale similarity search on GPUs (Johnson et al., 2021). Information retrieval made the corresponding move from lexical to dense retrieval, in which queries and passages are embedded by a pretrained encoder and matched by inner product (Karpukhin et al., 2020); the retrieval-augmented generation of Lewis et al. (2020) is the junction of that line with language modelling. Of all the non-AI areas, information retrieval is

the one whose baseline most closely anticipates the post-2022 period, and its category should be read accordingly.

Computer networks entered the period with programmability as its organising idea. OpenFlow had separated the control plane from the forwarding plane (McKeown et al., 2008); the P4 language made the forwarding plane itself programmable (Bosshart et al., 2014); and the QUIC transport, designed and deployed at internet scale by a single company before standardisation (Langley et al., 2017), exemplified a mode of networking research in which deployment precedes publication. Edge computing, defined as the placement of computation near the sources of data (Shi et al., 2016), grew with the internet of things and with the latency requirements of on-device inference; the ontology treats it as a child of distributed computing and shares its IoT topics with networks.

3.8. Human–computer interaction and computer graphics

Human–computer interaction had, by 2019, a research programme on interaction with AI systems that predates the conversational interfaces of 2022. Guidelines for human–AI interaction were derived from two decades of literature and validated against deployed products (Amershi et al., 2019); the particular difficulty of designing for systems whose capabilities are uncertain and whose outputs are open-ended was analysed by Yang et al. (2020); and the programme of human-centred AI, which places reliability, safety and trust above autonomy, was articulated by Shneiderman (2020). The HCI category is therefore one where a post-2022 surge, if observed, would be the intensification of an existing agenda rather than the arrival of a new one. Its venues (CHI, UIST, CSCW) are also among the most clearly delimited in DBLP, which makes the category's venue signal strong.

Computer graphics underwent a transformation of its own in 2020, when neural radiance fields showed that a multilayer perceptron could represent a scene well enough to synthesise photorealistic novel views from a handful of photographs (Mildenhall et al., 2020). Neural rendering had been surveyed as an emerging area in the same year (Tewari et al., 2020); within three years, explicit Gaussian primitives optimised by gradient descent had matched the quality of radiance fields at real-time frame rates (Kerbl et al., 2023). The sequence is instructive for the detection methods of Part IV: neural radiance fields are one of the pre-registered "known emergence" topics against which the emerging-topic detector is validated (H8), and they emerged in a non-AI category by the adoption of an AI method, which is exactly the pattern the detector must distinguish from vocabulary drift.

3.9. Perception, language, speech and robotics

Four leaves of the perception area lie outside the AI area in the ontology while being the historical home of deep learning, and their baseline is therefore the baseline of the deep-learning wave itself. Computer vision had, by 2022, completed the transition from convolutional to attention-based architectures and from supervised to self-supervised pretraining described in Section 2.3; its venues (CVPR, ICCV, ECCV) were among the largest in the discipline, and its vocabulary was already that of foundation models. Natural language processing is the subfield in which the transformer era began, and its baseline is paradoxical: the methods that defined the field in 2018, task-specific architectures, feature engineering, and the pipeline of tagging, parsing and semantic analysis, had been displaced by pretrained models before 2022, so that the post-2022 question for NLP is whether the field continued to exist as a distinct community once its central method became everyone's method. The ontology keeps NLP as a task-framed leaf distinct from large language models for that reason, and the ratio of the two is one of the quantities Part IX reports.

Speech and audio computing had followed the same trajectory a few years behind: hybrid acoustic models (Hinton et al., 2012) gave way to end-to-end sequence models and then to self-supervised pretraining on unlabelled audio (Baevski et al., 2020), and the weakly supervised Whisper models trained on hundreds of thousands of hours of transcribed speech (Radford et al., 2022) arrived in the same months as ChatGPT. Robotics is the slowest of the four, because physical data are expensive and simulation transfers imperfectly. End-to-end learned visuomotor policies had been demonstrated in 2016 (Levine et al., 2015), and reinforcement learning in simulation was an established research line, but most of the field's work in the baseline period remained in planning, control and perception for specific platforms. The vision-language-action models of 2023 (Brohan et al., 2023), which transfer web-scale pretraining to manipulation, are the point at which the foundation-model recipe reached robotics, and they fall inside the research period; robotics is therefore one of the categories where a late break, rather than one near the reference date, would be the expected signature of the transformation.

3.10. Quantum computing

Quantum computing at the baseline is defined by a term coined to describe it. Preskill (2018) named the "noisy intermediate-scale quantum" era, in which devices of fifty to a few hundred qubits without error correction might perform useful tasks, and warned that no such task had yet been demonstrated. The claim of quantum computational advantage on a sampling problem (Arute et al., 2019) and the review of algorithms designed for such devices (Bharti et al., 2022) bracket the period; the demonstration that logical error rates fall as a surface code is scaled (Google Quantum AI, 2023) and, later, that they fall below threshold (Google Quantum AI and Collaborators,

2025) mark the transition toward fault tolerance that the NISQ framing anticipated. The quantum-computing category is small, keyword-driven, and vetoes physics papers that lack a computing term (Appendix A); its dynamics are driven by hardware milestones and have no obvious mechanism of contact with language models, which makes it, with post-quantum cryptography, a natural placebo.

3.11. Scientific computing, computational biology and sustainable computing

Scientific computing entered the period through the door of physics-informed learning. Physics-informed neural networks, which impose a partial differential equation as a loss term (Raissi et al., 2019), and neural operators, which learn mappings between function spaces rather than between vectors (Li et al., 2020), created a literature that the review of Karniadakis et al. (2021) already presented as a field. Computational biology had its own transformation in the baseline window and does not need the post-2022 one to explain its trajectory: the first AlphaFold had won the 2018 structure-prediction assessment (Senior et al., 2020), the second effectively settled the single-chain problem in 2020–2021 (Jumper et al., 2021), protein language models predicted structure from sequence alone at scale (Lin et al., 2023), and the third AlphaFold extended prediction to complexes with nucleic acids and ligands (Abramson et al., 2024). The review of Wang et al. (2023) in 2023 took stock of "AI for science" across disciplines; the autonomous-scientific-discovery leaf of the ontology, which requires an AI or automation term, is the place where the corpus records the next step, in which models participate in the design of experiments rather than in the analysis of their results.

Sustainable computing, finally, was an established concern before 2022 but a small category. The careful recalibration of global data-centre energy use (Masanet et al., 2020) found that efficiency gains had largely offset growth in demand through 2018, a conclusion that the training and serving of large models would later put under pressure; the energy and carbon studies cited in Section 2.3 (Strubell et al., 2019; Schwartz et al., 2020; Patterson et al., 2021; Luccioni et al., 2022) connected the concern directly to AI. The category is keyword-driven and is expected to grow; whether its growth is driven by AI workloads or by the broader energy agenda is a question for Part IX.

Taken together, the baseline descriptions support a conclusion that the empirical parts must not forget: there was no AI-free area in 2022. Every non-AI category had a point of contact with machine learning before ChatGPT, ranging from marginal (complexity theory, post-quantum cryptography) to pervasive (information retrieval, software engineering, computational biology). The post-2022 question is therefore not whether AI *entered* these areas but whether the rate and the kind of contact changed, and the baseline window exists so that the rate can be estimated before the reference date.

4. November 2022 as a reference date

4.1. What ChatGPT was and was not

On 30 November 2022 OpenAI released ChatGPT as a "research preview": a conversational interface to a language model of the GPT-3.5 series, fine-tuned by the reinforcement-learning-from-human-feedback method of InstructGPT, free to use through a web browser (OpenAI, 2022). No technical report accompanied the release; the announcement described the training procedure by reference to Ouyang et al. (2022) and listed limitations (plausible but incorrect answers, sensitivity to phrasing, verbosity, incomplete refusal of harmful requests) that the subsequent literature would study at length. It is important for the design of this paper to be precise about what was new. The underlying capabilities, few-shot prompting, instruction following, code generation, had been published between 2020 and 2022 (Brown et al., 2020; Chen et al., 2021; Ouyang et al., 2022) and were available through a commercial programming interface. The architecture was five years old. What ChatGPT added was an interface that required no programming, a price of zero, and a conversational format that made the model's abilities and failures legible to anyone; its diffusion among the public was by all accounts faster than that of any previous software product, though the figures usually cited for it come from industry analysts rather than from verifiable sources and are not repeated here.

For a study of the *research* literature, this distinction has a consequence that is easily missed. The research community did not learn about large language models on 30 November 2022; it had been building and publishing them for three years. What changed on that date, if anything, was the attention of everyone else: funders, universities, firms, governments, and the researchers in subfields that had not previously had a reason to engage. If the date marks anything in the corpus, it should mark a change in *who* publishes on language models and *where*, more than a change in what the language-model community itself does. That expectation is formalised in H4 and H6 and in the reading of the cross-area citation flows in Part VIII.

4.2. The releases before and after

ChatGPT was one release in a dense sequence, and the sequence matters because any of its members is a candidate reference date. Table 2 lists the ones whose primary documents could be verified. Within the twelve months before November 2022 came the compute-optimal scaling result (Hoffmann et al., 2022), the PaLM, OPT and BLOOM models (Chowdhery et al., 2022; Zhang et al., 2022; BigScience Workshop, 2022), the chain-of-thought and emergent-abilities papers (Wei et al., 2022a,b), the second DALL-E and Imagen (Ramesh et al., 2022; Saharia et al., 2022), the open release of Stable Diffusion (Rombach et al., 2022), the Flamingo and Gato systems (Alayrac

et al., 2022; Reed et al., 2022), the Whisper speech model (Radford et al., 2022), and the ReAct formulation of language models that interleave reasoning with actions (Yao et al., 2022), posted in October 2022 and the direct ancestor of the agent literature. Within the twelve months after came LLaMA, the first competitive open-weight model family (Touvron et al., 2023b), Toolformer (Schick et al., 2023), the GPT-4 technical report (OpenAI et al., 2023), an early and much-debated qualitative evaluation of GPT-4, circulated as a preprint (Bubeck et al., 2023), Llama 2 with a permissive licence (Touvron et al., 2023a), the RT-2 vision-language-action model that transferred web-scale pretraining to robot control (Brohan et al., 2023), Gaussian splatting in graphics (Kerbl et al., 2023), the Mamba state-space architecture as the first serious challenger to attention (Gu and Dao, 2023), and the Gemini models (Team et al., 2023). Later milestones, Llama 3 (Grattafiori et al., 2024), AlphaFold 3 (Abramson et al., 2024), the o1 reasoning models (OpenAI et al., 2024) and DeepSeek-R1, which reported that reasoning behaviour could be elicited by reinforcement learning with verifiable rewards and released its weights (DeepSeek-AI et al., 2025), fall inside the observation window and are candidate dates for the *later* breaks that the changepoint methods of Part IV may find.

Table 2. Candidate reference dates around November 2022. "Document" is the verified primary source; dates are those of the first public version of the document or of the release it describes, as given in the source. The list is restricted to items cited in this Part and is not a complete chronology.

Date	Release or result	Document
2022-03	Compute-optimal scaling (Chinchilla)	Hoffmann et al. (2022), preprint
2022-04	PaLM; DALL-E 2; Flamingo	Chowdhery et al. (2022); Ramesh et al. (2022); Alayrac et al. (2022), preprints
2022-05	OPT; Gato; Imagen	Zhang et al. (2022); Reed et al. (2022); Saharia et al. (2022), preprints
2022-06	Latent diffusion (CVPR); open release as Stable Diffusion in August	Rombach et al. (2022)
2022-10	ReAct	Yao et al. (2022), preprint
2022-11	BLOOM; ChatGPT (30 November)	BigScience Workshop (2022), preprint; OpenAI (2022), web page
2022-12	Whisper	Radford et al. (2022), preprint
2023-02	LLaMA; Toolformer	Touvron et al. (2023b); Schick et al. (2023), preprints
2023-03	GPT-4	OpenAI et al. (2023), technical report (preprint)
2023-07	Llama 2; RT-2	Touvron et al. (2023a); Brohan et al. (2023), preprints
2023-08	3D Gaussian splatting (SIGGRAPH)	Kerbl et al. (2023)
2023-12	Mamba; Gemini	Gu and Dao (2023); Team et al. (2023), preprints
2024-05	AlphaFold 3	Abramson et al. (2024)
2024-07	Llama 3	Grattafiori et al. (2024), preprint
2024-12	o1 system card	OpenAI et al. (2024), preprint
2025-01	DeepSeek-R1	DeepSeek-AI et al. (2025), preprint

Two features of the table are relevant to identification. The first is that the great majority of the documents are preprints or technical reports; none of the model releases of 2022–2025 was described in a peer-reviewed venue at the time of release, and several never were. A corpus that excluded preprints would be blind to the events it is trying to date. The second is that the events are not sparse. Between March 2022 and March 2023 there was, on average, a landmark release every month, and a time-series method that is free to choose the break date will find a break somewhere in that interval for reasons that have nothing to do with ChatGPT in particular. That is the subject of the next subsection.

4.3. Why the date is used descriptively and not causally

The mandate of this study (CLAUDE.md, §6.5) states that the November 2022 boundary is a historical reference point and not proof of a causal effect of ChatGPT. The statement is a constraint on interpretation, but it is also a methodological position that can be made precise.

Let $N_{c,t}$ denote the number of works assigned to category c in month t, and consider the simplest model that the "post-ChatGPT" framing suggests, an interrupted time series of the form used in programme evaluation (Lopez Bernal et al., 2016a):

$$\log N_{c,t} \;=\; \alpha_c + \beta_c\, t + \delta_c\, \mathbf{1}[t \geq \tau] + \gamma_c\,(t-\tau)\,\mathbf{1}[t \geq \tau] + \mathbf{s}_c(t) + \varepsilon_{c,t}, \tag{2}$$

with τ fixed at December 2022, $\mathbf{s}_c(t)$ a seasonal component tied to the conference calendar, δ_c a level shift and γ_c a change in trend. Three conditions must hold for δ_c or γ_c to be read as the effect of the event at τ: the pre-period trend must be well specified, so that an acceleration already under way is not attributed to the break; nothing else that affects $N_{c,t}$ may change at τ; and the date τ must have been chosen independently of the data. The first condition is testable and is tested in Part IX with pre-period trend diagnostics. The second fails on its face: Table 2 lists a dozen co-occurring events, the compute trends of Sevilla et al. (2022) were already exponential, and the growth of the arXiv itself, documented month by month on its statistics page (arXiv, 2026), is a confounder that applies to every category. The third condition is the subtle one. The date was chosen because of what happened afterwards, which is the definition of selecting a break on the outcome; a test of a known break at τ (Lopez Bernal et al., 2016a) has the wrong size when τ was in fact selected from a set of candidates, and the appropriate tools are those for a break at an *unknown* date, the supremum-type tests of Andrews (1993), the multiple-break estimation of Bai and Perron (1998, 2003), and penalised changepoint search (Killick et al., 2012), which let the data locate the break and report whether its location is consistent with the reference date. Equation (2) is therefore the *descriptive* object of the study, a convenient summary of before and after; the inferential objects are the unknown-break estimates, their confidence sets, and the placebo breaks at other dates in the

baseline, as specified in Part IV.

The descriptive use of the date has a positive justification that goes beyond the avoidance of a fallacy. Dividing the window at a date that the whole community recognises makes the comparison between periods legible, and legibility is a scientific virtue when the quantities compared are themselves complicated. The baseline of 2018–2022 is long enough to contain the whole transformer era and the second half of the deep-learning wave, so that the "before" is not an AI-free past but a period of fast change in its own right; the research period of 2022–2026 is long enough for publication lags, which in computer science run from a few weeks for a preprint to more than a year for a journal, to have worked through for all but the last months. Any claim of the form "after November 2022, X increased" in this manuscript is to be read as a comparison of two periods so defined, with the structural-break analysis as a separate and more demanding question.

One further confounder is specific to the present study and is treated in the next section. If language models changed how papers are *written*, then every text-based measure, from keyword-driven classification to embedding-based topic detection, changed its meaning at approximately the reference date, and a break in such a measure may reflect a change in prose rather than in research.

5. What the literature already claims about the post-2022 period

A substantial literature has formed since 2023 on the consequences of large language models for science and for computer science. Its findings are the prior against which the corpus-level results of Part IX will be compared, and its limitations are the reason the corpus-level test is needed. Five claims recur.

5.1. The growth of AI research

That AI research is growing faster than other fields was established before ChatGPT. Krenn et al. (2023) built a semantic network from the AI and machine-learning literature of the preceding three decades, found that the number of papers in it had been growing exponentially with a doubling time of about two years, and used the network to forecast which concept pairs would be studied next. The Stanford AI Index reports, which compile counts from a range of sources, report continued growth in AI publications and in the number of notable models, with the share of such models produced by industry rising to a large majority by 2023 and 2024 (Maslej et al., 2024, 2025); the reports are compilations rather than primary studies, and their source definitions change from edition to edition, so their numbers are not comparable with those of this study and are cited here for the direction of the claim only. The arXiv's own monthly statistics show that computer-science submissions overtook those of every other archive during the 2010s and continued to accelerate after 2022 (arXiv, 2026). Industry's growing weight in the field, documented by Ahmed et al. (2023) for the period up

to 2022, is the component of this claim that the present study can test most directly, through the affiliation data of the corpus (H6).

What this literature does not establish is whether the growth after 2022 differs from the growth before it, either in rate or in kind, because the studies either end in 2022 or do not model the pre-period trend. H1 is the corpus-level version of the question.

5.2. LLM-assisted writing

The most direct evidence of a change in the literature after 2022 is a change in its language. In a preprint, Liang et al. (2024a) estimated, with a distributional method that measures the frequency of words whose use the models favour, that the fraction of abstracts in computer science with substantial LLM-modified content rose rapidly from the end of 2022, with computer science leading other fields; the same group found a comparable rise in the reviews of major machine-learning conferences (Liang et al., 2024b), and later extended the method to writing across society (Liang et al., 2025); both are cited here in their preprint versions. Kobak et al. (2025) obtained the same conclusion for biomedical abstracts by a different route, measuring the excess frequency of words such as "delves" and "underscores" relative to a pre-2023 baseline, in a study designed to need no training data from the models themselves; earlier and smaller preprint analyses of the same marker words had reached the same result (Geng and Trotta, 2024; Gray, 2024). Surveys of researchers confirm the adoption that the text analyses infer: a Nature survey of 1,600 scientists found wide use of generative tools for writing and coding, with strong concerns about reliability and misuse (Van Noorden and Perkel, 2023), and the multidisciplinary collection assembled by Dwivedi et al. (2023) and the integrity analysis of Bin-Nashwan et al. (2023) document the reactions of the research and education communities in the first months.

For this study the finding is a measurement problem before it is a result. The classification of Section 1.2 and the embedding methods of Part IV both read text, and text that is partly produced by a model trained on the pre-2023 literature is, in a precise sense, less informative about what is new in a paper. Two responses are built into the design. The venue and arXiv signals of equation (1) are immune to prose; and the robustness checks of Appendix D re-estimate the main breaks with classifications that use no keyword signals, so that a break present in the keyword-based series and absent in the venue-based one is flagged as possibly linguistic.

The measurement of LLM-generated text in *peer reviews* deserves separate mention because it bears on the quality of the venue signal. Liang et al. (2024b) estimated that a non-trivial share of the reviews submitted to several 2024 machine-learning conferences had been substantially modified by a language model, with the share higher among reviews submitted close to the deadline and among reviewers who did not respond to author rebuttals. If reviewing is partly delegated to models, then acceptance at a selective venue, which the ontology treats as the strongest community signal, is

partly a judgement rendered by the models under study. The corpus cannot measure this directly; it is recorded here as a limitation of the venue signal in the research period that does not apply to the baseline.

5.3. Illusions of understanding and the epistemology of AI in science

A second body of work concerns not the quantity of research but its quality. Messeri and Crockett (2024) argued that the uses of AI in science that researchers find most attractive, as oracle, surrogate, quant and arbiter, each create an illusion of understanding, and that the resulting monoculture of methods and of questions is a risk to the scientific enterprise rather than merely to individual studies. Birhane et al. (2023) raised the related concern that language models, used as instruments, embed the biases and the errors of their training corpora into the record. These arguments are theoretical and, so far, untested against the publication record; the knowledge-graph analyses of Part VIII, which track methods and datasets across papers, are the closest this study comes to a test of the monoculture claim.

5.4. Benchmark saturation and the reproducibility of evaluation

The benchmark culture of Section 2.2 produced a literature on its own dynamics. In a preprint, Ott et al. (2022) mapped the creation and saturation of benchmarks across AI tasks and found that many become saturated within a few years of their creation while others are abandoned before saturation; the critiques of Dehghani et al. (2021) and Raji et al. (2021) argued that the choice of benchmark is itself an under-examined research decision and that no benchmark can measure general capability. The post-2022 literature extended the concern to *contamination*, the presence of test data in the training corpora of models trained on the web, which makes a benchmark result uninterpretable without knowledge of the training data that the model reports of Table 2 typically withhold. Kapoor and Narayanan (2023) documented data leakage as a general cause of irreproducible results in machine-learning-based science, and the agenda of Hendrycks et al. (2021) placed robustness, monitoring and alignment among the open problems of the field. RQ7, which asks whether the period's advances are breakthroughs or incremental benchmark gains, and RQ9, on reproducibility and evaluation standards, are the corpus-level counterparts of these claims.

5.5. Concentration, disruption and the pace of science

The last claim is the broadest. Park et al. (2023) found, with a citation-based disruption index applied to papers and patents over six decades, that work has become less disruptive and more consolidating on average across fields; Chu and Evans (2021) found that in large and fast-growing fields, canonical papers become entrenched, new ideas are less likely to displace them, and progress

in the sense of turnover at the top slows. Both results predict that a field growing as fast as AI after 2022 should show rising concentration of citations on a few canonical works and a lower rate of disruption, independent of any product release. Hypothesis H5, on citation inequality, is formulated with that prediction in mind, and the difference between the post-2022 period and the baseline is the quantity of interest, not the level.

5.6. Why the claims need a corpus-level test

The studies reviewed above share four limitations that the design of this paper is intended to remove. They are, almost without exception, studies of AI alone, so that they cannot say whether the growth they document came at the expense of, or in addition to, the rest of computer science; the present corpus covers all forty categories with the same instrument. They rely on a single source (arXiv, a conference's reviews, a database of abstracts), so that coverage effects of the kind described in Section 1.3 cannot be separated from research effects; the present corpus links four sources and reports results by source. They use vocabulary as evidence, which, as Section 5.2 shows, is the quantity most directly altered by the models; the present design includes classifications that use no vocabulary. And they date the change to the release of ChatGPT without testing the alternative dates of Table 2 or the placebo dates of the baseline; the present design does both.

None of this guarantees that the corpus-level test will overturn the claims; it may confirm them. Its contribution is to subject them, for the first time at the scale of the whole discipline and with a preregistered protocol, to the conditions under which a claim of structural change can be distinguished from the ordinary growth of a large field. Part III describes the corpus on which that test is run.

Part III

Data and Systematic Review Protocol

6. Information sources and conditions of access

A field-level mapping study stands or falls with the bibliographic databases on which it is built. The choice of sources determines which works can be observed at all, how reliably they can be dated and typed, whether citation links are available, and under what licence derived data may be redistributed. This part documents these choices for WP44 and reports, stage by stage, how the mapping corpus was assembled from the raw sources. The exposition follows the order of the pipeline (identification, linkage, deduplication, eligibility, scope, dating) and closes with the quality indicators and the

research-integrity procedures that govern every later analysis. The reporting structure is modelled on the PRISMA 2020 items that apply to a mapping study (Page et al., 2021b,a); items concerned with risk of bias and effect measures have no counterpart in a bibliometric design and are replaced, as PRISMA-ScR does for scoping reviews (Tricco et al., 2018), by indicators of metadata quality and coverage (Section 13). The search-reporting recommendations of PRISMA-S (Rethlefsen et al., 2021) are followed to the extent that a database-scope strategy without keyword queries permits.

6.1. Why a multi-source design

No single database indexes computer science completely, and the gaps are not random. The venue-centred coverage of DBLP, the publisher-centred coverage of Crossref, the preprint-centred coverage of arXiv and the aggregating coverage of OpenAlex each miss a distinct part of the literature and type the part they do cover in distinct ways. Comparative studies of bibliographic data sources have documented large differences in coverage between Web of Science, Scopus, Dimensions, Crossref and Microsoft Academic (Visser et al., 2021; Martín-Martín et al., 2021), and the journal-level coverage of the two commercial indexes is known to under-represent conference proceedings, which carry a large share of computer-science output (Mongeon and Paul-Hus, 2016). For the open successor of Microsoft Academic, OpenAlex, the first independent assessments report strong coverage of works with DOIs together with weaker and uneven metadata on affiliations, document types and references (Alperin et al., 2024; Culbert et al., 2025; Scheidsteger and Haunschild, 2023; Delgado-Quirós and Ortega, 2024). These findings motivate the design adopted here: one source supplies the backbone (OpenAlex), a second supplies the definition of what counts as computer science (DBLP), a third supplies the preprint layer and its dates (arXiv), and a fourth checks the coverage of a subfield with a dense, curated venue structure (the ACL Anthology). Crossref is used for verification only.

6.2. OpenAlex

OpenAlex (Priem et al., 2022) is the backbone of the corpus. It supplies, for every work, the identifiers (OpenAlex W identifier, DOI, arXiv identifier where present), the publication date and type, the title and an inverted-index abstract where available, the venue (source) and its type, the authorships with institutions and countries, the topic assignment at three levels (field, subfield, topic) with a primary topic, the list of referenced works (the citation graph), the open-access status and licence, and the field-weighted citation impact together with yearly citation counts. OpenAlex is released under the CC0 public-domain dedication, which permits redistribution of record-level derived data.

Two access routes exist and both were used, for different purposes. The public snapshot is a

versioned bulk release in gzipped JSON Lines, published in a public S3 bucket; the release used here is dated 2026-10-14 and comprises 1,898 part files for the *works* entity, 675.3 GB compressed, with a manifest listing the record count of each file. The snapshot was streamed from the primary compute node, each file's content length checked against the manifest and a SHA-256 of the compressed bytes recorded in a SQLite ledger before parsing; every one of the 1,898 files parsed without error and the sum of the per-file record counts equals the manifest total of 464,765,485 works. Two outputs were written during this single pass. The first is the full JSON record of every work carrying *any* topic in OpenAlex field 17 (Computer Science) with a publication year between 2018 and 2026, stored as Zstandard-compressed JSON Lines and partitioned by the snapshot's own date partitions (15,490,159 records, about 20 GB). The second is a *slim* table of every work in the snapshot (identifier, DOI, year, type, primary topic, subfield, field and domain, citation count and a few flags), kept in Parquet so that works cited by computer-science papers but published in other fields can be resolved for the interdisciplinary analyses of Part VIII (decision D2 of `RESEARCH_PLAN.md`). Records flagged by OpenAlex as XPAC expansion records are retained at this stage with their flag so that the exclusion decision can be taken, and counted, later (Section 10). The smaller entity snapshots (topics, subfields, fields, domains, sources, institutions, publishers, funders, concepts) were downloaded the same day with a per-file SHA-256 ledger; author entities were not downloaded because the authorship data embedded in each work record suffice for the analyses planned.

The second route is the REST API. Without an API key the public endpoint operates under a shared daily budget of USD 0.10, list-and-filter calls are priced at USD 0.0001 each, and cursor paging is unavailable, so that a single filtered query cannot return more than 10,000 results through basic paging. Single-record lookups by identifier are free. The API was therefore used for two narrowly defined tasks. First, an incremental harvest of works with any computer-science topic and a publication date between 2026-10-01 and 2026-11-02, intended to capture records added after the snapshot was cut; the harvest is sliced by publication day and, where a day exceeds the paging limit, split into records with a computer-science primary field and records without one. Raw JSON pages are stored verbatim under a dated directory with a ledger of counts and cost, and the job resumes on the following day when the budget is exhausted. Second, free single-record lookups by DOI serve as one of the verification routes for the bibliography of this manuscript (Section 14). A free API key, which would raise the daily budget, requires the principal investigator to act on an e-mail and was deliberately not obtained without instruction; the mandate forbids assuming credentials that have not been tested.

6.3. DBLP

The DBLP computer science bibliography (Ley, 2002, 2009) is a curated, venue-centred index that has for three decades defined, in practice, the set of journals and conference series regarded as

computer science. It is used here in that capacity: a work indexed by DBLP is in computer science by construction, and DBLP venue streams feed the taxonomy of Part IV. The complete XML dump (`dblp.xml.gz`, dated 2026-11-02 22:39 UTC, 1.12 GB) was downloaded together with its DTD and MD5 checksum, the checksum verified, and the file parsed with a validating XML parser into one row per publication record (articles, conference papers, proceedings volumes, books, book chapters, doctoral and master's theses, data records; the `www` person records are skipped). The whole archive was kept so that per-year counts are available for the historical baseline and for context. The DBLP dump is released under the Open Data Commons Attribution licence (ODC-BY 1.0). DBLP contributes, in addition to the scope definition, author person identifiers, venue streams, and DOI links for most records; the `ee` fields were also scanned for arXiv URLs, with a result reported in Section 8.

6.4. arXiv

The arXiv e-print repository (Ginsparg, 2011) is the main preprint channel of computer science and, for the period under study, the place where a large share of the most cited work first appeared. Its metadata are harvested through the OAI-PMH interface (`set=cs`, `metadataPrefix=arXiv`) in monthly datestamp windows from January 2018 to 2 November 2026. The harvester waits at least 3.5 s between requests and honours `Retry-After` headers; every raw XML page is kept verbatim as the immutable snapshot and parsed afterwards into a single Parquet table. Two properties of OAI-PMH matter for what follows. The `from`/`until` parameters filter on the record's *datestamp*, the time of its last update, so that a monthly window contains every record updated in that month, not every record first submitted then; duplicates across windows are resolved at parse time with the latest datestamp winning. The `created` element gives the first submission date, which is the date used throughout this study for preprints. The harvest returned 915,115 distinct records, of which 890,852 were first submitted in 2018–2026; the remainder are older e-prints updated during the window. The arXiv metadata are released for reuse under the repository's metadata terms; e-prints themselves are governed by per-paper licences and are downloaded, for the high-priority set only, within those licences and at the repository's rate limits. The use of arXiv as a research dataset, and the caveats attached to its category labels and versioning, are discussed by Clement et al. (2019).

6.5. ACL Anthology

The ACL Anthology is the curated archive of the computational-linguistics community and has itself served as a bibliometric corpus (Radev et al., 2013a). Its BibTeX export with abstracts (`anthology+abstracts.bib.gz`, 46.7 MB, SHA-256 recorded) was downloaded on 2026-11-02 and parsed into a table of 138,796 entries, of which 135,558 are conference papers or journal articles

and 85,625 fall in 2018–2026. The Anthology plays a diagnostic role: because its venue structure is complete and hand-maintained, the share of its entries that the backbone can be matched to is a direct measure of OpenAlex coverage for a subfield that publishes overwhelmingly in conference proceedings. It also supplies abstracts for those entries in which OpenAlex has none. The export is licensed CC BY 4.0.

6.6. Crossref, Semantic Scholar and other sources

Crossref (Hendricks et al., 2020) is the registration agency behind most DOIs in the corpus and offers open metadata through a REST API with a polite pool. It is not used to build the corpus, whose backbone already carries Crossref-derived metadata through OpenAlex, but for three verification purposes: resolution of every DOI cited in this manuscript, a cross-check of publisher-assigned document types on samples, and a cross-check of citation counts (`is-referenced-by-count`) on a sample of highly cited works in Part IX.

The Semantic Scholar Academic Graph (Kinney et al., 2023a) was listed in the mandate among the priority sources because of two signals it alone provides, the influential-citation classification and citation intent. At the time of the feasibility audit, anonymous requests to its API returned HTTP 429 with an instruction to apply for a key, and the bulk-dataset endpoint requires a key outright. In keeping with the rule that inaccessible data be recorded rather than circumvented, Semantic Scholar is recorded as unavailable (decision D3) and no analysis in this manuscript uses influential-citation or citation-intent signals; where Part VIII discusses the strength of citation links, it does so with OpenAlex reference lists and the knowledge-graph extraction of Part IV, whose limits are stated there. The Microsoft Academic Graph (Wang et al., 2020), from which OpenAlex inherited much of its historical coverage, was discontinued at the end of 2021 and is not an independent source for the period studied. Kaggle credentials for the arXiv snapshot distributed there were absent and were not needed, OAI-PMH being sufficient. Benchmark repositories and code-hosting platforms enter only in the reproducibility and benchmark analyses scoped in Part IV and are not part of the bibliographic corpus.

Table 3 summarises the sources, the access route, the licence and the role of each.

7. Identification: what each source contributes

There is no keyword search in this study. A mapping of an entire discipline cannot be defined by a query string without importing the vocabulary of whoever wrote it; it is defined instead by database scope filters that are themselves documented and reproducible, which is the strategy recommended for mapping studies in software engineering and adopted here for computer science as a whole (Petersen et al., 2008, 2015). The identification stage (S0 in the notation of the research plan)

Table 3. Information sources of the WP44 corpus (all accessed 2026-11-02/03).

Source	Access and snapshot	Licence	Role
OpenAlex works snapshot	public S3 bucket, release 2026-10-14, 1,898 JSONL files, 464,765,485 works	CC0	backbone: identifiers, dates, types, topics, authorships, references, abstracts, OA status
OpenAlex REST API	no key; USD 0.10 per day shared budget; increment 2026-10-01 to 2026-11-02; single lookups	CC0	post-snapshot increment; reference verification
DBLP	XML dump 2026-11-02 22:39 UTC, MD5 verified	ODC-BY 1.0	computer-science scope; venue streams; author identifiers
arXiv	OAI-PMH set `cs`, 2018-01 to 2026-11, raw pages kept	arXiv metadata terms	preprint layer; first-submission dates; categories
ACL Anthology	BibTeX export with abstracts, 2026-11-02	CC BY 4.0	coverage check for NLP venues; abstracts
Crossref	REST API, polite pool	open metadata	verification of DOIs, types and citation counts
Semantic Scholar	API key required; anonymous calls refused	—	*unavailable, recorded*

therefore consists of four scope filters, one per source, applied to immutable snapshots whose dates and hashes are recorded in `data/raw/README.md`.

7.1. Scope filters

The OpenAlex filter retains every work with at least one topic, primary or secondary, in field 17 and a publication year in $[2018, 2026]$. The choice of *any* topic rather than the primary topic is deliberate and conservative at this stage: a work whose primary topic lies in, say, statistics or electrical engineering but which carries a secondary computer-science topic is a candidate, and the decision whether it belongs to the core or to the penumbra is deferred to the eligibility stage, where it is made explicit and counted (Section 11). The filter yields 15,490,159 records in the first pass over the snapshot. The DBLP filter retains the publication records of type article, inproceedings, incollection, book, phdthesis and data with a year in $[2018, 2026]$: 4,347,365 records. The arXiv filter retains the records of set `cs` whose first submission falls in the window: 890,852 records. The ACL filter retains the conference papers and journal articles of the Anthology with a year in the window: 85,625 records. The four counts overlap heavily, since most DBLP and ACL records and a majority of arXiv records also exist in OpenAlex; the overlap is resolved by record linkage, not by summation, and no total of identified records is reported as if the sources were disjoint.

7.2. The OpenAlex topic classifier as a scope instrument

Because the OpenAlex filter relies on an automated topic classifier, its properties bear on everything downstream. OpenAlex assigns to each work up to three topics from a fixed list of 4,516, organised

in 252 subfields, 26 fields and four domains; the assignment is made by a model applied to the title, abstract and venue, and the field label of a work is inherited from its topics rather than assigned directly. Two consequences follow. A work can be labelled computer science because one of its topics belongs to field 17 even when its content is only marginally computational, and conversely a work in a computer-science venue can receive no computer-science topic when its abstract is missing or its vocabulary is domain-specific. The first error inflates the candidate set with what will be called the penumbra; the second removes genuine computer-science works from the first-pass candidate set. DBLP is used to correct the second error, which is why the linkage stage has two passes (Section 8). The literature on OpenAlex has so far examined its coverage and metadata completeness rather than the accuracy of its topic labels (Alperin et al., 2024; Delgado-Quirós and Ortega, 2024); the present study treats the labels as a scope instrument whose errors are bounded by the DBLP and arXiv memberships and quantified in Section 13, not as ground truth.

7.3. Verification of the scale against the mandate

The mandate set planning targets of more than one million candidate records and more than one hundred thousand deeply classified records, subject to availability. The feasibility audit established, before any ingestion, that the OpenAlex API alone reported more than six million works with a computer-science primary field in 2018–2026 and that the DBLP dump held more than four million records in the same years; the targets were therefore attainable without recourse to any restricted source. The audit also established what was *not* attainable: a Semantic Scholar key, an OpenAlex key, Kaggle credentials and the AWS command-line tool were all absent, and the ingestion was designed around their absence (HTTPS streaming of the public bucket, basic paging of the public API). The record of this audit is `reports/PHASE0_FEASIBILITY.md`.

8. Record linkage across sources

The four candidate sets describe overlapping populations with different identifiers. Record linkage (stage S1) establishes, for each DBLP, arXiv and ACL record, the OpenAlex work it describes, so that one node per work can be formed with the union of the metadata. The methods are deterministic and ordered by reliability; the probabilistic framework of Fellegi and Sunter (1969) is not needed because the available keys (DOI, arXiv identifier) are close to exact, and the residual title-based step is restricted to blocks where it is unambiguous. The implementation follows the standard decomposition of data matching into key normalisation, blocking, comparison and classification (Christen, 2012); Part IV gives the formal statement of the keys and of the precision-estimation design, and this section reports what was done and what it produced.

8.1. Keys and matching order

All keys are normalised before comparison. DOIs are lower-cased. arXiv identifiers are lower-cased and stripped of their version suffix (`v2`, `v3`, ...), so that all versions of an e-print map to one identifier. Titles are reduced to a normalised string by lower-casing and deleting every character that is not a letter or digit; the normalised title is used only when it has at least 25 characters, is unique within its source, and is unique within OpenAlex, which is the blocking rule that keeps the title step from producing spurious links between generic titles.

For DBLP, the order is: (i) DOI equality, (ii) arXiv identifier equality, restricted to records not yet linked on either side, (iii) normalised title equality with a year difference of at most one, restricted to records not yet linked on either side. The DOI step linked 2,562,431 records and the title step 179,990; the arXiv step linked none, because the extraction of arXiv URLs from the DBLP `ee` fields returned no candidate (0 records with an extracted identifier). The `ee` fields of this dump evidently do not carry `arxiv.org/abs` URLs in the form the extractor expects; whether the CoRR entries are reached by the DOI step instead depends on DBLP's current conventions for the `ee` field, which the audit did not characterise, and the question is left to the linkage-precision review. The step is kept in the pipeline because other dumps may differ. In total 2,736,903 of the 4,347,365 DBLP records in the window were linked to an OpenAlex work after the first pass, leaving 1,610,462 unlinked. The composition of the unlinked set is informative: 1,084,640 journal articles, 477,235 conference papers, 26,841 doctoral theses, 16,986 data records, 3,362 chapters and 1,398 books. Of these, 1,077,407 carry a DOI that *does* exist in the slim table of all OpenAlex works, which means the work is in OpenAlex but was not caught by the computer-science topic filter; this is the second error of the topic classifier described above and the reason for the second pass.

For arXiv, the order is: (i) arXiv identifier equality against the identifier OpenAlex records for the work, (ii) DOI equality for e-prints that carry a journal DOI, (iii) normalised title equality with a year difference of at most two, the wider tolerance reflecting the typical lag between a preprint and its published version (Lin et al., 2020). The steps linked 792,672, 520 and 2,143 records respectively, for 744,680 distinct e-prints linked. For the ACL Anthology, the order is: (i) DOI equality, (ii) equality between the Anthology identifier and the identifier embedded in an `aclanthology.org` landing-page URL among the work's OpenAlex locations, (iii) normalised title equality with a year difference of at most one. The steps linked 57,474 records by DOI and 14,904 by title, with the URL step contributing the remainder, for 72,223 distinct 2018–2026 entries linked out of 85,625; the residual is examined in Section 13 as a measure of OpenAlex coverage of conference-heavy subfields.

8.2. The second snapshot pass

A DBLP record whose DOI resolves to an OpenAlex work outside the topic filter is, by the scope rule, a computer-science work that the first pass did not load. The same situation arises for arXiv e-prints whose journal DOI exists in OpenAlex without a computer-science topic. The linkage script therefore writes the list of OpenAlex identifiers that are needed but not loaded: 1,077,406 identifiers from DBLP DOIs and 39,200 from arXiv DOIs. A second pass over the works snapshot, restricted to this identifier list, retrieves the full records of these works; they are normalised into the same Parquet tables with a `_p2` suffix, the DuckDB views are rebuilt over the union of both passes, and linkage is re-run so that the DBLP-linked set gains the works that the topic classifier had missed. The second pass is what makes the DBLP membership rule of Section 11 operational: without it, "DBLP-indexed" would silently mean "DBLP-indexed and labelled computer science by OpenAlex", and the DBLP correction would be void precisely where it matters. DBLP records with no OpenAlex counterpart after both passes (1,610,462 records before the second pass) are kept in a separate table with their year and type and are reported as such; they enter the volume counts of Part IX only in the sensitivity analysis that adds DBLP-only records, because they lack the abstracts, references and affiliations on which every other indicator depends. Most of them are journal articles and conference papers with a DOI that OpenAlex does not hold at all, together with doctoral theses, which DBLP indexes from university repositories that OpenAlex covers unevenly.

8.3. Precision of the linkage

Exact-key links are correct whenever the key is correct in both sources; the known failure modes are DOIs reassigned by publishers between versions, DOIs registered for a proceedings volume rather than a paper, and e-prints whose OpenAlex record points to a different version. Title-based links can fail on near-identical titles of distinct works (extended abstracts and full papers, workshop and conference versions). The protocol pre-specifies the estimation of linkage precision on a sample of 500 DBLP–OpenAlex pairs stratified by method, with automatic agreement checks (normalised-title similarity, year difference, first-author surname) and a 200-pair subsample reserved for the principal investigator (checkpoint H-1). The sampling script draws 250 DOI links, 200 title links and 50 arXiv links with a fixed seed; since the arXiv step produced no links in this dump, the realised sample has 450 pairs, and the precision estimate for the arXiv step is undefined rather than reported as perfect. At the time of writing the sampling script exists but the sample has not been drawn and the human checkpoint is unsigned; the automatic agreement rates and the human verdicts will be reported in Part IX and Appendix B, and every conclusion that depends on the title-based links is labelled as pending human review until H-1 is signed.

9. Deduplication of preprint and published versions

OpenAlex creates distinct work records for an arXiv e-print and for the journal or conference version of the same paper whenever the two carry different DOIs, which is the normal case since arXiv registers DataCite DOIs for its e-prints. In a corpus where preprinting is widespread this is the dominant form of duplication, and it biases every count in the direction of fields that preprint most: a paper in machine learning that appears on arXiv and then at a conference would be counted twice, a paper in a theory journal once. OpenAlex itself links some preprint–publication pairs, but the links are incomplete, and prior work on discovering such links has shown that neither DOIs nor titles alone suffice (Cabanac et al., 2021). Stage S2 therefore merges preprint and published versions into one node per work, in the sense required by the mandate (one node per *work*, with both dates retained).

9.1. Pairing rule

The rule is conservative by design, favouring a small number of certain merges over a larger number of probable ones. A *preprint-like* record is one whose OpenAlex type is `preprint` or whose DOI is an arXiv DataCite DOI (prefix `10.48550/arxiv`); a *published-like* record is any other record. Two records are paired when their normalised titles are identical, both titles have at least 25 characters, and their publication years differ by at most two. When a published record matches several preprints, the earliest preprint is kept as its preprint; when a preprint matches several published records, the earliest published record is kept; all other pairs are discarded, so that the pairing is one-to-one. The pairing is confined to the loaded OpenAlex records; preprints that are not computer-science candidates cannot be paired, which is immaterial since they would not enter the corpus. No fuzzy title similarity is used at this stage, so that a preprint whose title changed at publication is not merged; the consequence, a residual double count, is discussed below.

9.2. What is kept from each member of a pair

The published record is the surviving node. The preprint record receives the flag `duplicate_of` with the identifier of the published work and is excluded with reason code E5 (Section 10). The published node receives `preprint_work_id` and, through the arXiv linkage, the first-submission date of the e-print, so that for every work the *first date* is defined as the earliest of its OpenAlex publication date, the publication date of its preprint record and the arXiv `created` date. This first date is the date variable of all monthly series in Part IX; the publication date of the version of record is kept alongside it. The distinction matters for structural-change analysis: dating a paper by its journal issue would displace much of the research activity of 2023 into 2024 and 2025 and would

blur any monthly series around the November 2022 boundary, whereas the first date places the work at the moment it entered the public record. Citation counts are those of the surviving published node; OpenAlex attributes to the preprint record the citations that cite the preprint DOI, and these are not added to the published node, a known under-count for works whose preprint remained the cited version. Part IV states the resulting measurement assumption and Part IX reports, for the high-priority set, the share of citations that accrue to preprint records.

9.3. Outcome and residual duplication

The pairing identified 114,115 preprint records as duplicates of a published work, of which 106,928 remained after the earlier exclusions had been applied. Set against the 624,265 preprints that remain in the eligible corpus as works with no published counterpart, the figure shows that most e-prints in the candidate set were not matched to a published version within OpenAlex. Three explanations coexist and cannot be separated without manual review: many e-prints are genuinely unpublished, consistent with the publication rates reported for computer-science preprints (Lin et al., 2020); some published versions carry a modified title; and some published versions are not in OpenAlex or lack a computer-science topic. The first explanation is benign; the second and third leave a residual double count whose size is bounded in Part IX by re-running the pairing with a title similarity threshold rather than exact equality and reporting the difference as a sensitivity, not as a correction. Because arXiv-linked works carry their e-print identifier, the sensitivity analyses can also remove every preprint node whose e-print has a journal DOI in the arXiv metadata, whether or not the pairing found the published record.

10. Eligibility criteria and exclusions

Stage S3 applies the eligibility criteria of the preregistered protocol (`protocol/PROTOCOL.md`, P2) to every loaded record and records, for each, a set of Boolean exclusion flags with reason codes. The flags are computed independently so that each code can be counted on its own and also sequentially, in the order E1–E7, so that the PRISMA flow (Figure 1) sums correctly. Reporting both views is deliberate: the independent counts show how large each problem is in the raw data, the sequential counts show how many records each criterion removed given those before it.

10.1. Reason codes

E1, outside window. The publication date is missing, earlier than 2018-01-01 or later than 2026-11-02. Because the first-pass filter already restricted the publication year, this code catches only records with a missing date or a date beyond the extraction date. Forward-dated records, that is

works whose publication date lies in the future relative to the snapshot, are a documented feature of OpenAlex, where publishers register articles for forthcoming issues; the window end removes those dated after the extraction date, and the remaining 2026 records are discussed in Section 12. 160 records are flagged.

E2, non-research document type. The protocol lists the document types that constitute research output: article, conference paper, preprint, book chapter, book, dissertation, report, review and software paper. The exclusion operates asymmetrically. A record that is *not* DBLP-linked is excluded when its OpenAlex type is outside the list. A record that *is* DBLP-linked is kept whatever its OpenAlex type, unless that type belongs to a hard-exclusion list of types that are never research outputs (editorial, erratum, retraction notice, peer review, paratext, conference abstract, reference entry, dataset, software, supplementary material, library guide, letter, grant, standard). The asymmetry reflects amendment A-001 and is explained below. Independently of order, 3,961,633 records carry this flag. The largest contributors are records typed as datasets (2,775,759), followed by the residual type `other` (522,654), library guides (191,658), software (177,500), conference abstracts (66,267) and peer reviews (60,176); paratext, reference entries, editorials, supplementary materials, errata, standards, book reviews, retraction notices, data papers and letters make up the remainder. The prominence of datasets and software among records carrying a computer-science topic is itself a finding about the backbone: OpenAlex indexes the DataCite registry, where repositories mint DOIs for research objects that are not publications, and the topic classifier labels them by their descriptions.

E3, XPAC expansion records. OpenAlex marks as XPAC the records it added from non-Crossref sources in its coverage expansion; these carry sparser metadata and a higher share of non-research objects. The protocol excludes them, and 3,950,876 records carry the flag independently. Amendment A-001 introduced one exception: an XPAC record that is linked to a DBLP record is a computer-science publication whatever its provenance in OpenAlex, and it is kept with the flag `xpac_kept_dblp` (87,288 records). Sequentially, after the type criterion, 1,421,692 records are removed under E3, far fewer than the independent count because most XPAC records are also non-research types.

E4, no title. Records with a missing or empty title cannot be linked, deduplicated or embedded and are excluded: 16,426 independently, 1,188 sequentially.

E5, duplicate version. The preprint member of a merged pair (Section 9): 114,115 independently, 106,928 sequentially.

E6, retracted. Works flagged `is_retracted` by OpenAlex, which draws on the Retraction Watch data now distributed through Crossref, are *not* removed from the mapping corpus: they are part of the published record and of its volume, and their presence is itself an object of study given the record numbers of retractions reported for 2023 (Van Noorden, 2023). They are kept with the flag

and excluded from every impact analysis (citation rankings, landmark selection, field-normalised scores). 25,165 loaded records carry the flag, of which 14,412 are in the eligible corpus.

E7, generic deposit platform. Introduced by amendment A-002, this code excludes records whose primary source is a generic deposit platform (Zenodo, Figshare, the Japanese Institutional Repositories DataBase, LA Referencia, Dialnet, the DOAJ-as-source entries and the OSTI OAI feed) unless the record is DBLP- or arXiv-linked. The number of records removed under this code is reported by the macro [pending human review: PrismaExcludedSequentialESevenGenericDeposit] once the corpus has been rebuilt under A-002.

One inconsistency of numbering must be recorded. In the frozen protocol, code E7 designates the penumbra of works with a secondary computer-science topic only; the implementation reuses the number E7 for the generic-deposit exclusion of amendment A-002 and carries the penumbra as a separate flag, `cs_penumbra`. The penumbra is not an exclusion in the same sense as E1–E7: it is a scope boundary, reported separately and used in sensitivity analyses, and is treated in Section 11. The conflict of numbering is flagged here for correction in the amendments log before the protocol is cited in its final form.

10.2. Amendments A-001 and A-002 and their evidence

Two eligibility decisions were taken after the protocol freeze, on the evidence of the first run of the corpus-definition script, and both are logged in `protocol/amendments.csv` with their date, their reason and their status (executing agent, pending confirmation by the principal investigator at checkpoint H-0). They are reported here in full because an eligibility rule changed after preregistration is exactly the kind of decision a reader must be able to audit (Nosek et al., 2018).

A-001, document types. The protocol's type list was written from the OpenAlex documentation available at the time; the 2026 snapshot uses the types `conference-paper` and `software-paper`, which the list did not contain. The first run of the eligibility stage excluded under code E2 a very large number of records typed `conference-paper`, close to half of them DBLP-linked; the figures of that superseded run are recorded in the amendment entry and are not repeated here, since the run was not retained. Conference papers are the principal publication form of several areas of computer science, and their exclusion would have invalidated the composition analyses of Part IX. The amendment adds the two types to the list and, more consequentially, makes DBLP linkage decisive: a DBLP-linked record is a computer-science research output whatever OpenAlex calls it, unless its type is one that can never be a research output. The same amendment keeps DBLP-linked XPAC records. The corrected run yields the counts reported above, in which 1,840,499 eligible works are conference papers.

A-002, generic deposit platforms and date precision. Inspection of the eligible corpus by source (`systematic_review/inspect_sources.py`) showed that a single generic repository, Zenodo,

contributed several times more eligible records dated 2026 than dated 2025, that almost none of them carried a reference list, and that random samples were slide decks, datasets, software releases and project reports typed as articles because the depositor had chosen that type; the counts are recorded in the amendment entry. Left in the corpus, such records would have produced a spurious surge of 2026 output concentrated in a repository that does not publish research articles, and would have done so in the months where right truncation already makes the series fragile. The amendment excludes records whose primary source is a generic deposit platform unless DBLP or arXiv vouch for them, and introduces the `date_precision` flag discussed in Section 12. The rule is a source-level rule, not a content judgement: a genuine research report deposited only on Zenodo is lost, which is accepted because the alternative, keeping every deposit, contaminates the primary estimand of RQ1. The sensitivity analyses of Part IX report the monthly series with and without the generic-deposit exclusion.

The amendments illustrate a general point about preregistered bibliometrics. The protocol can freeze the *rules* but not the data model of a database that changes between the time of writing and the time of extraction; when the data model changes, the honest course is an amendment with its evidence, not a silent adaptation.

11. The computer-science core, the penumbra and the strict core

The decision of what counts as computer science is the single most consequential modelling choice in a discipline-level mapping, and the protocol makes it with a three-part membership rule (decision D4) rather than with any one source. A record that survives the exclusions E1–E7 belongs to the *core* if at least one of the following holds:

1. it is linked to a DBLP record (venue-based membership);
2. its OpenAlex primary field is Computer Science (classifier-based membership);
3. it carries a secondary computer-science topic *and* is linked to an arXiv e-print whose primary category is in the `cs` archive (author-declared membership).

A record that survives the exclusions but satisfies none of these, which means it carries a secondary computer-science topic without venue or author support, belongs to the *penumbra*. The three conditions correspond to three independent judgements about disciplinary membership: the judgement of DBLP's editors about venues, the judgement of OpenAlex's classifier about content, and the judgement of the authors who chose a primary arXiv category. Requiring agreement of any one of them with the first-pass topic filter removes the over-inclusion that a secondary topic alone would produce, while keeping works that any single source would have missed.

11.1. Counts

After the sequential exclusions, 9,998,602 loaded records remain. Of these, 6,403,238 satisfy the membership rule and constitute the eligible mapping corpus, and 3,595,364 form the penumbra. The penumbra is thus a substantial fraction of the records that a naive "any computer-science topic" filter would have called computer science; its exclusion from the primary analyses is the main difference between the present corpus and what an OpenAlex query alone would return, and it is the reason why no count in this manuscript should be compared directly with the counts that the OpenAlex interface displays for field 17. Within the eligible corpus, the three membership conditions overlap: 2,640,780 works are DBLP-linked, 5,619,140 have a computer-science primary field, and 709,460 are linked to an arXiv e-print; 70,078 are ACL Anthology papers. The eligible corpus divides by period into 2,944,556 works whose first date falls in the baseline window, 3,457,315 in the post-boundary window, and a small residue of 1,367 whose first date, taken from a preprint or an e-print, precedes 2018 although the version of record falls in the window; the residue is kept in the corpus and assigned to the "pre" period, so that it enters no monthly series. Figure 1 renders these counts as a PRISMA-style flow, drawn from the same JSON file that generates the macros of this section (cf. Haddaway et al., 2022).

11.2. The strict core

The membership rule is a disjunction, and a sceptical reader may object that a disjunction of three imperfect judgements inherits the errors of each. The protocol anticipates the objection with a *strict core*, defined as the conjunction of the first two conditions: works that are both DBLP-linked and labelled computer science by the OpenAlex classifier. The strict core contains 1,868,188 works. Every headline result of Part IX is re-estimated on the strict core, and the manuscript reports a result as robust to scope only when the sign and the order of magnitude agree between the eligible corpus and the strict core. The strict core is not a better corpus; it is a narrower one, biased toward established venues and against preprints, dissertations and the newer conference series that DBLP has not yet adopted, and its role is to bound the influence of scope decisions, not to replace them.

11.3. What the design does not do

Three limits deserve statement. First, DBLP membership is venue membership: a paper in a DBLP-indexed journal of bioinformatics or operations research is in the core even when its content is not computational, and the taxonomy of Part IV, not the scope rule, is what distinguishes such works. Second, the rule has no language component; works in any language are eligible, and the share of non-English records, which is concentrated in dissertations and in regional journals, is reported among the quality indicators. Third, the rule is applied to the records loaded by the two

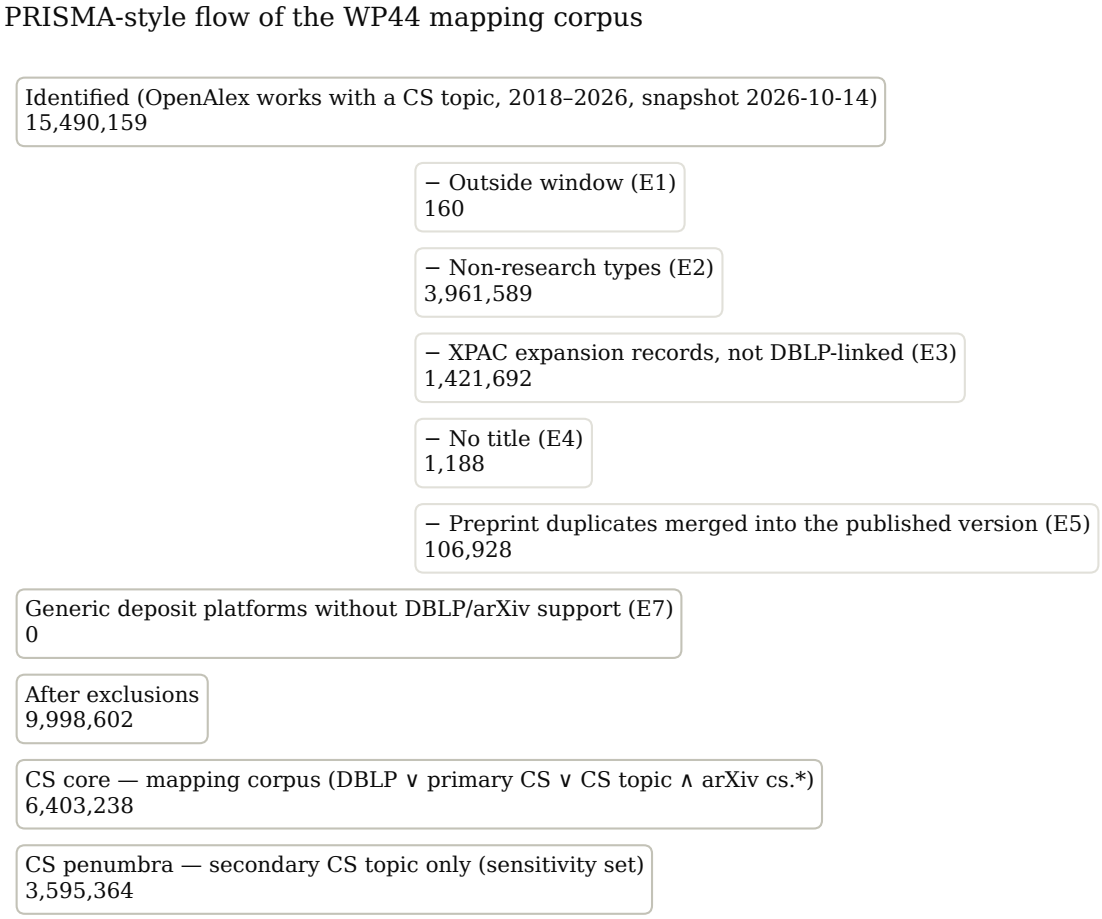


Figure 1. PRISMA-style flow of record identification, exclusion and the final mapping corpus. Identification counts are per source and overlap; exclusions are sequential in the order E1–E7; the core and the penumbra partition the records that survive the exclusions. Source: `data/metadata/prisma_counts.json`, rendered by `figures/make_figures.py`.

snapshot passes; a computer-science work that OpenAlex does not index at all, or indexes without any computer-science topic and without a DBLP or arXiv DOI, is not observed. The DBLP-only table and the unmatched ACL entries bound the size of this blind spot for two sources with complete venue lists, and Section 13 reports them.

12. Dates, date precision, indexing lag and right truncation

Monthly series are the primary temporal unit of this study, and the November 2022 boundary falls inside a year. Everything in the structural-change analysis of Part IX therefore depends on the quality of dates at the month level, which is considerably worse than at the year level in every bibliographic database. This section documents three distinct problems, their diagnosis in the corpus, and the rule adopted for each.

12.1. Which date

Each eligible work carries up to three dates: the OpenAlex publication date of the surviving record, the publication date of its merged preprint record where one exists, and the arXiv first-submission date where the work is linked to an e-print. The *first date* of a work is the earliest of the three (Section 9), and the *first month* is its calendar month. The first date measures the entry of the work into the public record, which is the quantity relevant to questions about the pace and composition of research activity; the publication date of the version of record measures the pace of the publishing system and is kept for the analyses of venues and of preprint-to-publication lags. The two coincide for works without a preprint, which are the majority outside the machine-learning areas.

12.2. Date precision and the January artefact

A large share of publication dates in OpenAlex are year-only dates stored as the first of January. The phenomenon is visible without any model: the raw monthly series of first appearances shows a spike in every January that no publishing calendar could explain (Figure 2), and inspection of the eligible corpus by year found that between about a fifth and somewhat over a quarter of records carry a date of the form *YYYY*-01-01, with the proportion varying by source and type; dissertations, book chapters and records from repositories are the most affected, journal articles with publisher-registered issue dates the least. Amendment A-002 introduced the flag `date_precision`, equal to `year` when the publication date is the first of January and `month` otherwise. The rule is a heuristic with a known cost: a work genuinely published on 1 January is classed as year-precision, and since the rule is applied to the publication date rather than to the first date, a year-precision work with an arXiv first date is correctly dated by its e-print. The counts of eligible works by precision class

will be reported by the macros [pending human review: PrismaEligibleDatePrecisionMonth] and [pending human review: PrismaEligibleDatePrecisionYear] once the corpus has been rebuilt under A-002.

All monthly series in Part IX use month-precision records only. Year-precision records enter the annual series, where their date is correct, and the composition analyses that do not depend on the month. The exclusion of year-precision records from monthly series is not innocuous: if the share of year-only dates differs between categories or between periods, the monthly composition is biased relative to the annual one. Part IX therefore reports, for each category, the share of year-precision records by year, and treats a divergence between the monthly and the annual growth estimates as a warning rather than as two independent results.

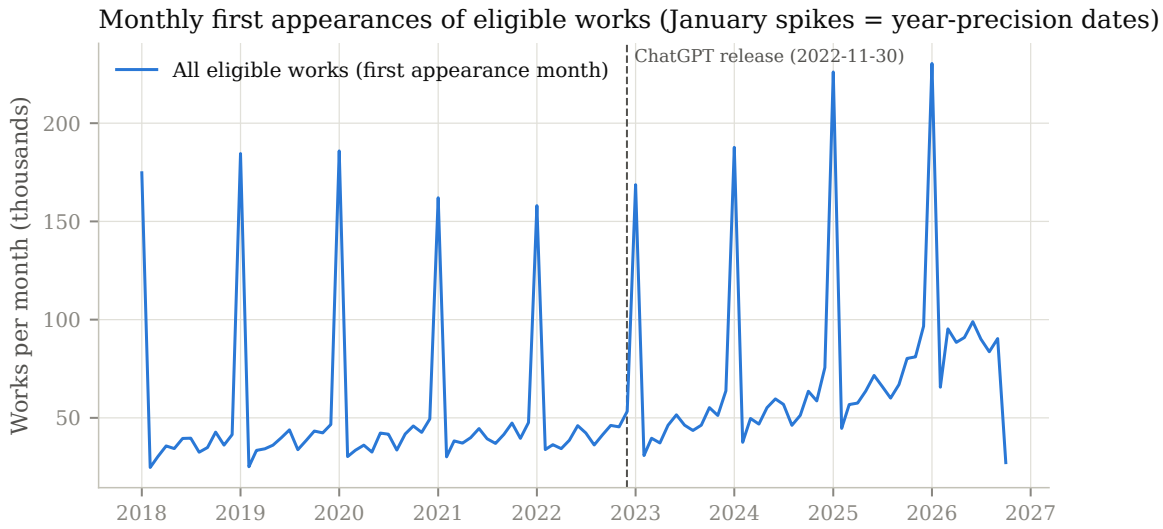


Figure 2. Monthly first appearances in the eligible corpus before the precision filter. The January spikes reflect year-only publication dates stored as 1 January and motivate the month-precision rule. Source: `results/bibliometrics/volume_monthly.csv`.

12.3. Indexing lag and why `created_date` cannot measure it

A work is observable only once the database has indexed it, and the delay between publication and indexing determines how incomplete the most recent months are. The research plan proposed to estimate the delay profile from the OpenAlex `created_date` field, the date on which the record was created in OpenAlex, and to use the profile to correct the right tail of the monthly series. The diagnostic run of this estimator (`results/bibliometrics/indexing_lag.csv`) refuted the proposal. For works published in 2018, the median difference between `created_date` and the publication date is close to eight years, and the median falls by almost exactly one month for each later publication month, which is the signature of a constant creation date: the field records the

date on which the record was created in the current OpenAlex data model, apparently during a re-ingestion in the second half of 2025, not the date on which the work was first indexed. For the great majority of works published before that re-ingestion, `created_date` carries no information about indexing delay, and for works published after it the field measures the delay only for the records created since. A lag profile estimated from it would be an artefact of database maintenance. The field is therefore not used, and the plan's lag-corrected estimates are replaced by the approach below. The episode is recorded in detail because a reader who reproduces the estimator on a later snapshot may find the field meaningful again, and because it shows why the mandate's requirement to preserve immutable raw snapshots is not a formality: the only way to measure indexing lag honestly is to compare two snapshots of the same database.

12.4. Right truncation

The last months of the window are incomplete for three distinct reasons that must not be conflated: works published but not yet indexed (indexing lag), works indexed but not yet linked to a DBLP record because DBLP's own curation lags the publishers, and works whose citation counts have had little time to accumulate. The first two shrink the counts of the recent months; the third affects only impact indicators and is treated in Part IV through fixed citation windows, which are the standard remedy for the age dependence of citation counts (Wang, 2013; Waltman, 2016).

For the counts, the diagnosis uses the sources against one another. The DBLP dump of 2026-11-02 holds, for 2026, a count well below its 2025 count although ten months of the year have elapsed, which shows that DBLP's own coverage of 2026 is incomplete; and the eligible corpus for 2026 is of the same order as that for 2025 (977,914 against 973,274), which on its face suggests that OpenAlex is *ahead* rather than behind. The two observations are reconciled by the composition of the 2026 records: a much lower share with a reference list, a much lower mean number of references, a higher share of preprints and, before amendment A-002, the Zenodo contribution described above. In other words, the recent months are not simply truncated; they are differently composed, with the fast channels (preprint servers, repositories) over-represented relative to the slow ones (journals, curated indexes). A correction factor estimated from totals would be wrong for every category whose channel mix differs from the average.

The rule adopted is conservative. The "complete" observation window for monthly inference ends in June 2026; the months from July to November 2026 are reported descriptively, separately, and never enter a breakpoint test. Within the complete window, the structural-change models include the last months with their observed values and the robustness section of Part IX re-estimates every break with the window ending in December 2025, so that a result that depends on the 2026 months is identified as such. The API increment harvested after the snapshot (Section 6) adds the works registered between the snapshot date and the extraction date, which narrows but does not close the

gap. Finally, forward-dated records, works carrying a 2026 publication date for an issue not yet released, are retained when their date precedes the extraction date because they are, by the definition of the first date, in the public record; the current build does not count them separately, and a count by month of the records whose publication date lies after the snapshot date is listed among the quality indicators to be added before Part IX is finalised.

13. Quality indicators and known biases

PRISMA 2020 asks for an assessment of the risk of bias in the included studies; a mapping study has no studies in that sense, and the protocol replaces the item with indicators of the quality of the bibliographic record itself, reported per year and per category in Appendix B. This section defines the indicators, reports the corpus-level values that are available at the time of writing, and sets out the known biases of the design, each with the direction in which it is expected to push the results of Part IX.

13.1. Metadata completeness

Four completeness indicators are computed for every work: presence of an abstract, presence of a non-empty reference list, presence of at least one affiliation with an institution identifier, and presence of a country for at least one author. In the eligible corpus, 4,867,840 works have an abstract and 3,715,037 have at least one resolved reference, out of 6,403,238. Both shares are lower than those reported for the DOI-bearing journal literature in comparisons of OpenAlex with the commercial indexes (Culbert et al., 2025; Delgado-Quirós and Ortega, 2024), which is expected in a corpus that includes dissertations, book chapters and preprints, and both fall sharply for the most recent year (Figure 3). The consequences are concrete. A work without an abstract cannot be embedded or classified by its content and is classified from its venue, its arXiv categories and its OpenAlex topics alone, with a lower confidence score; a work without references does not contribute out-citations to the knowledge graph, so that the citation flows of Part VIII are estimated on the subset with references and are weighted accordingly. The ACL Anthology abstracts fill part of the first gap for one subfield; no source fills the second.

13.2. Coverage overlap between sources

The overlap between sources is the study's main instrument for measuring its own coverage, because each source has a complete list of what it holds. Three overlaps are reported. The DBLP–OpenAlex overlap, after both passes, is the share of DBLP records in the window with an OpenAlex counterpart; its complement, the DBLP-only table, is tabulated by year and type in Appendix B. The arXiv–OpenAlex overlap is the share of `cs` e-prints with an OpenAlex counterpart (744,680 of 890,852

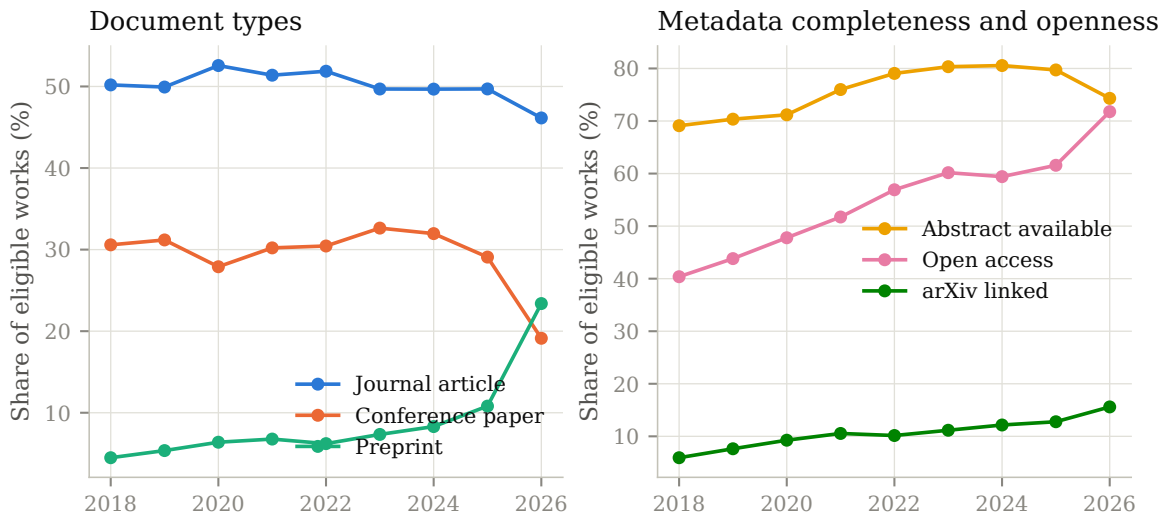


Figure 3. Document-type composition and metadata completeness of the eligible corpus by publication year (2026 incomplete). Source: `results/bibliometrics/volume_yearly.csv`.

e-prints first submitted in the window). The ACL–OpenAlex overlap is the share of Anthology papers linked (72,223 of 85,625). An unmatched ACL paper is a documented computer-science publication that the backbone either does not hold or holds under a title or identifier the linkage could not resolve; the residual is therefore an upper bound on the sum of backbone non-coverage and linkage failure for one conference-heavy subfield, and a manual review of a sample of unmatched entries, planned with checkpoint H-1, separates the two.

Figure 4 shows the eligible works by publication year and by source membership. The widening of the arXiv-linked layer and the relative stability of the DBLP-linked layer are the first descriptive facts of the study, and they are also a warning about right truncation: in the most recent years, the fastest source dominates.

13.3. Known biases and their expected direction

DBLP coverage bias. DBLP indexes established venues and adopts new conference series with a delay; within the core, the DBLP-linked subset over-represents long-standing journals and conferences and under-represents workshops, regional venues and the newest series. Because the strict core is defined through DBLP, every strict-core estimate inherits this bias, which runs against emerging topics and against the most recent years. Results that are stronger in the strict core than in the eligible corpus are therefore unlikely to be artefacts of over-inclusion; results that appear only in the eligible corpus may be.

OpenAlex topic noise. The topic classifier mislabels in both directions, and the membership rule bounds only the over-inclusion, through the penumbra. Under-inclusion is bounded by DBLP and

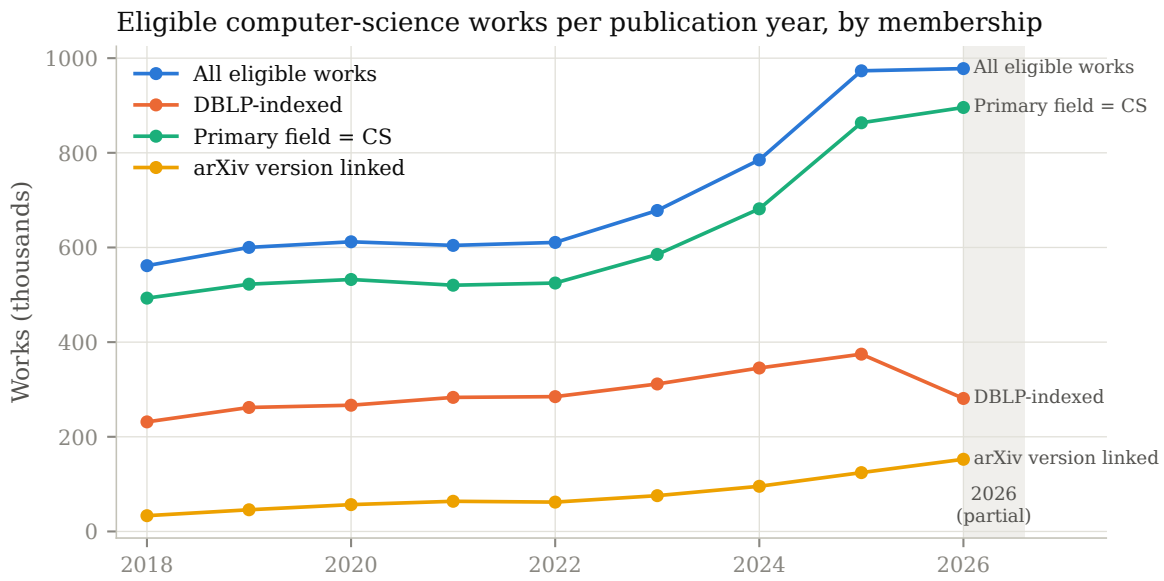


Figure 4. Eligible works per publication year by source membership (DBLP-linked, OpenAlex computer-science primary field, arXiv-linked, ACL Anthology); memberships overlap. 2026 is incomplete. Source: `results/bibliometrics/volume_yearly.csv`.

arXiv for the venues and preprints they cover and is unbounded elsewhere. Within the core, the topic labels are used as one of several signals in the taxonomy, never alone, and the taxonomy's validation on human-reviewed samples (Part IV, checkpoint H-2) is the only direct measure of how much label noise survives.

XPAC records. The expansion records excluded under E3 are not a random sample of the literature: they are concentrated in repositories, in non-English regions and in recent years. Their exclusion removes some genuine research, mostly outside the venue system, and biases the corpus toward the formal literature; the DBLP exception restores the part that the venue system vouches for. The sensitivity analysis that reinstates XPAC records of research types reports the difference.

Generic-deposit platforms. The exclusion under E7 removes, besides the deposits that motivated it, any research output whose only home is a generic repository; the effect is largest in 2025–2026 and in the areas, such as scientific computing and computational biology, where data and software deposits accompany papers. The direction of the bias is toward under-counting recent output in those areas, which is the opposite of the bias the rule corrects.

Missing abstracts. Abstract availability depends on publishers' metadata deposits and on OpenAlex's sources; it is lower for conference papers of some societies and for older records, and it is strongly correlated with the DOI registrant. Every content-based analysis is conditional on abstract availability, and the appendix reports the abstract share by category and year so that the reader can judge where the conditional population departs from the corpus.

Forward-dated and year-dated records. Both displace works along the time axis in a direction that is not symmetric: year-only dates pull works toward January, forward dates push them toward the future. The month-precision rule removes the first; the second is bounded by the window end and will be counted explicitly.

Retractions. Retracted works are in the corpus and out of the impact analyses. The retraction flag depends on the coverage of the Retraction Watch data through Crossref, which is incomplete for publishers that do not deposit retraction notices; some retracted works are therefore in the impact analyses unflagged. The direction of the bias is small but known: retraction is concentrated in a few venues and in a few years, and the landmark selection of Part V checks each candidate against the current retraction status at the time of review.

Preprint duplication. The residual double count after the exact-title pairing inflates the areas that preprint most, which are also the areas expected to show the strongest growth. The sensitivity with a similarity threshold and the removal of preprints with a journal DOI bound the effect; a growth result for machine-learning areas that does not survive these two variants is reported as not robust.

None of these biases is corrected by reweighting. The study reports them, bounds them where a second source permits, and re-estimates its results on alternative corpora (strict core; corpus with penumbra; corpus with XPAC research types; corpus with DBLP-only records; corpus with generic deposits) so that the reader can see how far each conclusion depends on each scope decision. This is the approach that the science-of-science literature recommends for databases whose coverage is known to be uneven (Fortunato et al., 2018a), and it is the only one compatible with a mandate that forbids the fabrication of corrections.

14. Research-integrity procedures

The mandate for this working paper lists integrity rules that are unusual for a bibliometric study: no fabricated reference, identifier, count or result; no paper content invented from metadata; no abstract treated as a full text; no model-generated summary treated as evidence; no temporal association read as causation; no preprint described as peer-reviewed; inaccessible data recorded explicitly; and human review of every scientific conclusion. The rules are a response to the way the study is conducted, by a principal investigator directing autonomous software agents, and this section describes the procedures by which they are enforced. The procedures are reported here, in the data part, because most of them operate on data and artefacts rather than on prose.

14.1. Preregistration and the amendments log

The protocol (`protocol/PROTOCOL.md`) was frozen on 2026-11-02 before any analysis of the corpus. It fixes the research questions, eight directional hypotheses H1–H8 with the sign of the

expected effect stated so that null and opposite results must be reported as such, the eligibility criteria and their codes, the sources and extraction dates, the data items, the quality indicators, the synthesis methods, the human-review checkpoints and the data-availability commitments. The preregistration is internal, in the repository rather than in a registry, which limits its evidential value relative to a time-stamped public record (Nosek et al., 2018); the repository's version history and the dated amendments log are the available substitute, and the release of Phase 9 will include both. Every departure from the frozen protocol is an entry in `protocol/amendments.csv` with an identifier, a date, the phase, the change, the reason and the approver, which until the principal investigator signs is "executing agent (pending PI)". At the time of writing the log holds five entries: A-000 records that the mandate's acceptance criteria were completed by the executing agent after the pasted text was truncated; A-001 and A-002 are the eligibility amendments of Section 10; A-003 and A-004 fix the budgets and the models of the LLM-based taxonomy validation and knowledge-graph extraction. Analyses added after the freeze are labelled exploratory in Part IX.

14.2. Immutable raw snapshots and provenance

Nothing in the corpus can be traced to a result unless the raw data from which it was built can be re-obtained or re-verified. The OpenAlex release is a public, versioned object; the study keeps its manifest, the per-file content lengths and SHA-256 hashes, and the filtered subset, rather than a copy of the full bucket (decision D6), so that a reader with the same release and the same scripts obtains byte-identical inputs, and a reader with a later release can measure the difference. The DBLP dump is kept with its MD5, the arXiv harvest as the raw OAI-PMH pages, the ACL export with its SHA-256, and the API increment as the raw JSON pages with a ledger of calls and cost. Each ingestion job writes a ledger with attempts, timings, hashes, counts and errors, which is the job-level provenance the mandate requires. The processing chain from raw snapshot to PRISMA counts is a sequence of scripts with measured wall times in `reproducibility/RUNBOOK.md`, executed in a locked Python environment whose package versions are recorded; the chain was executed on a single node, deliberately, because it is network-bound and because a single-node execution is easier to reproduce than a distributed one.

14.3. Numbers that cannot be typed

Every count, share and estimate quoted in this manuscript is a LaTeX macro generated by `tables/make_numbers.py` from the result files (`prisma_counts.json`, `linkage_report.json`, the bibliometric and econometric result tables). The macros cannot be edited by hand, a macro whose source file is absent is undefined and stops the compilation, and a number the authors need but the pipeline has not produced appears in the text as a visible pending marker rather than as a

figure. The procedure is the mechanical form of the mandate's requirement that every quantitative claim trace to a script, an input artefact and a result file. It has a cost in prose, since the text cannot anticipate a number, and a benefit in auditability, since a reader who regenerates the result files regenerates the manuscript's numbers.

14.4. Verified references

The bibliography of this manuscript is built under a rule that no reference may be cited unless it has been resolved to a bibliographic record by a single programmatic lookup against Crossref, OpenAlex or arXiv, with the returned title compared with the intended one and the comparison recorded in a verification table (`references_verification*.csv`: key, identifier, service, returned title, match). References that fail the lookup are removed from the bibliography and kept in the table with a failed status, and a second agent re-verifies a sample in the audit phase. The rule exists because the model-based agents that draft the prose are known to produce plausible but non-existent references when unconstrained; it is also why preprints are cited as preprints, with their arXiv identifier and without a venue, and why the bibliography distinguishes a technical report from a peer-reviewed article in its entry type.

14.5. Agents, artefacts and the register

The study is executed by a research-director agent that spawns specialised agents with written briefs; each brief, the artefacts it produced and the agent that verified them are recorded in `agents/REGISTER.md`. The rule that governs the register is that no agent approves its own findings: an artefact enters the manuscript only after a different agent, the scientific critic or the reproducibility auditor, has checked it, and the register records the checker. Agents exchange structured artefacts (tables, JSON, Markdown with declared schema) rather than conversational summaries, so that what one agent passes to another can be inspected after the fact. The writing agents, including the one that produced this part, work from the source-of-truth files named in their briefs and are instructed to describe what the code does rather than what the plan intended; where the two differ, as with the indexing-lag estimator of Section 12, the difference is reported.

14.6. Human-review checkpoints

Six checkpoints are defined (H-0 to H-5: plan and protocol; linkage sample; taxonomy validation sample; landmark list and specialist reviews; structural-change results and their interpretation; manuscript conclusions), each with a named artefact that the principal investigator reviews and signs (`reports/HUMAN_REVIEW_CHECKPOINTS.md`). Until a checkpoint is signed, every conclusion that depends on it is labelled "pending human review" in the text, with the same visible marker as

a missing number. At the time of writing no checkpoint is signed. The reader should take this literally: the conclusions of this manuscript are, in its present version, the conclusions of a procedure, reviewed by its own audit agents and awaiting the judgement of its human author.

14.7. What the procedures do not guarantee

The procedures guarantee traceability, not truth. A count traced to a script is still wrong if the script is wrong; a verified reference is still miscited if the prose misreports what it says; a preregistered hypothesis can still be tested with an inappropriate model. What the procedures make possible is detection: every number, reference and decision has a place where an auditor can look. The remaining risks are the ones this part has tried to state plainly: the coverage and labelling errors of the sources, the residual duplication, the fragility of the recent months, and the dependence of every scope decision on the judgement, recorded but not infallible, of the agents that took it.

Part IV

Computational Methodology

15. Notation and the corpus as a measurable object

Every quantity reported in Parts VIII and IX is a function of a finite set of bibliographic records and of a small number of deterministic transformations applied to them. This part states those transformations with enough precision that an independent analyst could recompute the results from the raw snapshots listed in Part III. The exposition follows the order of the pipeline (`reproducibility/RUNBOOK.md`): record linkage, deduplication and eligibility (Section 16), multi-label classification (Section 17), bibliometric indicators (Section 18), network measures (Section 19), semantic mapping (Section 20), structural change (Section 21), emerging-topic detection (Section 22), knowledge-graph extraction (Section 23) and the treatment of uncertainty and reproducibility (Section 24). Throughout, a design choice that departs from the preregistered plan (`RESEARCH_PLAN.md`, `protocol/PROTOCOL.md`) is stated as such, because the code, not the plan, is what produced the numbers.

15.1. Works, dates and periods

Let $\mathscr{W}_0$ denote the set of OpenAlex work records loaded from the snapshot of 14 October 2026 (the records carrying at least one topic of the Computer Science field, plus the second-pass records recovered through DBLP and arXiv identifiers, Section 16). Each record $w \in \mathscr{W}_0$ carries an

OpenAlex identifier, a type $\tau(w)$ drawn from the OpenAlex controlled vocabulary, a publication date $d_{\mathrm{pub}}(w)$, a creation date $d_{\mathrm{cr}}(w)$ (the date at which the record entered OpenAlex), a title, an optional abstract, authorships, a primary source (venue), OpenAlex topics with their subfield and field, a list of referenced works, and yearly citation counts. Linked records from DBLP, arXiv and the ACL Anthology attach to w a DBLP key, an arXiv identifier with its arXiv creation date $d_{\mathrm{arx}}(w)$ and primary category, and an ACL identifier.

The temporal position of a work is its *first date*,

$$d(w) \;=\; \min\bigl\{\, d_{\mathrm{pub}}(w),\; d_{\mathrm{pub}}(\pi(w)),\; d_{\mathrm{arx}}(w) \,\bigr\}, \tag{3}$$

where $\pi(w)$ is the preprint record merged into w by the deduplication rule of Section 16.2 when one exists, and each missing term is replaced by $d_{\mathrm{pub}}(w)$. The first date, not the date of the version of record, is the quantity that places a work on the time axis, because the research question concerns when ideas entered the literature rather than when a publisher released them. The publication year $y(w)$ used for citation normalisation is, in contrast, the OpenAlex `publication_year` of the surviving record, because citation windows are defined in calendar years of the OpenAlex `counts_by_year` table (Section 18.2). The two conventions are deliberately distinct and both are stored on every record.

The observation window is $[t_0, t_1] = [\text{2018-01-01}, \text{2026-11-02}]$, split at the reference date $t^* =$ 2022-11-30 into a baseline period $B = [t_0, t^*)$ and a post period $P = [t^*, t_1]$:

$$\mathrm{period}(w) \;=\; \begin{cases} B & \text{if } t_0 \le d(w) < t^*, \\ P & \text{if } t^* \le d(w) \le t_1. \end{cases} \tag{4}$$

Monthly series use the month $m(w)$ of the first date. Two facts about dates constrain everything that follows. First, a substantial share of OpenAlex records carry a publication date equal to the first of January of their year, which is the database's convention for year-only precision rather than a real date. The pipeline therefore attaches to every record a flag

$$\mathrm{prec}(w) \;=\; \begin{cases} \text{year} & \text{if } d_{\mathrm{pub}}(w) \text{ falls on 1 January}, \\ \text{month} & \text{otherwise}, \end{cases} \tag{5}$$

and every monthly series in this paper is computed on the month-precision subset $\mathscr{W}^{\mathrm{m}} = \{w : \mathrm{prec}(w) = \text{month}\}$ only (amendment A-002). A genuine 1 January publication is thereby misclassified as year precision and dropped from monthly series; the loss is small and, more importantly, it does not depend on the period, so it cannot manufacture a break. Annual series use all eligible records. Second, the right end of the window is censored: records published in the last months

before the snapshot are under-represented because OpenAlex indexes them with a lag. The treatment of this right-truncation is specified in Section 21.1.

15.2. Categories, areas and the measurable structure

Let $\mathscr{K}$ be the set of the forty leaf categories of the ontology (`taxonomy/ontology.yaml`, Part III and Appendix A) and $\mathscr{A}$ the set of its eight areas, with the surjection $a : \mathscr{K} \to \mathscr{A}$ assigning each leaf to its area. Seven leaves are additionally declared children of another leaf (for instance *llm* $\prec$ *foundation_models*, *compilers* $\prec$ *programming_languages*); this partial order $\prec$ is used only to break ties when a primary category is chosen. The classifier of Section 17 assigns to each work a label set $C(w) \subseteq \mathscr{K}$, possibly empty, and, when $C(w) \neq \emptyset$, a primary category $\kappa^*(w) \in C(w)$. Multi-label counting and primary counting give different answers to "how many works are in category k", and both are reported: for a set of works S,

$$N_k^{\mathrm{ml}}(S) = \big|\{w \in S : k \in C(w)\}\big|, \qquad N_k^{\mathrm{pr}}(S) = \big|\{w \in S : \kappa^*(w) = k\}\big|, \tag{6}$$

so that $\sum_k N_k^{\mathrm{pr}}(S) \leq |S|$ with equality when every work is classified, whereas $\sum_k N_k^{\mathrm{ml}}(S)$ exceeds $|S|$ by the number of secondary labels. Flows between categories (Section 19) and the emerging-topic panel (Section 22) use the primary label; co-classification statistics use the full label set.

The eligible corpus $\mathscr{W} \subseteq \mathscr{W}_0$ is the set of records passing the predicate $E(w)$ of Section 16.3. For a month t and a category k, the elementary counts are

$$N_t = \big|\{w \in \mathscr{W}^{\mathrm{m}} : m(w) = t\}\big|, \qquad N_{k,t} = \big|\{w \in \mathscr{W}^{\mathrm{m}} : m(w) = t,\ \kappa^*(w) = k\}\big|, \qquad s_{k,t} = N_{k,t}/N_t. \tag{7}$$

Every estimand of Parts VIII and IX is a functional of these counts, of the citation matrix of Section 18.2, of the authorship lists, or of the embedding vectors of Section 20. The corpus is finite and fully enumerated, so the descriptive quantities are population quantities rather than sample estimates; the inferential devices of Sections 21 and 22 address a different kind of uncertainty, namely whether an observed temporal pattern is distinguishable from the ordinary fluctuations of the series, and whether a predictor would have worked out of time.

15.3. Storage and computation

The pipeline keeps the raw snapshots immutable (per-file SHA-256 ledgers, Section 24) and materialises all derived tables as Parquet files queried through DuckDB (Raasveldt and Mühleisen, 2019); the linkage, deduplication, eligibility, counting and citation-window steps are SQL statements executed inside one database file, which makes every table in this part a deterministic function of the raw inputs and of the script that produced it. Python is used where SQL is inconvenient: the

regular-expression classifier, the structural-change toolkit, the embedding and clustering steps, and the detector. The slim table of all OpenAlex works (464,765,485 rows: identifier, DOI, year, type, primary topic hierarchy, citation count) is kept so that the field of a cited work outside computer science can be resolved (Section 19.1); this is decision D2 of the plan.

16. Record linkage, deduplication and eligibility

The four sources describe overlapping but differently delimited populations: OpenAlex is the backbone, DBLP defines what counts as a computer-science venue, arXiv contributes preprints and their categories, and the ACL Anthology serves as a coverage check for natural-language processing. Combining them is a record linkage problem in the classical sense of Fellegi and Sunter (1969): two records from different files must be declared the same work or not, on the basis of comparison vectors computed from their fields. The pipeline uses a deterministic cascade rather than a probabilistic model, because the dominant comparison fields (DOI and arXiv identifier) are persistent identifiers whose agreement is near-conclusive, and because a probabilistic model would require labelled training pairs that the project does not possess before the human checkpoint H-1. The cost of this choice is that the precision of the residual title-based step must be estimated from a reviewed sample rather than derived from the model; the design of that estimate is given in Section 16.4.

16.1. Normalised keys and the linkage cascade

For each source the script `ingestion/link_sources.py` builds a key table. A DOI is lower-cased. An arXiv identifier is lower-cased and stripped of its version suffix (`v[0-9]+$`); for DBLP records, which do not carry an arXiv field, the identifier is extracted from the electronic-edition URLs by the pattern `arxiv.org/abs/([^ ]+)`. A title is normalised by the map

$$\eta(\text{title}) \;=\; \text{delete}_{\neg[a\text{–}z0\text{–}9]}\big(\text{lower}(\text{title})\big), \tag{8}$$

which lower-cases and deletes every character outside $[a\text{–}z0\text{–}9]$, so that punctuation, whitespace, case and diacritics (which do not survive the ASCII filter) no longer separate otherwise identical titles. The normalised title is used only for records whose original title has at least 25 characters, a threshold that removes the many short generic titles (“Introduction”, “Editorial”, “Preface”) for which equality of η carries no evidence.

DBLP records are restricted to the years 2018–2026 and to the record types *article*, *inproceedings*, *incollection*, *book*, *phdthesis* and *data*. The DBLP–OpenAlex cascade is then

1. **DOI**: join on equal lower-cased DOI; every matching pair is kept (a DBLP key may match

several OpenAlex records sharing a DOI, and conversely; the many-to-many structure is resolved at the deduplication stage, not here);

2. **arXiv**: among records unmatched on both sides, join on equal normalised arXiv identifier;
3. **title–year**: among records still unmatched on both sides, restrict each side to normalised titles that are *unique* within that side, join on equal η, and keep the pair if the years differ by at most one.

The uniqueness restriction in step 3 is the blocking device: it turns the join into a one-to-one map and prevents a frequent title from generating a clique of spurious pairs. The price is recall, since a genuine pair whose title appears twice on either side (a journal version and a workshop version with the same title, for instance) cannot be linked at this step; such pairs are usually caught by the DOI step or by the preprint merge of Section 16.2. The arXiv–OpenAlex cascade is analogous (arXiv identifier, then DOI, then unique normalised title with a year tolerance of two, because the arXiv creation year and the OpenAlex publication year of the version of record can legitimately differ by more than one). The ACL–OpenAlex cascade uses DOI, then the ACL identifier parsed from OpenAlex landing-page URLs of the form `aclanthology.org/<id>`, then unique normalised title with a year tolerance of one. Each link carries its method label, which is the stratification variable of the precision sample.

The counts of the first run are stored in `data/metadata/linkage_report.json` and reproduced in Part III; here it suffices to note the shape of the result. Of the 4,347,365 DBLP records in window, 4,052,326 carry a DOI, 2,562,431 are linked by DOI and 179,990 by title and year, for 2,736,903 distinct DBLP keys linked and 1,610,462 unlinked. No DBLP record was linked through the arXiv step (0 DBLP keys with a parsable arXiv identifier), a result consistent with DBLP recording arXiv e-prints through their DataCite DOIs (prefix 10.48550), which the DOI step captures, rather than through `arxiv.org/abs` URLs; the cause is to be confirmed in the audit of Part III. The arXiv–OpenAlex cascade links 744,680 of the 890,852 arXiv `cs` records created in window, almost all of them (792,672) by identifier. The 1,077,407 unlinked DBLP records whose DOI does exist in the slim table of all OpenAlex works are precisely the records that OpenAlex does not label with any Computer Science topic; they motivate the second snapshot pass, in which these identifiers (1,077,406 from DBLP and 39,200 from arXiv) are fetched in full and the linkage is rerun with the union of both passes. The corpus is therefore not the set of works that OpenAlex calls computer science; it is the union of that set with what DBLP and arXiv call computer science, which is the sense in which the mandate's requirement to "cover the complete breadth of computer science" rather than only the OpenAlex topic model is operationalised.

16.2. Preprint–publication deduplication

A work that exists as an arXiv preprint and as a published version is two OpenAlex records whenever OpenAlex has not merged them. Counting both would double-count exactly the kind of work (AI preprints) whose growth is at issue (CLAUDE.md §6.2). The script `systematic_review/define_corpus.py` identifies such pairs within $\mathscr{W}_0$ by a title rule:

$$(p,b) \text{ is a candidate pair} \iff \eta(p)=\eta(b),\ |y(p)-y(b)|\le 2,\ p\in\mathscr{P},\ b\notin\mathscr{P},\ |\text{title}|\ge 25, \quad (9)$$

where $\mathscr{P}=\{w:\tau(w)=\text{preprint or DOI}(w)\text{ starts with 10.48550/arxiv}\}$ is the preprint-like set. A preprint may match several published records and conversely; the rule keeps, for each published record, the earliest-dated preprint and, for each preprint, the earliest-dated published record, and discards any pair that is not the first in both orderings. The surviving pairs define the map π of equation (3): the published record b receives $\pi(b)=p$, preprint_date$(b)=d_{\text{pub}}(p)$ and the earlier of the two dates as its first date; the preprint p receives the flag *e5_duplicate* and leaves the eligible corpus. The version relationship is therefore kept on the surviving node, as the plan requires, but the match is by exact normalised title only. Two limitations follow. A preprint whose title changed between versions is not merged, and the pair survives as two works; a preprint and an unrelated published paper with an identical normalised title within two years are wrongly merged. The second error is rare for titles of at least 25 characters; the first is the more consequential one and biases the preprint share upward, which is the conservative direction for hypothesis H7. Deduplication across the OpenAlex–DBLP boundary is not needed because DBLP records are never counted as works: a DBLP record either links to an OpenAlex record, in which case the OpenAlex record carries the flag *in_dblp*, or it does not, in which case it is counted in the separate DBLP-only table used for coverage assessment.

16.3. Eligibility predicate

Each record receives seven exclusion flags and two membership flags, computed once and stored, so that every PRISMA count is a filter on the same table. With T_{keep} = {article, conference-paper, preprint, book-chapter, book, dissertation, report, review, software-paper}, T_{hard} = {editorial, erratum, retraction, peer-review, paratext, conference-abstract, reference-entry, dataset, software, supplementary-materials, libguides, letter, grant, standard}, and $\mathscr{G}$ the list of generic deposit

platforms (Zenodo, Figshare, IRDB, LA Referencia, Dialnet, DOAJ-as-source, OSTI), the flags are

$$e_1(w) : d_{\text{pub}}(w) \text{ missing or outside } [t_0, t_1]; \tag{10}$$

$$e_2(w) : \tau(w) \notin T_{\text{keep}} \ \wedge \ \neg\big(\text{in_dblp}(w) \wedge \tau(w) \notin T_{\text{hard}}\big); \tag{11}$$

$$e_3(w) : \text{is_xpac}(w) \wedge \neg\text{in_dblp}(w); \tag{12}$$

$$e_4(w) : \text{title empty}; \tag{13}$$

$$e_5(w) : w \text{ is the preprint member of a pair of (9)}; \tag{14}$$

$$e_6(w) : \text{is_retracted}(w); \tag{15}$$

$$e_7(w) : \text{source}(w) \in \mathscr{G} \wedge \neg\text{in_dblp}(w) \wedge \text{no linked arXiv record}. \tag{16}$$

The membership flags encode the computer-science rule of the protocol:

$$\text{core}(w) = \text{in_dblp}(w) \ \vee \ \text{cs_primary}(w) \ \vee \ \big(\text{cs_any}(w) \wedge \text{arxiv_cs_primary}(w)\big), \tag{17}$$

$$\text{penumbra}(w) = \text{cs_any}(w) \wedge \neg\,\text{core}(w), \tag{18}$$

where cs_primary means that the primary OpenAlex topic belongs to field 17, cs_any that some topic does, and arxiv_cs_primary that the linked arXiv record's primary category starts with `cs.`. The eligibility predicate is

$$E(w) \ = \ \neg e_1 \wedge \neg e_2 \wedge \neg e_3 \wedge \neg e_4 \wedge \neg e_5 \wedge \neg e_7 \wedge \text{core}(w). \tag{19}$$

Three features of this predicate deserve comment. Retraction (e_6) is *not* part of E: retracted works stay in the mapping corpus with their flag and are excluded only from impact-based selections (Section 23), as the protocol specifies (E6). The type rule e_2 is asymmetric by amendment A-001: a DBLP-indexed record is kept whatever its OpenAlex type unless that type is in the hard-exclusion list, because the first run showed that OpenAlex typed more than a million DBLP conference papers in ways the original list rejected; XPAC records (OpenAlex's expansion of coverage from secondary sources) are likewise kept when DBLP-indexed and flagged *xpac_kept_dblp*. The deposit-platform rule e_7 was added by amendment A-002 after Zenodo alone was found to contribute a six-figure number of 2026 records typed as articles with almost no references; the protocol's original E7 (penumbra) is now the membership flag rather than an exclusion, so the reason code E7 in the protocol and the flag e_7 in the code denote different things. This is a documentation discrepancy, not an analytical one, but readers comparing the protocol and the PRISMA table should be aware of it. The penumbra set—works with a secondary computer-science topic but neither DBLP nor arXiv support—is counted separately and enters only sensitivity analyses; the strict core used for the sensitivity analyses in the other direction is $\{w : E(w) \wedge \text{in_dblp}(w) \wedge \text{cs_primary}(w)\}$.

The PRISMA counts are computed both marginally (how many records carry each flag) and sequentially (how many are removed by each flag after the previous ones have been applied, in the order $e_1, e_2, e_3, e_4, e_5, e_7$), because the flags overlap heavily and only the sequential version sums to the final count. The order is a reporting convention; the final eligible set does not depend on it.

16.4. Design of the linkage-precision estimate

Precision of the deterministic cascade is estimated on a stratified sample of 500 DBLP–OpenAlex pairs drawn by `systematic_review/linkage_sample.py`: 250 pairs linked by DOI, 200 by title–year and 50 by arXiv identifier (the last stratum is empty in the current run, for the reason given in Section 16.1), selected by ordering each stratum on a hash of the DBLP key salted with the seed 44 and taking the first records, which is a reproducible simple random sample within stratum. For each pair the script computes an automatic agreement vector: a string similarity between the two raw titles (Python's `difflib` ratio, twice the number of matched characters in the longest-matching-block alignment divided by the total length), the absolute year difference, and whether the last tokens of the two first-author names coincide; a pair is *auto_ok* if the similarity is at least 0.9, the years differ by at most one and the surnames agree whenever both are present. Two of every five pairs are reserved for the principal investigator (*pi_review*), whose verdicts constitute the human checkpoint H-1. Let n_s be the stratum sizes and $\hat{p}_s$ the share of pairs judged correct in stratum s; the overall precision of the link table is the weighted estimate

$$\hat{p} = \sum_s \frac{L_s}{L}\,\hat{p}_s, \qquad \widehat{\mathrm{Var}}(\hat{p}) = \sum_s \left(\frac{L_s}{L}\right)^2 \frac{\hat{p}_s(1-\hat{p}_s)}{n_s}, \tag{20}$$

where L_s is the number of links of method s in the full table and L their sum, so that the title–year stratum, which is the only one where errors are plausible, is weighted by its small share of the links. The estimate [pending human review: linkage precision from the H-1 sample, data/metadata/linkage_sample_summary.json] is reported in Part III once the checkpoint is signed; until then the automatic agreement rate is the only available figure and is labelled as such. Recall is not estimated by this design: a DBLP record without an OpenAlex counterpart may be a true absence from OpenAlex or a missed link, and distinguishing the two would require searching OpenAlex by title for a sample of unlinked DBLP records, which is left to the robustness appendix.

17. Multi-label classification into the ontology

The ontology assigns works to forty categories by rules over four kinds of metadata signals and one textual signal. A rule-based classifier was preferred to a trained one for three reasons that are worth stating because they are also its limitations. The rules are inspectable: every assignment can

be traced to the signals that produced it, and the mapping from OpenAlex topics, arXiv categories and DBLP venues to categories is a document (`taxonomy/ontology.yaml`, with provenance notes per category) that a reader can contest line by line. The rules need no labelled training set, which the project did not have before checkpoint H-2, and so cannot inherit the biases of one. And the rules are cheap enough to run on every eligible work, whereas an LLM rater at the quality used for validation would cost orders of magnitude more. The corresponding limitations are that rules do not generalise beyond the vocabulary they encode, that the weights are expert judgements rather than estimates, and that validation against independent raters is therefore the only evidence of their accuracy.

17.1. Signals and the score

For a work w, let $T(w)$ be its set of OpenAlex topic identifiers (up to three per work), $F(w)$ its set of subfield identifiers, $X(w)$ the set of arXiv categories of its linked arXiv record (primary and cross-lists), $D(w)$ the set of DBLP venue streams of its linked DBLP records, and $\mathrm{title}(w)$, $\mathrm{abs}(w)$ its title and abstract. Each category k specifies weight maps $\omega_k^{\mathrm{T}} : T \to [0,1]$, ω_k^{F}, ω_k^{X}, ω_k^{D} over the respective vocabularies (zero outside the listed entries), a list of keyword rules (r, ω, ϕ) consisting of a regular expression r, a weight ω and a field $\phi \in \{\mathrm{title}, \mathrm{abstract}, \mathrm{both}\}$, and a list of exclusion regular expressions Z_k. The score of w for k is

$$s_k(w) \;=\; \sum_{\sigma \in \{\mathrm{T,F,X,D}\}} \min\Big\{1,\; \sum_{x \in \sigma(w)} \omega_k^{\sigma}(x)\Big\} \;+\; \min\Big\{1,\; \sum_{(r,\omega,\phi) \in R_k} \omega \cdot \mathbf{1}\big[r \text{ matches } \phi(w)\big]\Big\}, \quad (21)$$

where $\phi(w)$ is the title, the abstract, or their concatenation according to the field of the rule. Each of the five signal types is thus capped at one before the types are summed, so $s_k(w) \in [0,5]$, and the assignment rule is

$$k \in C(w) \iff s_k(w) \geq \theta \;\wedge\; \neg \exists z \in Z_k : \; z \text{ matches } \mathrm{title}(w) \| \mathrm{abs}(w), \qquad \theta = 1. \quad (22)$$

The cap is what gives the weights their meaning. A single signal with weight one (a topic that *is* the category, a venue that publishes nothing else, a keyword that names the object of study) suffices on its own; a signal with weight below one is a partial indication that must be corroborated by a signal of another type, because repetitions of the same type cannot push the capped sum above one. The ontology uses this device deliberately: the six AI categories that the OpenAlex topic model does not separate (large language models, foundation models, multimodal AI, generative AI, agents, AI safety) rely on keyword rules of weight one that fire on the name of the object, while their topic and venue signals are fractional. The exclusion list is a veto applied after the threshold: a category that would otherwise be assigned is removed if any exclusion pattern matches the title or abstract (for

instance a cryptocurrency pattern vetoing applied cryptography). The veto is not a negative weight; it cannot be outvoted.

One discrepancy between the ontology document and the implementation must be recorded. The notes of the `assignment_rule` block state that the classifier takes the *maximum* over matched topics before capping, whereas `taxonomy/classify.py` takes the *sum* and then caps, as equation (21) states. The two agree whenever at most one listed topic matches or the matched weights already sum to at least one; they differ when two fractional topics of the same category co-occur, in which case the implemented rule is the more permissive. The unit tests in `tests/test_classify_rules.py` fix the implemented behaviour (a topic listed three times contributes its weight once, because the topic set is deduplicated, and a 0.6 topic plus a 0.4 arXiv category reach exactly 1.0). The manuscript describes the implemented rule.

17.2. Primary category

When $C(w) \neq \emptyset$ the primary category is

$$\kappa^*(w) \;=\; \operatorname*{arg\,max}_{k \in C(w)} \big(s_k(w),\; -\mathbf{1}[k \text{ is a parent of some } k' \in C(w)],\; -\text{number}(k) \big) \tag{23}$$

in lexicographic order: the highest score wins; among equal scores a category that is the declared parent of another assigned category loses to its child (so that a work scoring equally on *llm* and *foundation_models* is primarily an LLM paper); among remaining ties the lower ontology number wins. The last rule is arbitrary and is stated so that the reader can see that it is. The number of assigned categories $|C(w)|$ and the primary score are stored with every work; works with $C(w) = \emptyset$ (eligible works none of whose signals reach the threshold) are counted and characterised in Part IX, and their share is one of the quality indicators of the classification. The classifier is run on the eligible corpus and on the penumbra, so that sensitivity analyses on the wider set do not require a second run.

17.3. Validation design and metrics

The rules are validated against two independent LLM raters on a stratified sample (`taxonomy/VALIDATION.md`, `taxonomy/validate_sample.py`): fifty eligible works with an abstract per rule-assigned primary category (forty strata) plus two hundred works with no category, drawn by ordering each stratum on a salted hash with seed 44; two hundred of the resulting works are reserved for the principal investigator (checkpoint H-2). Each rater receives the forty category definitions, and for each work the title, abstract, venue and year, and returns under a JSON schema a label set of at most five categories, a primary category or null, a confidence and a short rationale. The raters never see the

rule-based assignment, so there is no leakage from the rules to the reference labels, and the rule revision policy (a category with consensus $F_1 < 0.60$ triggers a mapping revision, logged as an amendment, and a re-run) can be applied without re-rating. The two raters are different models of the same family (`claude-haiku-5-5`, `claude-sonnet-5-5`); their agreement therefore does not measure the agreement that two human experts would reach, and the 200-work human subsample is the only anchor for that.

With $\hat{C}(w)$ the rule label set, $C^r(w)$ the label set of rater r and $C^{\cap}(w) = C^A(w) \cap C^B(w)$ the consensus set, the per-category confusion counts against a reference C^{ref} are $\text{TP}_k = |\{w : k \in \hat{C}(w) \wedge k \in C^{\text{ref}}(w)\}|$, $\text{FP}_k = |\{w : k \in \hat{C}(w) \wedge k \notin C^{\text{ref}}(w)\}|$, $\text{FN}_k = |\{w : k \notin \hat{C}(w) \wedge k \in C^{\text{ref}}(w)\}|$, with precision $P_k = \text{TP}_k/(\text{TP}_k + \text{FP}_k)$, recall $R_k = \text{TP}_k/(\text{TP}_k + \text{FN}_k)$ and $F_{1,k} = 2P_kR_k/(P_k + R_k)$, the standard label-wise decomposition of multi-label evaluation (Zhang and Zhou, 2014). Micro-averages pool the counts over categories before forming the ratios; the macro-F_1 is the unweighted mean of $F_{1,k}$ over categories with at least one positive or predicted label. Because the sample is stratified by rule-assigned primary category, the per-category precision is estimated on roughly equal support for every category, whereas recall is estimated only among works that the rules placed somewhere; a category that the rules systematically miss (true members scattered among other strata or among the unassigned) will show inflated recall in this design. The 200 unassigned works partly address this, but a proper recall estimate would require sampling from the whole corpus with weights, and the per-category recall figures should be read with that caveat.

Agreement on the primary category is measured by Cohen's κ (Cohen, 1960),

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \qquad p_o = \frac{1}{n}\sum_w \mathbf{1}[\kappa^A(w) = \kappa^B(w)], \qquad p_e = \sum_{k \in \mathcal{K} \cup \{\varnothing\}} \frac{n_k^A}{n}\frac{n_k^B}{n}, \tag{24}$$

where "no category" is treated as a forty-first label and n_k^r counts the works to which rater r assigned primary k; κ is computed between the two raters and between the rules and each rater. Agreement on the label sets is measured by the mean Jaccard index $\bar{J} = n^{-1}\sum_w |C^A(w) \cap C^B(w)|/|C^A(w) \cup C^B(w)|$ (with the convention $0/0 = 0$). The validation note announces Krippendorff's α with the MASI distance for the multi-label agreement; the metrics script computes the mean Jaccard index instead, and the manuscript reports what is computed. A work is exported as a disagreement for human review when the rules, rater A and rater B have no category in common although at least one of them assigned one.

17.4. Known failure modes

The design makes some errors predictable, and they are listed here so that the results of the validation can be read against expectations rather than as surprises. Keyword rules on abstracts fire on mentions

as well as on contributions: a systems paper that evaluates its scheduler on a large language model workload matches the LLM keyword and, since that keyword alone reaches the threshold, acquires the *llm* label; the exclusion lists remove the most frequent such cases but cannot enumerate them. Venue signals are coarse: general machine-learning venues carry fractional weights for several AI categories at once, so a work with no abstract and a generic title can be assigned to a parent-level category on venue evidence alone, or to none. OpenAlex topics are themselves the output of a classifier with documented error, and a work whose topics are wrong inherits that error with weight; the protocol lists this among the limitations declared before analysis. Multi-label assignment with a cap of one per type and a threshold of one tends to over-assign categories to long abstracts that mention many objects, which inflates $|C(w)|$ in the post period, when abstracts increasingly cite models and benchmarks by name; the co-classification statistics of Section 19 are affected by this in a direction that would mimic convergence (hypothesis H4) and must be read together with the primary-label flow statistics, which are not. Finally, works without an abstract (a known share of the corpus) are classified from the title and the metadata only and have lower recall for the keyword-dependent categories; the share of such works is reported by year because it is not constant.

18. Bibliometric indicators

The indicators of this section are the estimands of research questions RQ1, RQ3 and RQ6. They are standard in the scientometric literature, and the definitions follow the review of Waltman (2016) where a choice had to be made; what is specific to this study is the citation-window discipline imposed by the eight-year span of the corpus and the explicit treatment of the two counting conventions. The scripts are `bibliometrics/01_volume_series.py`, `02_collaboration.py` and `03_citation_impact.py`.

18.1. Volume, composition and growth

The monthly count N_t and the category counts $N_{k,t}$ of equation (7) are tabulated by first month, period, date precision, type and source membership (DBLP-indexed, OpenAlex-primary, arXiv-linked), with the numbers of works having an abstract, having at least one resolvable reference, and being open access; the annual table adds the mean number of authors, references and countries per work. Composition is the share $s_{k,t}$; its annual counterpart uses the primary label on all eligible works. Growth between two dates separated by Δ years is summarised by the compound annual growth rate

$$\mathrm{CAGR}(t_a,t_b) \;=\; \left(\frac{N_{t_b}}{N_{t_a}}\right)^{1/\Delta} - 1, \qquad \Delta = (t_b - t_a) \text{ in years}, \tag{25}$$

applied to annual totals and to annual category totals; it is a descriptive summary of two end points and says nothing about the path between them, which is why the structural-change analysis of Section 21 works on the full monthly series instead. The CAGR is computed at table-making time from the annual counts rather than by a dedicated script; the plan lists it among the Phase-4 indicators and it is reported for complete years only, since 2026 is right-censored (Section 21.1).

18.2. Fixed citation windows

OpenAlex provides, for each work, the number of citations received in each calendar year (`counts_by_year`). Let $c_w(u)$ be the citations received by w in calendar year u and $y(w)$ its publication year. The fixed-window counts are

$$c_w^{(12)} = \sum_{u \le y(w)+1} c_w(u), \qquad c_w^{(36)} = \sum_{u \le y(w)+2} c_w(u), \qquad c_w^{(48)} = \sum_{u \le y(w)+3} c_w(u), \tag{26}$$

so that $c^{(12)}$ counts citations in the publication year and the following calendar year, $c^{(36)}$ in the publication year and the two following years, and so on. The superscripts are the names used in the code and in the tables and must be read as calendar-year windows rather than as exact month counts: a work published in January and one published in December of the same year receive windows that differ by eleven months in effective length. This granularity is imposed by the data (OpenAlex does not provide citation dates below the year) and is the same for the baseline and the post period, so it blurs comparisons within a year but does not bias comparisons between years. The window is the device that neutralises citation-age bias: comparing the total citation count of a 2019 work with that of a 2025 work compares seven years of exposure with one, whereas comparing $c^{(36)}$ compares equal exposures. The cost is that the last window-length of the corpus cannot be evaluated; $c^{(36)}$ is complete for publication years up to 2023 given a snapshot taken late in 2026, and $c^{(12)}$ up to 2024. Tables flag the years for which a window is incomplete, and the choice of window length follows the evidence in Wang (2013) that short windows rank works materially differently from long ones in slow-moving fields, which is why two lengths are reported rather than one.

18.3. Field normalisation

Raw citation counts differ across subfields and document types by factors that have nothing to do with influence. The normalised citation score of a work is its window count divided by the mean window count of its reference class (Waltman, 2016; Waltman and van Eck, 2019):

$$\mathrm{NCS}_w^{(h)} = \frac{c_w^{(h)}}{\bar{c}^{(h)}\big(y(w), \tau(w), f(w)\big)}, \qquad \bar{c}^{(h)}(y,\tau,f) = \frac{1}{|\mathscr{W}_{y,\tau,f}|} \sum_{v \in \mathscr{W}_{y,\tau,f}} c_v^{(h)}, \tag{27}$$

where $f(w)$ is the primary OpenAlex subfield of w and $\mathscr{W}_{y,\tau,f}$ the set of eligible works sharing year, type and subfield. The reference class is computed on the eligible corpus, not on all of OpenAlex, so the normalisation is relative to computer science as delimited here: an NCS of 2 means twice the mean of comparable computer-science works, not twice the world mean. The score is undefined when the class mean is zero (very small classes in the most recent years) and is then left missing rather than set to zero. Normalisation by the mean of a heavy-tailed distribution is sensitive to a few very highly cited works in the class; the median-based alternatives discussed by Waltman (2016) are not used because the median of $c^{(36)}$ is zero in many classes of this corpus, which would make them uninformative.

OpenAlex's own field-weighted citation impact (FWCI) is kept as a secondary indicator. Its documentation defines it as "citations received divided by citations expected for works of the same type, year, and subfield", with a world average of one (OpenAlex, 2026). It differs from equation (27) in three respects: the expected count is computed over all of OpenAlex rather than over this corpus; the citation count is the total to date rather than a fixed window, so that FWCI values of recent works are based on a short and unequal exposure; and the computation is OpenAlex's, recomputed nightly, so that the value depends on the snapshot date. For these reasons FWCI is reported for comparison and never used to select works.

18.4. Inequality and concentration

Citation inequality within a publication year is measured by the Gini coefficient of $c^{(36)}$ among the n eligible works of that year. With the works sorted so that $c_{(1)} \le c_{(2)} \le \cdots \le c_{(n)}$, the estimator implemented in SQL is the rank form

$$G = \frac{2\sum_{i=1}^{n} i\, c_{(i)}}{n\sum_{i=1}^{n} c_{(i)}} - \frac{n+1}{n}, \tag{28}$$

which equals one minus twice the area under the Lorenz curve and lies in $[0, 1-1/n]$; the coefficient is named after Ceriani and Verme (2012)'s account of its origin. Uncited works (a large share within a three-year window) contribute zero to the numerator and pull G upward, so the share of uncited works is reported next to it. The protocol (H5) refers to the Gini of 24-month citations; the implemented statistic uses the three-calendar-year window $c^{(36)}$, which is complete for one more publication year than the longer alternatives and less dominated by zeros than $c^{(12)}$. This is a deviation from the protocol wording and is recorded as such.

Venue concentration within a year is the Herfindahl–Hirschman index of the distribution of works over sources,

$$\mathrm{HHI}_y = \sum_{v} \sigma_{v,y}^2, \qquad \sigma_{v,y} = \frac{N_{v,y}}{\sum_{v'} N_{v',y}}, \tag{29}$$

where $N_{v,y}$ is the number of eligible works of year y whose primary source is v (Hirschman, 1964); works without a source are excluded from the denominator, and the number of distinct sources and the largest single share are reported with the index. Because arXiv is a single OpenAlex source, growth in preprints mechanically raises HHI_y even when the journal and conference landscape becomes more dispersed; this is the reason the protocol's H5 predicts a stable or falling HHI only "because of preprint growth" with the preprint source treated separately, and the tables report the index with and without the preprint type.

18.5. Collaboration, countries and institutions

Team size is summarised by the mean and median number of authors per work and by the shares of single-author works and of works with at least five or ten authors. Two counting conventions are used for countries and institutions, because they answer different questions (Waltman, 2016). Under full counting a work contributes one unit to every country among its authors' affiliations; under fractional counting it contributes $1/|K(w)|$ to each of its $|K(w)|$ distinct countries, so that the fractional totals sum to the number of works with at least one country. The same pair of conventions is applied to institutions, with the fraction taken over the distinct institutions of the work. The international collaboration share of year y is

$$\mathrm{ICS}_y \;=\; \frac{|\{w : y(w) = y,\ |K(w)| \geq 2\}|}{|\{w : y(w) = y,\ |K(w)| \geq 1\}|}, \tag{30}$$

that is, among works with at least one resolved country, the share with authors from two or more countries; works without affiliation data are excluded from the denominator and their share is reported, because affiliation completeness in OpenAlex varies by source and year and a change in completeness would otherwise masquerade as a change in collaboration. The industry share is the share of works with at least one affiliation of OpenAlex institution type *company* among works with at least one typed affiliation. Country-pair and institution-level tables apply minimum-support thresholds (fifty joint works for a country pair, two hundred works for an institution within a period) that keep the released tables to the robustly estimated part of the distribution; these thresholds are reporting choices and are stated in the table notes.

18.6. Diversity indices

Interdisciplinarity is measured on the distribution of a work's references, or of a category's citations, over the categories of the ontology. Let $p = (p_1, \ldots, p_K)$ be such a distribution and $\delta_{ij} \in [0,1]$ a

dissimilarity between categories i and j. The Rao–Stirling index (Rao, 1982; Stirling, 2007)

$$\Delta_{\mathrm{RS}}(p) \;=\; \sum_{i \neq j} p_i \, p_j \, \delta_{ij} \tag{31}$$

is the expected dissimilarity between two categories drawn at random from p; it is zero when all mass sits on one category and increases with variety (the number of categories present), balance (the evenness of p) and disparity (the dissimilarity of the categories involved), the three components that Stirling argues any diversity measure should respond to. Its use in scientometrics follows Rafols and Meyer (2010). Its known weakness is that it saturates: for very even distributions over many categories it approaches the mean dissimilarity and no longer discriminates (Zhang et al., 2016). The Leinster–Cobbold family (Leinster and Cobbold, 2012) repairs this by defining, for a similarity matrix $Z = (z_{ij})$ with $z_{ij} = 1 - \delta_{ij}$ and an order $q \geq 0$,

$$ {}^{q}D^{Z}(p) \;=\; \Big(\sum_{i : p_i > 0} p_i \, (Zp)_i^{\,q-1} \Big)^{1/(1-q)}, \qquad (Zp)_i = \sum_j z_{ij} \, p_j, \tag{32}$$

with the limit ${}^{1}D^{Z}(p) = \prod_i (Zp)_i^{-p_i}$ at $q = 1$. The quantity $(Zp)_i$ is the "ordinariness" of category i under p: how much of the distribution is similar to i. The index is an effective number of categories—it equals K for the uniform distribution over K mutually dissimilar categories—and the order q sets the weight given to rare categories (large q counts only the dominant ones). Following Zhang et al. (2016) the manuscript reports $q = 2$, which has the closed form ${}^{2}D^{Z}(p) = 1/\sum_{i,j} p_i z_{ij} p_j = 1/(1 - \Delta_{\mathrm{RS}}(p))$ when $z_{ii} = 1$, so that the two families are monotone transforms of each other at this order and differ only in scale and in their behaviour at the saturation end. The dissimilarity δ_{ij} between categories is one minus the cosine similarity of the category centroids of Section 20 in the baseline period, so that the disparity component is estimated from the corpus rather than imposed. The current pipeline computes the category flow matrices from which p is formed (Section 19) and the centroid similarities, but the two diversity indices are not yet produced by a script; they are defined here because the protocol commits to them, and their values in Part IX are [pending human review: Rao–Stirling and Leinster–Cobbold indices per category and period, not yet computed] until the computation is added and audited.

19. Citation and collaboration networks

19.1. The citation graph and its boundary

Citation edges come from the OpenAlex `referenced_works` lists, which are verified by construction in the sense that each edge is an assertion of the source database rather than an inference of

this study. The script `knowledge_graph/01_citation_graph.py` splits the references of eligible works into two edge sets:

$$\mathscr{E}_{\text{int}} = \{(w,v) : w \in \mathscr{W},\ v \in \mathscr{W},\ v \in \text{refs}(w)\}, \tag{33}$$

$$\mathscr{E}_{\text{ext}} = \{(w,v) : w \in \mathscr{W},\ v \notin \mathscr{W},\ v \in \text{refs}(w),\ v \in \text{slim table}\}, \tag{34}$$

the internal edges forming the citation graph of computer science proper and the external edges recording what the corpus cites outside itself, with the cited work's publication year, type and primary OpenAlex field resolved from the slim table of all works. References that resolve to no OpenAlex record are counted (the declared reference count of each work is stored) but cannot be typed. The yearly tables derived from these sets are the inflows by cited field (how much of each citing year's references go to mathematics, medicine, the social sciences, and so on), the distribution of citation age $y(w) - y(v)$ by citing year, and the reference coverage (declared, internal, external) by year. Coverage matters for every network statistic that follows: if the share of resolvable references rises over time, as it does in OpenAlex because recent works are better matched, then internal edge counts rise for reasons unrelated to citation behaviour, and all flow statistics are therefore expressed as shares of resolved edges within the citing year.

19.2. Category flow matrices

For each citing year t the script `knowledge_graph/02_category_flows.py` forms the $K \times K$ matrix

$$F_t[i,j] \;=\; \big|\{(w,v) \in \mathscr{E}_{\text{int}} : y(w) = t,\ \kappa^*(w) = i,\ \kappa^*(v) = j\}\big|, \tag{35}$$

the number of citations made in year t by works whose primary category is i to works whose primary category is j, together with the row-normalised shares $F_t[i,j] / \sum_{j'} F_t[i,j']$. Works without a primary category contribute no edges to F_t; their references are retained in $\mathscr{E}_{\text{int}}$ but fall outside the matrix. Aggregating rows and columns by area through $a(\cdot)$ gives the 8×8 area flow matrix, and the cross-area share of citing area α in year t is

$$X_{\alpha,t} \;=\; \frac{\sum_{i:a(i)=\alpha}\ \sum_{j:a(j)\neq\alpha} F_t[i,j]}{\sum_{i:a(i)=\alpha}\ \sum_{j} F_t[i,j]}, \tag{36}$$

the fraction of an area's internal references that leave the area. This is the statistic behind hypothesis H4. It is not a measure of interdisciplinarity in the sense of Section 18.6, because it treats all other areas as equally distant; it is a boundary-crossing rate, and its interpretation depends on the stability of the boundaries, which the classifier fixes once for all years. A co-classification matrix counts, for each year, the works carrying both category i and category j in their full label set $C(w)$; it is reported

alongside the flows but, as noted in Section 17, it is sensitive to abstract length and vocabulary in a way the primary-label flows are not.

19.3. Modularity of the yearly flow network

The degree to which citations stay within categories, or within areas, is summarised by the modularity of the weighted directed graph with adjacency $W = F_t$, evaluated at the fixed partition given by the categories themselves (each category its own community) and at the coarser partition given by the areas. The implemented statistic is

$$Q(W, g) = \frac{1}{m} \sum_{i,j} \left(W_{ij} - \frac{k_i^{\text{out}} k_j^{\text{in}}}{m} \right) \mathbf{1}[g_i = g_j], \qquad m = \sum_{i,j} W_{ij}, \quad k_i^{\text{out}} = \sum_j W_{ij}, \quad k_j^{\text{in}} = \sum_i W_{ij}, \tag{37}$$

which is the directed generalisation by Leicht and Newman (2008) of the Newman–Girvan modularity (Newman and Girvan, 2004): the observed weight within communities minus the weight expected if each citation were rewired at random while preserving every node's out-strength and in-strength. Because the partition is given rather than optimised, Q is used here as a descriptive index of assortativity by category, not as the objective of a community detection algorithm; the resolution limit that affects optimised modularity (Fortunato and Barthélemy, 2007; Fortunato, 2010) therefore does not arise, but the index inherits the dependence of the flow matrix on the classifier. Two values are reported per year, Q_{cat} with $g_i = i$ and Q_{area} with $g_i = a(i)$, together with the self-share $\text{tr}(F_t)/m$, which is the uncorrected fraction of citations staying within the citing category. Hypothesis H4 predicts a decrease in Q_{area} in the post period. A fall in Q can also arise if the composition of citing works shifts toward categories that were always boundary-crossing (large language models citing natural-language processing and machine-learning theory), so Part IX reports Q together with the composition-adjusted cross-area shares obtained by fixing the row weights at their baseline values.

19.4. Co-authorship, institution and country networks

The plan lists author, institution and country collaboration networks built with `igraph`. The implemented collaboration layer is tabular rather than graph-analytic: country pairs with at least fifty joint works per period (edges of the country network, with their weights), institution counts by period under full and fractional counting, and institution-type shares. The country network is thus available as an edge list from which the figures of Part VIII are drawn, and degree, strength and density can be computed from it; author-level co-authorship networks are not constructed in the current pipeline, because author disambiguation in OpenAlex is of uneven quality over the period and an author graph of this size would be dominated by that noise. This is a narrowing of the plan,

recorded here; the country and institution tables are the collaboration estimands actually reported.

20. Semantic mapping

20.1. Document embeddings

Each eligible work with an abstract is represented by a dense vector computed by a scientific sentence-embedding model from the string "title. abstract[1:2000]" (the abstract truncated to 2,000 characters), tokenised to at most 256 word pieces, and L2-normalised, so that the inner product of two vectors is their cosine similarity (`embeddings/embed_corpus.py`). The primary model is SPECTER2 base (Singh et al., 2023), the successor of SPECTER (Cohan et al., 2020), a transformer trained with a citation-informed triplet objective so that papers that cite one another are close in the embedding space; this inductive bias is the appropriate one for mapping a literature by its intellectual proximity rather than by lexical overlap. The protocol requires a second embedding family for comparison; the benchmark script evaluates SciNCL (Ostendorff et al., 2022), a general-purpose BGE model and a MiniLM sentence-transformer (Reimers and Gurevych, 2019) for throughput and dimensionality on the same hardware, and the atlas script accepts any of them through its model argument, so the comparison is a re-run with a different tag. The embeddings are stored as float16 shards with their work identifiers and a per-host manifest (model name, maximum sequence length, device, timings), computed on several nodes of the cluster in parallel; float16 storage changes cosine similarities by less than 10^{-3}, which is below any threshold used downstream.

Three assumptions underlie everything built on these vectors. The abstract is assumed to be a faithful summary of the work's content, which it is for most research articles and is not for extended abstracts and some proceedings papers. The embedding geometry is assumed stable across the period: a model trained on literature up to its release date represents 2026 work in a space shaped by earlier work, so genuinely new vocabulary is embedded by analogy to what the model knows, and the novelty statistic below measures distance in that space, not in some ideal space of ideas. And cosine similarity is assumed to be a sufficient proxy for topical proximity; this is the standard assumption of the field and is supported by the retrieval benchmarks on which these models are evaluated, with the caveat that all such benchmarks are dominated by pre-2023 literature.

20.2. Atlas: projection and clustering

The two-dimensional atlas and the cluster structure are computed on a uniform random sample of 200,000 eligible works with embeddings (seed 44), because density-based clustering of the full corpus in the native space is neither computationally practical nor, given the noise level of abstracts, more informative. UMAP (McInnes et al., 2018) with the cosine metric, 30 nearest neighbours and a

minimum distance of zero projects the sample to 50 dimensions; HDBSCAN (Campello et al., 2013; McInnes et al., 2017) with a minimum cluster size of 200 and a minimum sample count of 20 is run on the 50-dimensional projection; a second UMAP to two dimensions (minimum distance 0.1) is used for display only. The 50-dimensional intermediate step is what makes density-based clustering workable on 768-dimensional normalised vectors, whose pairwise distances concentrate; it is also a source of distortion, since UMAP preserves local neighbourhoods and not global distances, so cluster *membership* is meaningful and the distances between clusters on the two-dimensional map are not. HDBSCAN labels points in low-density regions as noise (-1), and the noise share is reported with the cluster count. Each cluster is described by the twelve title terms with the highest mean TF–IDF weight (unigrams and bigrams, English stop words removed, minimum document frequency five), by its modal primary category and that category's share, and by the share of its members in the post period; these descriptors are how clusters are related to the ontology, and the comparison of the two partitions (bottom-up clusters against top-down categories) is one of the checks on the ontology reported in Part IX.

20.3. Stability

Cluster stability is measured by a split-half design. Five times, the sample is randomly permuted and divided into two halves; HDBSCAN (minimum cluster size 100, since each half has half the points) is run separately on the two halves of the *same* 50-dimensional projection; a 5-nearest-neighbour classifier trained on the first half's labels predicts labels for the second half; and the adjusted Rand index (Hubert and Arabie, 1985)

$$\mathrm{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}} \tag{38}$$

between the transferred labels and the second half's own labels is recorded (here n_{ij} is the contingency table of the two labellings and a_i, b_j its margins). The mean of the five values is the reported stability. Two caveats bound what this measures. The projection is fitted once on the full sample, so the design tests the stability of the clustering conditional on the projection and not the stability of UMAP itself, whose stochastic optimisation is fixed by the seed; the protocol's phrase "bootstrap ARI" describes this split-half procedure, which resamples without replacement. And the nearest-neighbour transfer is itself a smoothing step: it maps noise points of the first half to the majority label of their neighbours, which raises the index relative to a direct comparison of two independent clusterings of the same points.

20.4. Category centroids, drift and novelty

In the native embedding space, the centroid of category k in year y is the normalised mean vector

$$\mu_{k,y} \;=\; \frac{\sum_{w:\kappa^*(w)=k,\; y(w)=y} e_w}{\left\|\sum_{w:\kappa^*(w)=k,\; y(w)=y} e_w\right\|}, \tag{39}$$

computed only for cells with at least 100 works, and the year-over-year drift of a category is the cosine distance $1-\mu_{k,y}^{\top}\mu_{k,y-1}$. Period centroids average the yearly centroids over calendar years 2018–2022 (baseline) and 2023–2026 (post), and the inter-category cosine similarity matrices of the two periods are the semantic counterpart of hypotheses H4 and H5: convergence between two categories appears as an increase in the cosine similarity of their centroids. The period boundary used here is the calendar year, not the 30 November 2022 date of Section 15; the eleven months of 2022 that belong to the baseline are kept in the baseline and December 2022 is attached to it rather than to the post period, a deliberate simplification at the level of yearly centroids that moves a single month across the boundary.

The novelty of a work is its distance to the recent past of its own category:

$$\nu(w) \;=\; 1-e_w^{\top}\,\bar{\mu}_{\kappa^*(w),\; m(w)-24:m(w)-1}, \qquad \bar{\mu}_{k,a:b}=\frac{\sum_{t=a}^{b}\sum_{v:\kappa^*(v)=k,\; m(v)=t} e_v}{\left\|\cdot\right\|}, \tag{40}$$

the cosine distance between the work's vector and the normalised sum of the vectors of all works of the same primary category whose first month lies in the 24 months strictly before the work's own month. A work in the first months of a category, with no trailing window, has no novelty value. The definition uses only information dated before the work, which is what allows it to serve as a signal in the out-of-time detector of Section 22; but the centroid is computed with the *final* primary categories of the earlier works, assigned by rules fixed in 2026, so the category boundaries themselves are not out-of-time. This is a limitation shared by every category-conditional statistic in the paper and is discussed in Section 22. Novelty defined as distance to a category centroid rewards works that sit between categories as much as works that are new; the interpretation of high novelty as "new" rather than "interdisciplinary" requires the clusters and the flow statistics as corroboration, and the literature on novelty indicators warns that citation-based and text-based novelty measures do not agree well (Uzzi et al., 2013; Wang et al., 2017).

21. Structural change in monthly series

The question behind RQ2 and hypotheses H1 and H3 is whether the monthly series of computer-science output, in total and by category, changed level or slope around the reference date, and

whether any such change stands out against the ordinary variability of the series. Four methods are applied to every series by `econometrics/run_structural_change.py`, using the pure functions of `econometrics/structural_change.py` (unit-tested on a simulated series with a known break). None of them identifies a cause. An interrupted time series with a single pre-specified date answers "did the series change at that date, net of trend and season?"; the placebo distribution answers "is that date special?"; the multiple-break and changepoint methods answer "where does the series change, if one does not specify a date?"; and the Bayesian online detector answers "when would an observer following the series month by month have concluded that it had changed?". The reference date is a historical marker, and the manuscript uses no causal vocabulary about it (CLAUDE.md §6.5).

21.1. Series construction and truncation

Each series is a vector of monthly counts N_t on the month-precision subset, from February 2018 (the default start of the driver script, which drops one month and is immaterial) to the last complete month t_{L}, by default June 2026, with sensitivity runs at March and August 2026. Category series $N_{k,t}$ use the multi-label count restricted to month-precision works, and a series is analysed only if it has at least 48 non-zero months and at least 500 works in total; smaller series yield unstable fits and are reported descriptively. All methods operate on $y_t = \log(N_t + 1)$, the shift making the transform defined for empty months and immaterial for counts in the hundreds or thousands.

Right-truncation is handled by truncation, not by correction. The plan proposed to correct recent months by the indexing-delay profile estimated from the creation dates; the implemented pipeline estimates that profile (`indexing_lag.csv`: for each publication month, the median and 90th percentile of $d_{\mathrm{cr}}(w) - d_{\mathrm{pub}}(w)$ and the shares indexed within 90 and after 180 days) and reports it, but the structural-change runs simply stop the series at t_{L} and vary t_{L}. The reason is that an inflation correction based on the lag profile of earlier years assumes that the profile is stationary, and the profile is itself affected by the events under study (OpenAlex's own ingestion changed over the period); a correction would import that assumption into every test. Truncating to the last month at which the lag profile indicates near-complete indexing loses a few months of data but introduces no model. Donut windows (dropping the months adjacent to the event) are not implemented; the alternative event dates of Section 21.7 serve the related purpose of showing whether the estimated change is tied to a particular month.

21.2. Interrupted time series with seasonality and HAC errors

With $t = 0, 1, \dots, n-1$ indexing the months and t_e the index of the event month, the segmented regression is

$$y_t \;=\; \beta_0 + \beta_1 t + \beta_2 \,\mathbf{1}[t \ge t_e] + \beta_3 \,(t - t_e)\,\mathbf{1}[t \ge t_e] + \sum_{j=2}^{12} \gamma_j \,\mathbf{1}[\text{month}(t) = j] + \varepsilon_t, \tag{41}$$

the standard interrupted-time-series specification (Lopez Bernal et al., 2016b) with eleven month-of-year dummies (January as reference). The coefficient β_2 is the change in log level at the event, β_3 the change in monthly log slope, and β_1 the pre-event slope; since y is a log count, $e^{\beta_2} - 1$ is the proportional level shift and β_1 and β_3 are monthly growth rates. The event month is coded as the first full month after the reference date, December 2022 for the 30 November 2022 marker, and likewise for the alternative dates. Ordinary least squares estimates the coefficients; the covariance matrix is the heteroskedasticity- and autocorrelation-consistent estimator of Newey and West (1987) with a Bartlett kernel and bandwidth six months,

$$\hat{V} = (X^\top X)^{-1} \Big(\hat{\Gamma}_0 + \sum_{\ell=1}^{6} \big(1 - \tfrac{\ell}{7}\big)\big(\hat{\Gamma}_\ell + \hat{\Gamma}_\ell^\top\big) \Big) (X^\top X)^{-1}, \qquad \hat{\Gamma}_\ell = \sum_t \hat{\varepsilon}_t \hat{\varepsilon}_{t-\ell} x_t x_{t-\ell}^\top, \tag{42}$$

which makes the reported standard errors robust to serial correlation of the residuals up to half a year. The bandwidth is fixed rather than chosen by the data-dependent rule of Andrews (1991); with roughly one hundred observations the difference is immaterial and a fixed choice is easier to preregister. The fitted values are back-transformed as $e^{\hat{y}_t} - 1$ for the figures. The specification's known weaknesses are those of any single-break segmented regression: it assumes the change is immediate and permanent, it attributes to β_2 any curvature of the pre-trend that a linear $\beta_1 t$ cannot absorb, and with a reference date placed near the centre of the sample it is well identified only because the post period is long. The placebo analysis is the main guard against the second weakness.

21.3. Placebo distribution

The ITS of equation (41) is re-estimated with the event placed at every month t' from the 25th month of the series to the 12th month before its end, yielding the t-statistics $\{t^{(2)}_{t'}\}$ and $\{t^{(3)}_{t'}\}$ of the level and slope changes as a function of the hypothetical event date. The placebo p-value of the true event is its rank among these:

$$p^{\text{plac}}_{\beta_2} \;=\; \frac{1}{\lvert \mathscr{T}' \rvert} \sum_{t' \in \mathscr{T}'} \mathbf{1}\big[\, \lvert t^{(2)}_{t'} \rvert \ge \lvert t^{(2)}_{t_e} \rvert \,\big], \tag{43}$$

and likewise for β_3. The set $\mathscr{T}'$ contains the true event, so the smallest attainable value is $1/|\mathscr{T}'|$. This statistic is the "placebo in time" diagnostic recommended for interrupted time series (Lopez Bernal et al., 2018): if a level shift of comparable t-statistic is found at a quarter of all candidate dates, a conventional p-value from the HAC standard error of β_2 overstates the evidence, because the series has enough curvature and persistence that any date looks like a break. The value is not a p-value under a null hypothesis in the usual sense (the candidate dates are not exchangeable draws from a null distribution, and the regressions share the same data), and the manuscript calls it a placebo rank. Two implementation details depart from the plan's wording. The plan speaks of placebo breaks "at every month of the baseline"; the implementation ranges over the baseline *and* the post period, so that a change in 2023 or 2024 that is larger than the one at the reference date lowers the rank of the reference date, which is the conservative direction for hypothesis H1 and is why the wider range was kept. And the minimum of 24 pre-months and 12 post-months means that candidate dates near the ends of the series, where the level-change coefficient is poorly identified, are excluded from the comparison.

21.4. Multiple breaks: dynamic programming with information criteria

The Bai–Perron framework (Bai and Perron, 1998, 2003) estimates m unknown break dates $\tau_1 < \cdots < \tau_m$ as the global minimiser of the sum of squared residuals of a segment-wise regression. The implementation is the pure dynamic programme of Bai and Perron (2003) on the log series with a segment-wise linear trend: for a segment $[i, j)$ of at least h observations,

$$S(i,j) \;=\; \min_{a,b} \sum_{t=i}^{j-1} \big(y_t - a - b\,t\big)^2, \tag{44}$$

and the optimal partition of the first j observations into $m+1$ segments is obtained recursively,

$$C_m(j) \;=\; \min_{mh \le i \le j-h} \big\{ C_{m-1}(i) + S(i,j) \big\}, \qquad C_0(j) = S(0,j), \tag{45}$$

with the break positions recovered by back-tracking. The minimum segment length is $h = \max\{8, \lfloor 0.15\,n \rfloor\}$ months, a trimming of 15 per cent that is the usual choice in this literature; it forbids breaks in the first and last h months, which for the default window of 101 months ($h = 15$) means that no break can be placed after April 2025, and it also forbids two breaks closer than h months. The number of breaks, up to three, is selected by two criteria computed on the global

minima $\mathrm{SSR}_m = C_m(n)$,

$$\mathrm{BIC}(m) = n\log\big(\mathrm{SSR}_m/n\big) + p_m \log n, \tag{46}$$

$$\mathrm{LWZ}(m) = n\log\big(\mathrm{SSR}_m/(n-p_m)\big) + p_m\, c_0\, (\log n)^{2.1}, \qquad c_0 = 0.299, \tag{47}$$

with $p_m = 2(m+1)+m$ parameters (an intercept and a slope per segment plus the m break dates). The BIC is Schwarz's criterion (Schwarz, 1978); the LWZ criterion of Liu et al. (1997), in the form used by Bai and Perron (2003), penalises additional breaks more heavily and is known to select fewer spurious breaks when the errors are serially correlated, which they are in these series (the monthly residual autocorrelation is what the HAC errors of the ITS correct for). Both selections are reported. The sequential sup-F tests of Bai and Perron are not implemented; in their absence the break dates carry no p-value, and their stability is assessed instead through the agreement of the two criteria, the sensitivity to t_{L}, and the agreement with the two methods below. The regression in each segment contains no seasonal terms, so a seasonal pattern can only be absorbed by the error; with a trimming of eight or more months per segment this cannot create a break by itself but it inflates SSR_m uniformly.

21.5. PELT

The pruned exact linear time algorithm (Killick et al., 2012) finds the changepoint set $\mathscr{C}$ minimising a penalised cost,

$$\min_{\mathscr{C}} \sum_{\text{segments } [i,j) \text{ induced by } \mathscr{C}} \sum_{t=i}^{j-1} (y_t - \bar{y}_{i:j})^2 \;+\; \lambda\, |\mathscr{C}|, \qquad \lambda = 2\,\widehat{\mathrm{Var}}(y)\,\log n, \tag{48}$$

with the ℓ_2 cost (piecewise constant mean) and a minimum segment length of six months, as implemented in the `ruptures` library (Truong et al., 2020). The penalty is of the Schwarz type studied by Yao (1988) for the piecewise-constant Gaussian model, scaled by the sample variance of the series; it is a conservative choice in a trending series, because a piecewise-constant mean must spend changepoints to follow a trend, so the reported PELT dates should be read as the points where the level moves most, not as the Bai–Perron break dates of a trend model. The two methods are complementary for that reason: a date flagged by both is a change in level that is not explained by a change in slope, and a date flagged only by the trend model is a kink.

21.6. Bayesian online changepoint detection

The online detector of Adams and MacKay (2007) maintains, after each observation, a posterior distribution over the *run length* r_t, the number of months since the last changepoint. Let H be the

constant hazard (the prior probability that a change occurs at any month, here $1/60$, a prior mean run length of five years), and let the observations within a run be modelled as $y \sim \mathcal{N}(\mu, \sigma^2)$ with the conjugate Normal–Inverse-Gamma prior $\mu \mid \sigma^2 \sim \mathcal{N}(\mu_0, \sigma^2/\kappa_0)$, $\sigma^2 \sim \mathrm{IG}(\alpha_0, \beta_0)$. After r observations $y_1, \dots, y_r$ of the current run the hyperparameters update as

$$\kappa_r = \kappa_0 + r, \quad \mu_r = \frac{\kappa_0 \mu_0 + \sum_{i \le r} y_i}{\kappa_r}, \quad \alpha_r = \alpha_0 + \frac{r}{2}, \quad \beta_r = \beta_0 + \tfrac{1}{2} \sum_{i \le r} (y_i - \bar{y})^2 + \frac{\kappa_0 \, r \, (\bar{y} - \mu_0)^2}{2 \, \kappa_r}, \tag{49}$$

which the code computes recursively, one observation at a time, as $\mu' = (\kappa \mu + y)/(\kappa + 1)$, $\kappa' = \kappa + 1$, $\alpha' = \alpha + \tfrac{1}{2}$, $\beta' = \beta + \kappa (y - \mu)^2 / (2(\kappa + 1))$. The predictive distribution of the next observation given a run of length r is the Student-t

$$y_{t+1} \mid r_t = r \;\sim\; t_{2\alpha_r}\!\left(\mu_r, \; \frac{\beta_r (\kappa_r + 1)}{\alpha_r \kappa_r}\right), \tag{50}$$

and the run-length recursion is

$$P(r_{t+1} = r + 1, \, y_{1:t+1}) = P(r_t = r, \, y_{1:t}) \; p(y_{t+1} \mid r_t = r) \; (1 - H), \tag{51}$$

$$P(r_{t+1} = 0, \, y_{1:t+1}) = \sum_r P(r_t = r, \, y_{1:t}) \; p(y_{t+1} \mid r_t = r) \; H, \tag{52}$$

normalised at each step to give the posterior $R_{t+1}(r) = P(r_{t+1} = r \mid y_{1:t+1})$. The hyperparameters are initialised from the first six months of the series (μ_0 their mean, β_0 their variance, $\kappa_0 = 1$, $\alpha_0 = 1$), an empirical-Bayes choice that uses the start of the series twice and whose effect vanishes after a few dozen observations. Two detection statistics are derived. Because a constant hazard places mass H on $r = 0$ at every step by construction, the probability of a change "at t" is uninformative on its own; the code reports instead $P(r_t \le 3 \mid y_{1:t})$, the posterior probability that a change occurred within the last four months, and the maximum of this quantity over December 2022 to March 2023 is the statistic reported for the reference date. The second statistic is the sequence of maximum-a-posteriori run lengths $\hat{r}_t = \arg\max_r R_t(r)$; a *reset* is declared at month t when $\hat{r}_t \le 2$ after $\hat{r}_{t-1} > 12$, that is, when the most probable explanation of the data switches from a run of more than a year to a run that has just begun, and the list of reset months is the online detector's set of changepoints. The exact offline analogue (Fearnhead, 2006) would give the smoothed rather than the filtered posterior; the filtered version is used because the question it answers—when would a contemporaneous observer have detected the change—is the one relevant to the lead-time evaluation of Section 22.

21.7. Alternative event dates

The ITS is re-estimated for four alternative markers (June 2020 for GPT-3, May 2022 for DALL·E 2, April 2023 for GPT-4, October 2024 for o1), each coded as the first full month after the public release, and an event is skipped when fewer than twelve months precede or follow it in the series. The exercise is not a horse race among causes. Its purpose is to show whether the estimated level and slope changes are specific to the November 2022 marker or whether the series exhibits similar changes at other dates associated with the same technological sequence, in which case the honest description is of a multi-year acceleration rather than a single break; the Bai–Perron dates are the data-driven complement to this pre-specified list. For the total series the sensitivity of the reference-date ITS and of the Bai–Perron dates to the truncation month t_{L} is also tabulated.

22. Emerging-topic detection

The framework of RQ10 asks whether signals observable at a month T predict which topics will have grown unusually by $T+h$. Its scientific content lies less in the models, which are deliberately simple, than in the discipline of the evaluation: features are computed from data dated at or before T, labels from data strictly after T, training cutoffs end before test cutoffs begin, and the test labels are observed in a period the training never saw. The implementation is `experiments/emerging_detection.py`.

22.1. Units and panel

The units are OpenAlex topics (the `rank 0` topic of each work) with at least fifty eligible works over the whole window. Topics are used rather than the forty ontology categories because the categories are too few and too broad to emerge, and rather than the semantic clusters of Section 20 because a cluster defined on the full corpus is not an out-of-time object: it would be known to exist only because of works after T. The plan lists semantic clusters re-estimated at each cutoff and keyword n-grams as additional unit types; neither is implemented in the current pipeline, and the manuscript reports topic-level results only. Topics have their own limitation in this role: the OpenAlex topic model is itself fixed in 2026 and assigns 2019 works to topics defined with knowledge of 2019–2026 literature, so a topic that "exists" in 2019 with a handful of works may be a 2024 construct projected backward. The label definition below partly neutralises this, because it conditions on the topic having at least twenty works in the year before T, but the caveat applies to any retrospective evaluation with a fixed vocabulary and is stated as the first limitation of the framework.

The panel consists of two $K \times M$ matrices over topics k and months t from January 2018 to

the last month (October 2026 by default): the volume $n_{k,t}$, the number of month-precision eligible works of topic k with first month t, and the citation inflow $c_{k,t}$, the number of internal citation edges whose cited work belongs to topic k and whose citing work has first month t. Citation inflow is dated by the citing work's first month, not by the OpenAlex calendar-year counts, which is what makes it usable at monthly resolution and out of time.

22.2. Features at a cutoff

Write $V_k(a,b] = \sum_{t=a+1}^{b} n_{k,t}$ for the volume of topic k in the months $a < t \le b$ (and likewise C_k for citations). At cutoff T the feature vector of topic k is

$$g^{12}_{k,T} = \log\big(1+V_k(T-12,T]\big) - \log\big(1+V_k(T-24,T-12]\big), \tag{53}$$

$$\mathrm{acc}_{k,T} = g^{12}_{k,T} - g^{12}_{k,T-12}, \tag{54}$$

$$z_{k,T} = \frac{\frac{1}{3}V_k(T-3,T] - \bar{n}_{k,[T-27,T-3]}}{s_{k,[T-27,T-3]} + 10^{-6}}, \tag{55}$$

$$\mathrm{cit}_{k,T} = \log\big(1+C_k(T-12,T]\big) - \log\big(1+C_k(T-24,T-12]\big), \tag{56}$$

$$\mathrm{share}_{k,T} = V_k(T-12,T] \Big/ \sum_{k'} V_{k'}(T-12,T], \tag{57}$$

$$\mathrm{age}_{k,T} = T - \min\{t : \textstyle\sum_{u\le t} n_{k,u} \ge 5\}, \tag{58}$$

where $\bar{n}$ and s are the mean and standard deviation of the monthly volumes over the 25 months from $T-27$ to $T-3$. The first is twelve-month log growth, the second its change over a year (acceleration), the third a z-score of the last quarter's monthly volume against the trailing two years (the "abnormal growth" signal, which also serves as the baseline detector), the fourth citation acceleration, the fifth a size control, the sixth the topic's age in months since it reached five works. Every window ends at T; the one-month overlap between the z-score's trailing window and the quarter it is compared with is immaterial. The log-ratio form of growth with the $+1$ shift keeps the features finite for topics that were empty in the earlier window. Of the seven candidate signals of the mandate, the implemented set covers abnormal growth, citation acceleration and (through the panel of Section 20.4, which the script can merge but does not by default) semantic novelty; interdisciplinary connectivity is described in the script's documentation as a cross-area share but is not computed, and methodological adoption, benchmark proliferation and open-source activity are not implemented. These absences are recorded as scope limits of the current framework; the evaluation design does not depend on which features are present.

22.3. Label at horizon h

For a horizon $h \in \{12, 24\}$ months, the future growth of topic k after cutoff T is

$$g^{\text{fut}}_{k,T,h} = \log\big(1 + V_k(T, T+h]\big) - \log\big(1 + V_k(T-h, T]\big), \tag{59}$$

and the topic is *eligible* at T if $V_k(T-12, T] \geq 20$. The label is

$$\ell_{k,T,h} = \mathbf{1}\Big[g^{\text{fut}}_{k,T,h} \geq q_{0.9}\big(\{g^{\text{fut}}_{k',T,h} : k' \text{ eligible at } T\}\big) \wedge V_k(T, T+h] \geq 2\max\{V_k(T-h, T],\, 1\} \wedge k \text{ eligible} \Big], \tag{60}$$

the topic is in the top decile of future growth among eligible topics at that cutoff and at least doubles its volume. The decile is defined within cutoff, so that the positive rate is roughly ten per cent at every T regardless of the overall growth of the corpus; the doubling condition removes topics that reach the top decile by growing from a small base in a quiet period. The protocol phrases the label as "top-decile growth and citation share"; the implemented label uses volume growth only, and citation share is a feature rather than part of the outcome. This is a deliberate simplification—a citation-based outcome at a twelve-month horizon is dominated by the right-truncation of citations—and is recorded as a deviation. The label uses only $n_{k,t}$, so a topic can be "emerging" in this sense while its citation impact is unremarkable; Part IX reports the citation profile of the positives separately.

22.4. Cutoffs, models and metrics

Training cutoffs are every third month from December 2019 to December 2021; test cutoffs are every third month from January 2023 to October 2024 for which $T + h$ lies within the panel. With $h = 24$ the last training label uses data to December 2023, which is after the reference date but more than a year before the first test label window closes, and no test cutoff precedes 2023; the twelve months of 2022 are used by neither set as a cutoff, which is the temporal gap that separates training from test in the sense of Bergmeir and Benítez (2012). The plan proposed training cutoffs through 2022; the implementation stops at December 2021 so that the training labels do not extend into the test period at $h = 24$, a stricter separation than the plan. Rows are the eligible (topic, cutoff) pairs; features are the six quantities above with non-finite values replaced by zero.

Three predictors are compared: the z-score baseline, which ranks topics by $z_{k,T}$ alone and involves no fitting; an ℓ_2-regularised logistic regression (regularisation strength $C = 1$, on unstandardised features, so the penalty is not scale-invariant and acts most on the small-range features); and a gradient boosting classifier (Friedman, 2001) with 300 trees of depth three and learning rate 0.05, seed 44, as implemented in scikit-learn (Pedregosa et al., 2011). The metrics on the pooled test rows are the area under the ROC curve (Hanley and McNeil, 1982), which is the probability that a randomly chosen positive is ranked above a randomly chosen negative, and precision at $k \in \{20, 50\}$,

computed within each test cutoff as the share of positives among the k highest-ranked topics and averaged over cutoffs; the AUC is also reported per cutoff to show its stability. Hypothesis H8 sets the bar at an AUC above 0.70 and precision at 50 above 0.40 against the baseline. Because the positives are defined by a within-cutoff decile, precision at 50 has a chance level of about ten per cent, and a precision at 50 of 0.40 means that twenty of the fifty flagged topics were among the positives of that cutoff.

22.5. Lead time and the preregistered known-emergence set

The protocol fixes before estimation a list of twelve developments that anyone familiar with the period would call emergent (large language models, diffusion models, neural radiance fields, LLM agents, retrieval-augmented generation, state-space sequence models, vision–language models, mechanistic interpretability, speculative decoding, mixture of experts at scale, test-time compute and reasoning models, AI for code generation) and predicts that the detector flags each at least six months before its peak growth month. The evaluation requires a mapping from each development to one or more OpenAlex topics, and this mapping is the weak point of the design: it must be made by hand, after the taxonomy work, and a development that has no topic of its own (because the OpenAlex topic model folds it into a larger one) cannot be detected by a topic-level detector whatever its merits. The script reserves a table for this mapping and it is empty in the current version; the lead-time statistic is therefore [pending human review: known-emergence lead times, experiments/known_set.csv not yet populated], and the mapping, when made, will be listed in the appendix with the reason for each choice. Lead time itself is defined, for a development mapped to topic k, as the number of months between the first test cutoff at which k ranks in the top 50 and the month of maximum $g^{12}_{k,t}$ over the window; a development whose peak growth precedes the first test cutoff has undefined lead time and is reported as such rather than dropped.

22.6. Look-ahead safeguards and what they do not cover

The safeguards are structural: features are functions of panel columns up to T (enforced by index arithmetic, not by convention), labels of columns after T, cutoffs are separated by a gap, and the panel is built once from the corpus so that the same data serve all cutoffs. What they do not cover is information that entered the panel through steps performed in 2026 with full knowledge of the period: the topic vocabulary, the eligibility rules, the linkage, and the month-precision filter. A detector evaluated this way answers "given the 2026 topic model, would these signals have ranked the eventual winners early?", which is a legitimate question about the signals and a weaker one about the deployability of the system. The semantic-cluster extension that the plan describes, with clusters re-estimated at each cutoff from embeddings of works dated before it, is the design that

would address this, and it is left as the stated next step of the framework.

23. Knowledge-graph extraction

23.1. Structural and extracted layers

The knowledge graph has two layers of very different epistemic status. The structural layer—works, authors, institutions, countries, sources, OpenAlex topics, arXiv categories, taxonomy categories, and the citation and classification edges among them—is derived from the source databases by the deterministic rules of Sections 16–19; its errors are the errors of the sources and of the classifier, which are quantified. The extracted layer—methods, models, datasets, benchmarks, problems, metrics and claims, with typed edges to the works that introduce, use, evaluate on, extend, compare against or address them—is produced by a language model from abstracts, for the high-priority set only, and its errors are of a different kind: omissions, hallucinated entities, wrong roles, and claims paraphrased beyond what the abstract states. The mandate requires that every inferred relationship carry provenance, confidence and a verification status, and the extraction is designed so that these three fields are not decorative.

23.2. High-priority and landmark sets

The high-priority set is selected by `literature_synthesis/select_landmarks.py` from eligible, non-retracted works with a primary category: within each (primary category, publication year) cell the works are ranked by $\mathrm{NCS}^{(36)}$ (for 2025 and 2026, for which the window is empty, by $c^{(12)}$ divided by its year mean), and the top $\max\{25,\ \mathrm{round}(10{,}000\,n_{\mathrm{cell}}/n)\}$ are taken, so that cells are represented in proportion to their size with a floor that keeps small categories and recent years present. The landmark set takes the top $\max\{15, \lfloor 1000/40 \rfloor\} = 25$ works of each category by the same score, adds the 200 works with the highest $c^{(36)}$ overall, and removes duplicates; it is flagged for human review (checkpoint H-3) in its entirety. Both selections are citation-based and inherit the biases of citation counts: they favour the categories and years whose works are cited early, and they cannot contain a work whose influence has not yet appeared in citations, which for 2025–2026 means that the "high-priority" recent works are the early-cited ones, not necessarily the important ones.

23.3. Schema, provenance and machine verification

For each work in the high-priority set with an abstract, the extractor (`knowledge_graph/03_extract_entities.p`
submits the title, year and abstract (truncated to 3,000 characters), in batches of five works, to `claude-sonnet-5-5` with a JSON schema enforced by the API's structured output mode. The

schema requires, per work, a contribution type from {theory, method, empirical, benchmark, engineering, survey, other}; a list of entities each with a name, a kind in {method, model, dataset, benchmark, problem, metric}, a role in {introduced, used, evaluated_on, extended, compared_against, addressed}, a supporting *span* and a confidence; and a list of claims each with a text, a kind in {result, improvement, negative_result, reproduction, contradiction}, a span and a confidence. The system prompt instructs the model to quote a verbatim span of at most 25 words from the abstract for every entity and claim, to use canonical names, and never to infer anything not stated. The role vocabulary maps onto the edge types of the mandate (*introduces method*, *uses dataset*, *evaluates*, *extends*, *addresses research problem*); *contradicts* and *reproduces* are claim kinds rather than entity roles, because a contradiction or a reproduction is a relation between a work and a prior finding that an abstract rarely names as an entity.

Provenance is the pair (work identifier, span). Machine verification is the check that the span occurs verbatim, after lower-casing and trimming, in the abstract the model was given:

$$\text{status}(e) \;=\; \begin{cases} \text{machine-verified} & \text{if } \text{lower}(\text{span}(e)) \subseteq \text{lower}(\text{abs}(w)), \\ \text{unverified} & \text{otherwise,} \end{cases} \tag{61}$$

with *human-verified* reserved for edges the principal investigator has reviewed. The check is weak in a precise sense: it establishes that the evidence quoted for an edge exists, not that it supports the edge. A model that quotes a genuine sentence and attaches the wrong role to it produces a machine-verified wrong edge. The confidence field is the model's own stated probability that the extraction is correct and is not calibrated by this study; it is stored so that downstream users can threshold it and so that its calibration can be estimated on the human-verified subsample. Every record carries `evidence_type = abstract`, and the manuscript treats abstract-derived edges as what they are: evidence that an abstract asserts a relation, which is not a full-text review and is never cited as establishing a scientific finding (CLAUDE.md §1). Extractions are appended to a resumable file, requests that the model refuses are skipped and counted, and the cost is tracked against the budget of amendment A-004. What is and is not evidence can therefore be stated plainly: citation edges are evidence of citation; classification edges are evidence of rule firing with measured agreement; extracted edges are evidence that a language model, reading an abstract, asserted a relation and quoted a sentence that exists.

24. Uncertainty, provenance and reproducibility

24.1. What is uncertain and how it is quantified

The descriptive quantities of Sections 18 and 19 are computed on the whole eligible corpus and carry no sampling error; their uncertainty is measurement error, which enters through linkage (Section 16.4), classification (Section 17.3), reference resolution and the completeness of affiliations and abstracts, each of which is estimated and reported as a rate rather than propagated into intervals. Propagating a classification precision of, say, 0.85 into an interval for a category share would require a model of how errors are distributed over time, and the honest statement is that a category's share is known up to the classifier's error, which is reported by category and, where the validation sample permits, by period.

Three families of quantities do carry explicit intervals. The ITS coefficients carry HAC standard errors and the placebo rank of Section 21.3; the Bai–Perron break dates carry no interval in the current implementation and are reported with the agreement of the two criteria and the sensitivity to truncation. The detector metrics carry per-cutoff values, whose dispersion across cutoffs is the relevant measure of stability, since the rows within a cutoff are not independent. And the cluster count and the category similarity matrices carry the split-half ARI. Where the manuscript reports a bootstrap interval for a corpus statistic (the Gini coefficient by year, the cross-area shares), it is the percentile interval of the non-parametric bootstrap (Efron, 1979) over works, with 1,000 replications and seed 44, computed by the table-making script at reporting time; the protocol's commitment to "bootstrap or HAC intervals" is met by these two devices, and a reported figure without an interval is a population count for which none is appropriate. The indicators of Section 18 that depend on OpenAlex's nightly recomputation (FWCI, total citation counts) are snapshot quantities and are dated.

24.2. Seeds and determinism

Every stochastic step is seeded with the constant 44: the validation and linkage samples (hash ordering salted with the seed), the atlas sample and the two UMAP fits, the split-half permutations, the gradient boosting classifier, and the bootstrap. The regular-expression classifier, the SQL stages and the structural-change toolkit are deterministic. Two steps are deterministic only up to the platform: embedding on Apple-silicon GPUs with float16 storage produces vectors that differ at the fourth decimal from a CPU run, and UMAP's parallel optimisation is reproducible only with a fixed seed and a fixed thread count, which the runbook records. The LLM-based steps (two taxonomy raters, one extractor) are not deterministic even at zero temperature across model versions; their outputs are archived in full (the JSONL rating and extraction files, with the model identifier and the

usage of every call) so that every downstream number can be recomputed from the archived outputs without re-querying the model, and a re-run with a later model version is a new experiment, not a replication.

24.3. Ledgers, hashes and the chain of custody

The raw inputs are pinned as follows. Each OpenAlex snapshot file downloaded from the public bucket is hashed with SHA-256 on arrival and the hash, record count, computer-science record count and malformed-record count are written to a SQLite ledger keyed by the file's URL and manifest entry; a file whose hash or count does not match the manifest is re-downloaded. The DBLP XML dump is identified by the MD5 published with it (verified before parsing) and its size; the ACL Anthology export by the SHA-256 of the file; the arXiv OAI-PMH harvest by the raw XML pages, kept verbatim, from which the Parquet table is rebuilt. Every derived table is written by a script whose invocation, inputs, outputs and measured wall time are listed in `reproducibility/RUNBOOK.md`, and the Python environment is locked (`reproducibility/environment_M3U96b.txt`). Numbers enter the manuscript only through `manuscript/tables/numbers.tex`, generated by `tables/make_numbers.py` from the result files, so that a number in the text can be traced to a macro, the macro to a result file, the result file to a script and the script to a ledger entry. The chain has one weak link, stated here: the result files on the compute node are not themselves hashed into the repository, so a reader verifying the chain must either re-run the pipeline or trust the node's copy; the release archive of Phase 9 will add their hashes.

24.4. Human checkpoints and the status of conclusions

Five checkpoints (H-1 to H-5, `reports/HUMAN_REVIEW_CHECKPOINTS.md`) require the principal investigator's review of the linkage sample, the taxonomy validation sample, the landmark list, the structural-change results and the conclusions. Until a checkpoint is signed, the manuscript marks the conclusions that depend on it as pending human review, and the numbers that depend on it are rendered by the `\pending` macro rather than typed. This is the mechanism by which the mandate's requirement that no agent approve its own findings is made visible in the text rather than asserted in a preface; it also means that a version of this manuscript compiled before the checkpoints are signed contains visibly incomplete statements, which is the intended behaviour.

Part V

The Artificial Intelligence Transformation

Part VI

Systems and Computing Infrastructure

Part VII

Theory, Algorithms, and Security

Part VIII

Scientific Knowledge Networks

Part IX

Empirical Results

Part X

The Next Research Frontiers

Part XI

Discussion and Conclusion

References

Abadi, Martin, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang, "Deep Learning with Differential Privacy," in "Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security" ACM 2016, pp. 308–318.

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: A system for large-scale machine learning," arXiv preprint arXiv:1605.08695 2016.

Abramson, Josh, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick et al., "Accurate structure prediction of biomolecular interactions with AlphaFold 3," *Nature*, 2024, *630* (8016), 493–500.

Adams, Ryan Prescott and David J. C. MacKay, "Bayesian Online Changepoint Detection," arXiv preprint arXiv:0710.3742 2007.

Ahmed, Nur and Muntasir Wahed, "The De-democratization of AI: Deep Learning and the Compute Divide in Artificial Intelligence Research," arXiv preprint arXiv:2010.15581 2020.

_ , _ , and Neil C. Thompson, "The growing influence of industry in AI research," *Science*, 2023, *379* (6635), 884–886.

Alayrac, Jean-Baptiste, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan, "Flamingo: a Visual Language Model for Few-Shot Learning," arXiv preprint arXiv:2204.14198 2022.

Allamanis, Miltiadis, Earl T. Barr, Premkumar Devanbu, and Charles Sutton, "A Survey of Machine Learning for Big Code and Naturalness," *ACM Computing Surveys*, 2018, *51* (4), 1–37.

Alperin, Juan Pablo, Jason Portenoy, Kyle Demes, Vincent Larivière, and Stefanie Haustein, "An analysis of the suitability of OpenAlex for bibliometric analyses," arXiv preprint arXiv:2404.17663 2024.

Amershi, Saleema, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz, "Guidelines for Human-AI Interaction," in "Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems" ACM 2019, pp. 1–13.

Andrews, Donald W. K., "Heteroskedasticity and autocorrelation consistent covariance matrix estimation," *Econometrica*, 1991, *59* (3), 817–858.

_ , "Tests for Parameter Instability and Structural Change With Unknown Change Point," *Econometrica*, 1993, *61* (4), 821.

Armbrust, Michael, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia, "A view of cloud computing," *Communications of the ACM*, 2010, *53* (4), 50–58.

Arute, Frank, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis, "Quantum supremacy using a programmable superconducting processor," *Nature*, 2019, *574* (7779), 505–510.

arXiv, "arXiv monthly submission statistics," Cornell University 2026.

Association for Computational Linguistics, "ACL Anthology," ACL 2026.

Baek, Minkyung, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N. Kinch, R. Dustin Schaeffer, Claudia Millán, Hahnbeom Park, Carson Adams, Caleb R. Glassman, Andy DeGiovanni, Jose H. Pereira,

Andria V. Rodrigues, Alberdina A. van Dijk, Ana C. Ebrecht, Diederik J. Opperman, Theo Sagmeister, Christoph Buhlheller, Tea Pavkov-Keller, Manoj K. Rathinaswamy, Udit Dalwadi, Calvin K. Yip, John E. Burke, K. Christopher Garcia, Nick V. Grishin, Paul D. Adams, Randy J. Read, and David Baker, "Accurate prediction of protein structures and interactions using a three-track neural network," *Science*, 2021, *373* (6557), 871–876.

Baevski, Alexei, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," arXiv preprint arXiv:2006.11477 2020.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," arXiv preprint arXiv:1409.0473 2014.

Bai, Jushan and Pierre Perron, "Estimating and Testing Linear Models with Multiple Structural Changes," *Econometrica*, 1998, *66* (1), 47.

_ and _ , "Computation and analysis of multiple structural change models," *Journal of Applied Econometrics*, 2003, *18* (1), 1–22.

Bai, Yuntao, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan, "Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback," arXiv preprint arXiv:2204.05862 2022.

Barroso, Luiz André, Urs Hölzle, and Parthasarathy Ranganathan, *The Datacenter as a Computer: Designing Warehouse-Scale Machines, Third Edition* Synthesis Lectures on Computer Architecture, Morgan & Claypool, 2018.

Belkin, Mikhail, Daniel Hsu, Siyuan Ma, and Soumik Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proceedings of the National Academy of Sciences*, 2019, *116* (32), 15849–15854.

Bender, Emily M., Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell, "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?," in "Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency" ACM 2021, pp. 610–623.

Bergmeir, Christoph and José M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences*, 2012, *191*, 192–213.

Bernal, James Lopez, Steven Cummins, and Antonio Gasparrini, "Interrupted time series regression for the evaluation of public health interventions: a tutorial," *International Journal of Epidemiology*, 2016, p. dyw098.

_ , _ , and _ , "Interrupted time series regression for the evaluation of public health interventions: a tutorial," *International Journal of Epidemiology*, 2016, p. dyw098.

_ , _ , and _ , "The use of controls in interrupted time series studies of public health interventions," *International Journal of Epidemiology*, 2018, *47* (6), 2082–2093.

Berners-Lee, Tim, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret, "The World-Wide Web," *Communications of the ACM*, 1994, *37* (8), 76–82.

Bernstein, Daniel J. and Tanja Lange, "Post-quantum cryptography," *Nature*, 2017, *549* (7671), 188–194.

Bharti, Kishor, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik, "Noisy intermediate-scale quantum algorithms," *Reviews of Modern Physics*, 2022, *94* (1).

Biggio, Battista and Fabio Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, 2018, *84*, 317–331.

BigScience Workshop, "BLOOM: A 176B-Parameter Open-Access Multilingual Language Model," arXiv preprint arXiv:2211.05100 2022.

Bin-Nashwan, Saeed Awadh, Mouad Sadallah, and Mohamed Bouteraa, "Use of ChatGPT in academia: Academic integrity hangs in the balance," *Technology in Society*, 2023, *75*, 102370.

Birhane, Abeba, Atoosa Kasirzadeh, David Leslie, and Sandra Wachter, "Science in the age of large language models," *Nature Reviews Physics*, 2023, *5* (5), 277–280.

Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong,

Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang, "On the Opportunities and Risks of Foundation Models," arXiv preprint arXiv:2108.07258 2021.

Bonneau, Joseph, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten, "SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies," in "2015 IEEE Symposium on Security and Privacy" IEEE 2015, pp. 104–121.

Bornmann, Lutz and Rüdiger Mutz, "Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references," *Journal of the Association for Information Science and Technology*, 2015, *66* (11), 2215–2222.

_ , Robin Haunschild, and Rüdiger Mutz, "Growth rates of modern science: a latent piecewise growth curve approach to model publication numbers from established and new literature databases," *Humanities and Social Sciences Communications*, 2021, *8* (1).

Bosshart, Pat, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker, "P4: programming protocol-independent packet processors," *ACM SIGCOMM Computer Communication Review*, 2014, *44* (3), 87–95.

Brin, Sergey and Lawrence Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, 1998, *30* (1-7), 107–117.

Brohan, Anthony, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol

Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," arXiv preprint arXiv:2307.15818 2023.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei, "Language Models are Few-Shot Learners," arXiv preprint arXiv:2005.14165 2020.

Bubeck, Sébastien, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang, "Sparks of Artificial General Intelligence: Early experiments with GPT-4," arXiv preprint arXiv:2303.12712 2023.

Cabanac, Guillaume, Theodora Oikonomidi, and Isabelle Boutron, "Day-to-day discovery of preprint–publication links," *Scientometrics*, 2021, *126*, 5285–5304.

Campello, Ricardo J. G. B., Davoud Moulavi, and Joerg Sander, "Density-Based Clustering Based on Hierarchical Density Estimates," in "Lecture Notes in Computer Science," Springer Berlin Heidelberg, 2013, pp. 160–172.

Carlini, Nicholas and David Wagner, "Towards Evaluating the Robustness of Neural Networks," in "2017 IEEE Symposium on Security and Privacy (SP)" IEEE 2017, pp. 39–57.

Caron, Mathilde, Hugo Touvron, Ishan Misra, Herve Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin, "Emerging Properties in Self-Supervised Vision Transformers," in "2021 IEEE/CVF International Conference on Computer Vision (ICCV)" IEEE 2021, pp. 9630–9640.

Ceriani, Lidia and Paolo Verme, "The origins of the Gini index: extracts from Variabilità e Mutabilità (1912) by Corrado Gini," *The Journal of Economic Inequality*, 2012, *10* (3), 421–443.

Chen, Mark, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba, "Evaluating Large Language Models Trained on Code," arXiv preprint arXiv:2107.03374 2021.

Chen, Tianqi, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Meghan Cowan, Haichen Shen, Leyuan Wang, Yuwei Hu, Luis Ceze, Carlos Guestrin, and Arvind Krishnamurthy, "TVM: An Automated End-to-End Optimizing Compiler for Deep Learning," arXiv preprint arXiv:1802.04799 2018.

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," arXiv preprint arXiv:2002.05709 2020.

Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel, "PaLM: Scaling Language Modeling with Pathways," arXiv preprint arXiv:2204.02311 2022.

Christen, Peter, *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, Berlin, Heidelberg: Springer, 2012.

Christiano, Paul, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei, "Deep reinforcement learning from human preferences," arXiv preprint arXiv:1706.03741 2017.

Chu, Johan S. G. and James A. Evans, "Slowed canonical progress in large fields of science," *Proceedings of the National Academy of Sciences*, 2021, *118* (41).

Clement, Colin B., Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi, "On the use of arXiv as a dataset," arXiv preprint arXiv:1905.00075 2019.

Cohan, Arman, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld, "SPECTER: Document-level Representation Learning using Citation-informed Transformers," in "Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics" Association for Computational Linguistics 2020, pp. 2270–2282.

Cohen, Jacob, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, 1960, *20* (1), 37–46.

Culbert, Jack, Anne Hobert, Najko Jahn, Nick Haupka, Marion Schmidt, Paul Donner, and Philipp Mayr, "Reference Coverage Analysis of OpenAlex compared to Web of Science and Scopus," arXiv preprint arXiv:2401.16359 2024.

Culbert, Jack H., Anne Hobert, Najko Jahn, Nick Haupka, Marion Schmidt, Paul Donner, and Philipp Mayr, "Reference coverage analysis of OpenAlex compared to Web of Science and Scopus," *Scientometrics*, 2025, *130*, 2475–2492.

Dao, Tri, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré, "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness," arXiv preprint arXiv:2205.14135 2022.

de Moura, Leonardo and Sebastian Ullrich, "The Lean 4 Theorem Prover and Programming Language," in "Lecture Notes in Computer Science," Springer International Publishing, 2021, pp. 625–635.

de Moura, Leonardo, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer, "The Lean Theorem Prover (System Description)," in "Automated Deduction – CADE-25" Lecture Notes in Computer Science, Springer International Publishing, 2015, pp. 378–388.

Dean, Jeffrey and Luiz André Barroso, "The tail at scale," *Communications of the ACM*, 2013, *56* (2), 74–80.

_ **and Sanjay Ghemawat**, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, 2008, *51* (1), 107–113.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," arXiv preprint arXiv:2501.12948 2025.

Dehghani, Mostafa, Yi Tay, Alexey A. Gritsenko, Zhe Zhao, Neil Houlsby, Fernando Diaz, Donald Metzler, and Oriol Vinyals, "The Benchmark Lottery," arXiv preprint arXiv:2107.07002

2021.

Delgado-Quirós, Lorena and José Luis Ortega, "Completeness degree of publication metadata in eight free-access scholarly databases," *Quantitative Science Studies*, 2024, *5* (1), 31–49.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in "2009 IEEE Conference on Computer Vision and Pattern Recognition" IEEE 2009, pp. 248–255.

Dennard, R.H., F.H. Gaensslen, Hwa-Nien Yu, V.L. Rideout, E. Bassous, and A.R. LeBlanc, "Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, 1974, *9* (5), 256–268.

Denning, Peter J., "Is computer science science?," *Communications of the ACM*, 2005, *48* (4), 27–31.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in "Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)" Association for Computational Linguistics 2019, pp. 4171–4186.

Dhariwal, Prafulla and Alex Nichol, "Diffusion Models Beat GANs on Image Synthesis," arXiv preprint arXiv:2105.05233 2021.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," arXiv preprint arXiv:2010.11929 2020.

Dwivedi, Yogesh K., Nir Kshetri, Laurie Hughes, Emma Louise Slade, Anand Jeyaraj, Arpan Kumar Kar, Abdullah M. Baabdullah, Alex Koohang, Vishnupriya Raghavan, Manju Ahuja, Hanaa Albanna, Mousa Ahmad Albashrawi, Adil S. Al-Busaidi, Janarthanan Balakrishnan, Yves Barlette, Sriparna Basu, Indranil Bose, Laurence Brooks, Dimitrios Buhalis, Lemuria Carter, Soumyadeb Chowdhury, Tom Crick, Scott W. Cunningham, Gareth H. Davies, Robert M. Davison, Rahul Dé, Denis Dennehy, Yanqing Duan, Rameshwar Dubey, Rohita Dwivedi, John S. Edwards, Carlos Flavián, Robin Gauld, Varun Grover, Mei-Chih Hu, Marijn Janssen, Paul Jones, Iris Junglas, Sangeeta Khorana, Sascha Kraus, Kai R. Larsen, Paul Latreille, Sven Laumer, F. Tegwen Malik, Abbas Mardani, Marcello Mariani, Sunil Mithas, Emmanuel Mogaji, Jeretta Horn Nord, Siobhan O'Connor, Fevzi Okumus,

Margherita Pagani, Neeraj Pandey, Savvas Papagiannidis, Ilias O. Pappas, Nishith Pathak, Jan Pries-Heje, Ramakrishnan Raman, Nripendra P. Rana, Sven-Volker Rehm, Samuel Ribeiro-Navarrete, Alexander Richter, Frantz Rowe, Suprateek Sarker, Bernd Carsten Stahl, Manoj Kumar Tiwari, Wil van der Aalst, Viswanath Venkatesh, Giampaolo Viglia, Michael Wade, Paul Walton, Jochen Wirtz, and Ryan Wright, "Opinion Paper: "So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy," *International Journal of Information Management*, 2023, *71*, 102642.

Dwork, Cynthia and Aaron Roth, "The Algorithmic Foundations of Differential Privacy," *Foundations and Trends® in Theoretical Computer Science*, 2014, *9* (3-4), 211–487.

_ , Frank McSherry, Kobbi Nissim, and Adam Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in "Theory of Cryptography (TCC 2006)" Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 265–284.

Efron, B., "Bootstrap methods: Another look at the jackknife," *The Annals of Statistics*, 1979, *7* (1), 1–26.

Esmaeilzadeh, Hadi, Emily Blem, Renee St. Amant, Karthikeyan Sankaralingam, and Doug Burger, "Dark silicon and the end of multicore scaling," in "Proceedings of the 38th annual international symposium on Computer architecture" ACM 2011, pp. 365–376.

Fearnhead, Paul, "Exact and efficient Bayesian inference for multiple changepoint problems," *Statistics and Computing*, 2006, *16* (2), 203–213.

Fedus, William, Barret Zoph, and Noam Shazeer, "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity," arXiv preprint arXiv:2101.03961 2021.

Fellegi, Ivan P. and Alan B. Sunter, "A theory for record linkage," *Journal of the American Statistical Association*, 1969, *64* (328), 1183–1210.

Fortnow, Lance, "ViewpointTime for computer science to grow up," *Communications of the ACM*, 2009, *52* (8), 33–35.

Fortunato, Santo, "Community detection in graphs," *Physics Reports*, 2010, *486* (3-5), 75–174.

_ and Marc Barthélemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, 2007, *104* (1), 36–41.

_ , Carl T. Bergstrom, Katy Börner, James A. Evans, Dirk Helbing, Staša Milojević, Alexander M. Petersen, Filippo Radicchi, Roberta Sinatra, Brian Uzzi, Alessandro Vespignani, Ludo Waltman, Dashun Wang, and Albert-László Barabási, "Science of science," *Science*, 2018, *359* (6379), eaao0185.

_ , _ , Katy Börner, James A. Evans, Dirk Helbing, Staša Milojević, Alexander M. Petersen, Filippo Radicchi, Roberta Sinatra, Brian Uzzi et al., "Science of science," *Science*, 2018, *359* (6379).

Friedman, Jerome H., "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, 2001, *29* (5), 1189–1232.

Geng, Mingmeng and Roberto Trotta, "Is ChatGPT Transforming Academics' Writing Style?," arXiv preprint arXiv:2404.08627 2024.

Gentry, Craig, "Fully homomorphic encryption using ideal lattices," in "Proceedings of the forty-first annual ACM symposium on Theory of computing" ACM 2009, pp. 169–178.

Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung, "The Google file system," in "Proceedings of the nineteenth ACM symposium on Operating systems principles" ACM 2003, pp. 29–43.

Ginsparg, Paul, "ArXiv at 20," *Nature*, 2011, *476* (7359), 145–147.

Glänzel, Wolfgang and András Schubert, "A new classification scheme of science fields and subfields designed for scientometric evaluation purposes," *Scientometrics*, 2003, *56* (3), 357–367.

Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative Adversarial Networks," arXiv preprint arXiv:1406.2661 2014.

Google Quantum AI, "Suppressing quantum errors by scaling a surface code logical qubit," *Nature*, 2023, *614* (7949), 676–681.

Google Quantum AI and Collaborators, "Quantum error correction below the surface code threshold," *Nature*, 2025, *638* (8052), 920–926.

Grattafiori, Aaron, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan et al., "The Llama 3 Herd of Models," arXiv preprint arXiv:2407.21783 2024.

Gray, Andrew, "ChatGPT "contamination": estimating the prevalence of LLMs in the scholarly literature," arXiv preprint arXiv:2403.16887 2024.

Gu, Albert and Tri Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," arXiv preprint arXiv:2312.00752 2023.

Gundersen, Odd Erik and Sigbjørn Kjensmo, "State of the Art: Reproducibility in Artificial Intelligence," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, *32* (1).

Haddaway, Neal R., Matthew J. Page, Chris C. Pritchard, and Luke A. McGuinness, "PRISMA2020 : An R package and Shiny app for producing PRISMA 2020-compliant flow diagrams, with interactivity for optimised digital transparency and Open Synthesis," *Campbell Systematic Reviews*, 2022, *18* (2).

HALES, THOMAS, MARK ADAMS, GERTRUD BAUER, TAT DAT DANG, JOHN HARRISON, LE TRUONG HOANG, CEZARY KALISZYK, VICTOR MAGRON, SEAN MCLAUGHLIN, TAT THANG NGUYEN et al., "A FORMAL PROOF OF THE KEPLER CONJECTURE," *Forum of Mathematics, Pi*, 2017, *5*.

Hanley, James A. and Barbara J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, 1982, *143* (1), 29–36.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," in "2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)" IEEE 2016, pp. 770–778.

_ , Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollar, and Ross Girshick, "Masked Autoencoders Are Scalable Vision Learners," in "2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)" IEEE 2022, pp. 15979–15988.

Hellerstein, Joseph M., Jose Faleiro, Joseph E. Gonzalez, Johann Schleier-Smith, Vikram Sreekanti, Alexey Tumanov, and Chenggang Wu, "Serverless Computing: One Step Forward, Two Steps Back," arXiv preprint arXiv:1812.03651 2018.

Hendricks, Ginny, Dominika Tkaczyk, Jennifer Lin, and Patricia Feeney, "Crossref: The sustainable source of community-owned scholarly metadata," *Quantitative Science Studies*, 2020, *1* (1), 414–427.

Hendrycks, Dan, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt, "Measuring Massive Multitask Language Understanding," arXiv preprint arXiv:2009.03300 2020.

_ , **Nicholas Carlini, John Schulman, and Jacob Steinhardt**, "Unsolved Problems in ML Safety," arXiv preprint arXiv:2109.13916 2021.

Hennessy, John L. and David A. Patterson, "A new golden age for computer architecture," *Communications of the ACM*, 2019, *62* (2), 48–60.

Hinton, Geoffrey, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, 2012, *29* (6), 82–97.

Hirschman, Albert O., "The paternity of an index," *American Economic Review*, 1964, *54* (5), 761–762. OpenAlex W2163989766.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel, "Denoising Diffusion Probabilistic Models," arXiv preprint arXiv:2006.11239 2020.

Hochreiter, Sepp and Jürgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, 1997, *9* (8), 1735–1780.

Hoffmann, Jordan, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre, "Training Compute-Optimal Large Language Models," arXiv preprint arXiv:2203.15556 2022.

Hubert, Lawrence and Phipps Arabie, "Comparing partitions," *Journal of Classification*, 1985, *2* (1), 193–218.

Jackson, Brad, Jeffrey D. Scargle, David Barnes, Sundararajan Arabhi, Alina Alt, Peter Gioumousis, Elyus Gwin, Paungkaew Sangtrakulcharoen, Linda Tan, and Tun Tao Tsai, "An algorithm for optimal partitioning of data on an interval," *IEEE Signal Processing Letters*, 2005, *12* (2), 105–108.

Jacot, Arthur, Franck Gabriel, and Clément Hongler, "Neural Tangent Kernel: Convergence and Generalization in Neural Networks," arXiv preprint arXiv:1806.07572 2018.

Johnson, Jeff, Matthijs Douze, and Herve Jegou, "Billion-Scale Similarity Search with GPUs," *IEEE Transactions on Big Data*, 2021, *7* (3), 535–547.

Jonas, Eric, Johann Schleier-Smith, Vikram Sreekanti, Chia-Che Tsai, Anurag Khandelwal, Qifan Pu, Vaishaal Shankar, Joao Carreira, Karl Krauth, Neeraja Yadwadkar, Joseph E. Gonzalez, Raluca Ada Popa, Ion Stoica, and David A. Patterson, "Cloud Programming Simplified: A Berkeley View on Serverless Computing," arXiv preprint arXiv:1902.03383 2019.

Jouppi, Norman P., Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Daniel Killebrew, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon, "In-Datacenter Performance Analysis of a Tensor Processing Unit," in "Proceedings of the 44th Annual International Symposium on Computer Architecture" ACM 2017, pp. 1–12.

Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis, "Highly accurate protein structure prediction with AlphaFold," *Nature*, 2021, *596* (7873), 583–589.

Jung, Ralf, Jacques-Henri Jourdan, Robbert Krebbers, and Derek Dreyer, "RustBelt: securing the foundations of the Rust programming language," *Proceedings of the ACM on Programming Languages*, 2018, *2* (POPL), 1–34.

Kairouz, Peter and H. Brendan McMahan, "Advances and Open Problems in Federated Learning," *Foundations and Trends® in Machine Learning*, 2021, *14* (1-2), 1–210.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei, "Scaling Laws for Neural Language Models," arXiv preprint arXiv:2001.08361 2020.

Kapoor, Sayash and Arvind Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, 2023, *4* (9), 100804.

Karniadakis, George Em, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang, "Physics-informed machine learning," *Nature Reviews Physics*, 2021, *3* (6), 422–440.

Karpukhin, Vladimir, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih, "Dense Passage Retrieval for Open-Domain Question Answering," in "Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)" Association for Computational Linguistics 2020, pp. 6769–6781.

Kerbl, Bernhard, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Transactions on Graphics*, 2023, *42* (4), 1–14.

Killick, R., P. Fearnhead, and I. A. Eckley, "Optimal Detection of Changepoints With a Linear Computational Cost," *Journal of the American Statistical Association*, 2012, *107* (500), 1590–1598.

Kingma, Diederik P and Max Welling, "Auto-Encoding Variational Bayes," arXiv preprint arXiv:1312.6114 2013.

Kinney, Rodney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan et al., "The Semantic Scholar Open Data Platform," arXiv preprint arXiv:2301.10140 2023.

_ , _ , _ , _ , _ , _ , _ , _ , _ , _ , Miles Crawford, Doug Downey, Jason Dunkelberger, Oren Etzioni, Rob Evans, Sergey Feldman, Joseph Gorney, David Graham, Fangzhou Hu, Regan Huff, Daniel King, Sebastian Kohlmeier, Bailey Kuehl, Michael Langan, Daniel Lin, Haokun Liu, Kyle Lo, Jaron Lochner, Kelsey MacMillan, Tyler Murray, Chris Newell, Smita Rao, Shaurya Rohatgi, Paul Sayre, Zejiang Shen, Amanpreet Singh, Luca Soldaini, Shivashankar Subramanian, Amber Tanaka, Alex D. Wade, Linda Wagner, Lucy Lu Wang, Chris Wilhelm, Caroline Wu, Jiangjiang Yang, Angele Zamarron, Madeleine Van Zuylen, and Daniel S. Weld, "The Semantic Scholar Open Data Platform," arXiv preprint arXiv:2301.10140 2023.

Klavans, Richard and Kevin W. Boyack, "Which Type of Citation Analysis Generates the Most Accurate Taxonomy of Scientific and Technical Knowledge?," *Journal of the Association for Information Science and Technology*, 2017, *68* (4), 984–998.

Klein, Gerwin, Kevin Elphinstone, Gernot Heiser, June Andronick, David Cock, Philip Derrin, Dhammika Elkaduwe, Kai Engelhardt, Rafal Kolanski, Michael Norrish, Thomas Sewell, Harvey Tuch, and Simon Winwood, "seL4: formal verification of an OS kernel," in "Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles" ACM 2009, pp. 207–220.

Kleinberg, Jon M., "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, 1999, *46* (5), 604–632.

Kobak, Dmitry, Rita González-Márquez, Emőke Ágnes Horvát, and Jan Lause, "Delving into LLM-assisted writing in biomedical publications through excess vocabulary," *Science Advances*, 2025, *11* (27).

Kocher, Paul, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom, "Spectre Attacks: Exploiting Speculative Execution," in "2019 IEEE Symposium on Security and Privacy (SP)" IEEE 2019, pp. 1–19.

Kraska, Tim, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis, "The Case for Learned Index Structures," in "Proceedings of the 2018 International Conference on Management of Data" ACM 2018, pp. 489–504.

Krenn, Mario, Lorenzo Buffoni, Bruno Coutinho, Sagi Eppel, Jacob Gates Foster, Andrew Gritsevskiy, Harlin Lee, Yichao Lu, João P. Moutinho, Nima Sanjabi, Rishi Sonthalia, Ngoc Mai Tran, Francisco Valente, Yangxinyu Xie, Rose Yu, and Michael Kopp, "Forecasting the future of artificial intelligence with machine learning-based link prediction in an exponentially growing knowledge network," *Nature Machine Intelligence*, 2023, *5* (11), 1326–1335.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, 2017, *60* (6), 84–90.

Langley, Adam, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar et al., "The QUIC Transport Protocol: Design and Internet-Scale Deployment," in "Proceedings of the Conference of the ACM Special Interest Group on Data Communication" ACM 2017, pp. 183–196.

Lattner, Chris, Mehdi Amini, Uday Bondhugula, Albert Cohen, Andy Davis, Jacques Pienaar, River Riddle, Tatiana Shpeisman, Nicolas Vasilache, and Oleksandr Zinenko, "MLIR: Scaling Compiler Infrastructure for Domain Specific Computation," in "2021 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)" IEEE 2021, pp. 2–14.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, 2015, *521* (7553), 436–444.

Leicht, E. A. and M. E. J. Newman, "Community structure in directed networks," *Physical Review Letters*, 2008, *100* (11), 118703.

Leinster, Tom and Christina A. Cobbold, "Measuring diversity: the importance of species similarity," *Ecology*, 2012, *93* (3), 477–489.

Leiserson, Charles E., Neil C. Thompson, Joel S. Emer, Bradley C. Kuszmaul, Butler W. Lampson, Daniel Sanchez, and Tao B. Schardl, "There's plenty of room at the Top: What will drive computer performance after Moore's law?," *Science*, 2020, *368* (6495).

Leroy, Xavier, "Formal verification of a realistic compiler," *Communications of the ACM*, 2009, *52* (7), 107–115.

Levine, Sergey, Chelsea Finn, Trevor Darrell, and Pieter Abbeel, "End-to-End Training of Deep Visuomotor Policies," arXiv preprint arXiv:1504.00702 2015.

Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," arXiv preprint arXiv:2005.11401 2020.

Ley, Michael, "The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives," in "Lecture Notes in Computer Science," Springer Berlin Heidelberg, 2002, pp. 1–10.

_ , "DBLP: some lessons learned," *Proceedings of the VLDB Endowment*, 2009, *2* (2), 1493–1500.

Li, Zongyi, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar, "Fourier Neural Operator for Parametric Partial Differential Equations," arXiv preprint arXiv:2010.08895 2020.

Liang, Weixin, Yaohui Zhang, Mihai Codreanu, Jiayu Wang, Hancheng Cao, and James Zou, "The Widespread Adoption of Large Language Model-Assisted Writing Across Society," arXiv preprint arXiv:2502.09747 2025.

_ , _ , Zhengxuan Wu, Haley Lepp, Wenlong Ji, Xuandong Zhao, Hancheng Cao, Sheng Liu, Siyu He, Zhi Huang, Diyi Yang, Christopher Potts, Christopher D Manning, and James Y. Zou, "Mapping the Increasing Use of LLMs in Scientific Papers," arXiv preprint arXiv:2404.01268 2024.

_ , Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Haotian Ye, Sheng Liu, Zhi Huang, Daniel A. McFarland, and James Y. Zou, "Monitoring AI-Modified Content at Scale: A Case Study on the Impact of ChatGPT on AI Conference Peer Reviews," arXiv preprint arXiv:2403.07183 2024.

Lim, Kevin, Jichuan Chang, Trevor Mudge, Parthasarathy Ranganathan, Steven K. Reinhardt, and Thomas F. Wenisch, "Disaggregated memory for expansion and sharing in blade servers," in "Proceedings of the 36th annual international symposium on Computer architecture" ACM 2009, pp. 267–278.

Lin, Jialiang, Yao Yu, Yu Zhou, Zhiyang Zhou, and Xiaodong Shi, "How many preprints have actually been printed and why: a case study of computer science preprints on arXiv," *Scientometrics*, 2020, *124* (1), 555–574.

Lin, Zeming, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli et al., "Evolutionary-scale prediction of atomic-level protein structure with a language model," *Science*, 2023, *379* (6637), 1123–1130.

Lipp, Moritz, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, Mike Hamburg, and Raoul Strackx, "Meltdown: reading kernel memory from user space," *Communications of the ACM*, 2020, *63* (6), 46–56.

Lipton, Zachary C. and Jacob Steinhardt, "Troubling Trends in Machine Learning Scholarship: Some ML papers suffer from flaws that could mislead the public and stymie future research.," *Queue*, 2019, *17* (1), 45–77.

Liu, Jian, Shiying Wu, and James V. Zidek, "On segmented multivariate regression," *Statistica Sinica*, 1997, *7* (2), 497–525. OpenAlex W2142183678.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv preprint arXiv:1907.11692 2019.

Lu, Shuai, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang et al., "CodeXGLUE: A Machine Learning

Benchmark Dataset for Code Understanding and Generation," arXiv preprint arXiv:2102.04664 2021.

Luccioni, Alexandra Sasha, Sylvain Viguier, and Anne-Laure Ligozat, "Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model," arXiv preprint arXiv:2211.02001 2022.

Malkov, Yu A. and D. A. Yashunin, "Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, *42* (4), 824–836.

Martín-Martín, Alberto, Mike Thelwall, Enrique Orduna-Malea, and Emilio Delgado López-Cózar, "Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations' COCI: a multidisciplinary comparison of coverage via citations," *Scientometrics*, 2021, *126*, 871–906.

Martín-Martín, Alberto, Mike Thelwall, Enrique Orduna-Malea, and Emilio Delgado López-Cózar, "Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations' COCI: a multidisciplinary comparison of coverage via citations," *Scientometrics*, 2021, *126* (1), 871–906.

Masanet, Eric, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey, "Recalibrating global data center energy-use estimates," *Science*, 2020, *367* (6481), 984–986.

Maslej, Nestor, Loredana Fattorini, Raymond Perrault, Vanessa Parli, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, and Jack Clark, "Artificial Intelligence Index Report 2024," arXiv preprint arXiv:2405.19522 2024.

_ , _ , _ , Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, Toby Walsh, Armin Hamrah, Lapo Santarlasci, Julia Betts Lotufo, Alexandra Rome, Andrew Shi, and Sukrut Oak, "Artificial Intelligence Index Report 2025," arXiv preprint arXiv:2504.07139 2025.

mathlib Community, The, "The lean mathematical library," in "Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs" ACM 2020, pp. 367–381.

Matsakis, Nicholas D. and Felix S. Klock, "The rust language," in "Proceedings of the 2014 ACM SIGAda annual conference on High integrity language technology" ACM 2014, pp. 103–104.

McInnes, Leland, John Healy, and James Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," arXiv preprint arXiv:1802.03426 2018.

_ , _ , and Steve Astels, "hdbscan: Hierarchical density based clustering," *The Journal of Open Source Software*, 2017, *2* (11), 205.

McKeown, Nick, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, 2008, *38* (2), 69–74.

McMahan, H. Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," arXiv preprint arXiv:1602.05629 2016.

Messeri, Lisa and M. J. Crockett, "Artificial intelligence and illusions of understanding in scientific research," *Nature*, 2024, *627* (8002), 49–58.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv preprint arXiv:1301.3781 2013.

Mildenhall, Ben, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in "Lecture Notes in Computer Science," Springer International Publishing, 2020, pp. 405–421.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis, "Human-level control through deep reinforcement learning," *Nature*, 2015, *518* (7540), 529–533.

Mongeon, Philippe and Adèle Paul-Hus, "The journal coverage of Web of Science and Scopus: a comparative analysis," *Scientometrics*, 2016, *106*, 213–228.

National Institute of Standards and Technology (US), "Module-lattice-based digital signature standard," Technical Report, National Institute of Standards and Technology (U.S.) 2024.

_ , "Module-lattice-based key-encapsulation mechanism standard," Technical Report, National Institute of Standards and Technology (U.S.) 2024.

_ , "Stateless hash-based digital signature standard," Technical Report, National Institute of Standards and Technology (U.S.) 2024.

Newey, Whitney K. and Kenneth D. West, "A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix," *Econometrica*, 1987, *55* (3), 703.

Newman, M. E. J. and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, 2004, *69* (2), 026113.

Noorden, Richard Van, "More than 10,000 research papers were retracted in 2023 — a new record," *Nature*, 2023, *624*, 479–481.

_ **and Jeffrey M. Perkel**, "AI and science: what 1,600 researchers think," *Nature*, 2023, *621* (7980), 672–675.

Nosek, Brian A., Charles R. Ebersole, Alexander C. DeHaven, and David T. Mellor, "The preregistration revolution," *Proceedings of the National Academy of Sciences*, 2018, *115* (11), 2600–2606.

OpenAI, "ChatGPT: Optimizing Language Models for Dialogue," OpenAI blog, 30 November 2022 (Internet Archive snapshot of 30 November 2022; live URL https://openai.com/index/chatgpt/) 2022.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Bohan Zhang, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason

Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wenting Zhan, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li, "OpenAI o1 System Card," arXiv preprint arXiv:2412.16720 2024.

__ **, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah**

Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter

Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph, "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774 2023.

OpenAlex, "Topics – Aboutness," OpenAlex documentation 2024.

_ , "Topics," OpenAlex documentation, https://help.openalex.org/data/topics/ 2025. Accessed 3 November 2026.

_ , "Works: attributes (Field-Weighted Citation Impact, `fwci`)," OpenAlex documentation, https://help.openalex.org/data/works/attributes/ 2026. Accessed 3 November 2026.

Ostendorff, Malte, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm, "Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings," in "Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing" Association for Computational Linguistics 2022, pp. 11670–11688.

Ott, Simon, Adriano Barbosa-Silva, Kathrin Blagec, Jan Brauner, and Matthias Samwald, "Mapping global dynamics of benchmark creation and saturation in artificial intelligence," *Nature Communications*, 2022, *13* (1).

Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe, "Training language models to follow instructions with human feedback," arXiv preprint arXiv:2203.02155 2022.

Page, Matthew J., David Moher, Patrick M. Bossuyt, Isabelle Boutron, Tammy C. Hoffmann, Cynthia D. Mulrow, Larissa Shamseer, Jennifer M. Tetzlaff, Elie A. Akl, Sue E. Brennan, Roger Chou, Julie Glanville, Jeremy M. Grimshaw, Asbjørn Hróbjartsson, Manoj M. Lalu, Tianjing Li, Elizabeth W. Loder, Evan Mayo-Wilson, Steve McDonald, Luke A. McGuinness, Lesley A. Stewart, James Thomas, Andrea C. Tricco, Vivian A. Welch, Penny Whiting, and Joanne E. McKenzie, "PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews," *BMJ*, 2021, *372*, n160.

Page, Matthew J, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, Roger Chou, Julie Glanville, Jeremy M Grimshaw, Asbjørn Hróbjartsson, Manoj M Lalu,

Tianjing Li, Elizabeth W Loder, Evan Mayo-Wilson, Steve McDonald, Luke A McGuinness, Lesley A Stewart, James Thomas, Andrea C Tricco, Vivian A Welch, Penny Whiting, and David Moher, "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews," *BMJ*, 2021, p. n71.

Park, Michael, Erin Leahey, and Russell J. Funk, "Papers and patents are becoming less disruptive over time," *Nature*, 2023, *613* (7942), 138–144.

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," arXiv preprint arXiv:1912.01703 2019.

Patterson, David, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean, "Carbon Emissions and Large Neural Network Training," arXiv preprint arXiv:2104.10350 2021.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research 12, 2825–2830; arXiv:1201.0490 2011. Verified through the arXiv record.

Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, "Deep Contextualized Word Representations," in "Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)" Association for Computational Linguistics 2018, pp. 2227–2237.

Petersen, Kai, Robert Feldt, Shahid Mujtaba, and Michael Mattsson, "Systematic Mapping Studies in Software Engineering," in "Electronic Workshops in Computing" BCS Learning & Development 2008.

_ , Sairam Vakkalanka, and Ludwik Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, 2015, *64*, 1–18.

Pineau, Joelle, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle, "Improving Reproducibility in

Machine Learning Research (A Report from the NeurIPS 2019 Reproducibility Program)," arXiv preprint arXiv:2003.12206 2020.

Preskill, John, "Quantum Computing in the NISQ era and beyond," *Quantum*, 2018, *2*, 79.

Priem, Jason, Heather Piwowar, and Richard Orr, "OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts," arXiv preprint arXiv:2205.01833 2022.

Raasveldt, Mark and Hannes Mühleisen, "DuckDB: an embeddable analytical database," in "Proceedings of the 2019 International Conference on Management of Data (SIGMOD '19)" 2019, pp. 1981–1984.

Radev, Dragomir R., Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara, "The ACL anthology network corpus," *Language Resources and Evaluation*, 2013, *47*, 919–944.

_ , _ , _ , and _ , "The ACL anthology network corpus," *Language Resources and Evaluation*, 2013, *47* (4), 919–944.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," arXiv preprint arXiv:2103.00020 2021.

_ , _ , Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," arXiv preprint arXiv:2212.04356 2022.

Rae, Jack W., Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young et al., "Scaling Language Models: Methods, Analysis & Insights from Training Gopher," arXiv preprint arXiv:2112.11446 2021.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," arXiv preprint arXiv:1910.10683 2019.

Rafols, Ismael and Martin Meyer, "Diversity and network coherence as indicators of interdisciplinarity: case studies in bionanoscience," *Scientometrics*, 2010, *82* (2), 263–287.

Raissi, M., P. Perdikaris, and G.E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, 2019, *378*, 686–707.

Rajbhandari, Samyam, Jeff Rasley, Olatunji Ruwase, and Yuxiong He, "ZeRO: Memory Optimizations Toward Training Trillion Parameter Models," arXiv preprint arXiv:1910.02054 2019.

Raji, Inioluwa Deborah, Emily M. Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna, "AI and the Everything in the Whole Wide World Benchmark," arXiv preprint arXiv:2111.15366 2021.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in "Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing" Association for Computational Linguistics 2016, pp. 2383–2392.

Ramesh, Aditya, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever, "Zero-Shot Text-to-Image Generation," arXiv preprint arXiv:2102.12092 2021.

_ , Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen, "Hierarchical Text-Conditional Image Generation with CLIP Latents," arXiv preprint arXiv:2204.06125 2022.

Rao, C. Radhakrishna, "Diversity and dissimilarity coefficients: a unified approach," *Theoretical Population Biology*, 1982, *21* (1), 24–43.

Reed, Scott, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg et al., "A Generalist Agent," arXiv preprint arXiv:2205.06175 2022.

Reimers, Nils and Iryna Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in "Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)" Association for Computational Linguistics 2019, pp. 3980–3990.

Rethlefsen, Melissa L., Shona Kirtley, Siw Waffenschmidt, Ana Patricia Ayala, David Moher, Matthew J. Page, Jonathan B. Koffel, PRISMA-S Group, Heather Blunt, Tara Brigham, Steven Chang, Justin Clark, Aislinn Conway, Rachel Couban, Shelley de Kock, Kelly Farrah, Paul Fehrmann, Margaret Foster, Susan A. Fowler, Julie Glanville, Elizabeth Harris, Lilian Hoffecker, Jaana Isojarvi, David Kaunelis, Hans Ket, Paul Levay, Jennifer Lyon, Jessie McGowan, M. Hassan Murad, Joey Nicholson, Virginia Pannabecker, Robin Paynter, Rachel Pinotti, Amanda Ross-White, Margaret Sampson, Tracy Shields, Adrienne Stevens, Anthea Sutton, Elizabeth Weinfurter, Kath Wright, and Sarah Young, "PRISMA-S:

an extension to the PRISMA Statement for Reporting Literature Searches in Systematic Reviews," *Systematic Reviews*, 2021, *10* (1).

Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in "2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)" IEEE 2022, pp. 10674–10685.

Rous, Bernard, "Major update to ACM's Computing Classification System," *Communications of the ACM*, 2012, *55* (11), 12–12.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, 2015, *115* (3), 211–252.

Saharia, Chitwan, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi, "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding," arXiv preprint arXiv:2205.11487 2022.

Scheidsteger, Thomas and Robin Haunschild, "Which of the metadata with relevance for bibliometrics are the same and which are different when switching from Microsoft Academic Graph to OpenAlex?," *Profesional de la información*, 2023.

Schick, Timo, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom, "Toolformer: Language Models Can Teach Themselves to Use Tools," arXiv preprint arXiv:2302.04761 2023.

Schwartz, Roy, Jesse Dodge, Noah A. Smith, and Oren Etzioni, "Green AI," *Communications of the ACM*, 2020, *63* (12), 54–63.

Schwarz, Gideon, "Estimating the dimension of a model," *The Annals of Statistics*, 1978, *6* (2), 461–464.

Senior, Andrew W., Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W. R. Nelson, Alex Bridgland et al., "Improved protein structure prediction using potentials from deep learning," *Nature*, 2020, *577* (7792), 706–710.

Sevilla, Jaime, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos, "Compute Trends Across Three Eras of Machine Learning," in "2022 International Joint Conference on Neural Networks (IJCNN)" IEEE 2022, pp. 1–8.

Shi, Weisong, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, 2016, *3* (5), 637–646.

Shneiderman, Ben, "Human-Centered Artificial Intelligence: Reliable, Safe & Trustworthy," *International Journal of Human–Computer Interaction*, 2020, *36* (6), 495–504.

Shoeybi, Mohammad, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro, "Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism," arXiv preprint arXiv:1909.08053 2019.

Shor, Peter W., "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Journal on Computing*, 1997, *26* (5), 1484–1509.

Silver, David, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, 2016, *529* (7587), 484–489.

_ , Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis, "Mastering the game of Go without human knowledge," *Nature*, 2017, *550* (7676), 354–359.

_ , Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, 2018, *362* (6419), 1140–1144.

Singh, Amanpreet, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman, "SciRepEval: A Multi-Format Benchmark for Scientific Document Representations," in "Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing" Association for Computational Linguistics 2023, pp. 5548–5566.

Sjögårde, Peter and Per Ahlgren, "Granularity of algorithmically constructed publication-level classifications of research publications: Identification of topics," *Journal of Informetrics*, 2018, *12* (1), 133–152.

Sohl-Dickstein, Jascha, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli, "Deep Unsupervised Learning using Nonequilibrium Thermodynamics," arXiv preprint arXiv:1503.03585 2015.

Song, Yang, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole, "Score-Based Generative Modeling through Stochastic Differential Equations," arXiv preprint arXiv:2011.13456 2020.

Srivastava, Aarohi, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak,

Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël

Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu, "Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models," arXiv preprint arXiv:2206.04615 2022.

Stiennon, Nisan, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano, "Learning to summarize from human feedback," arXiv preprint arXiv:2009.01325 2020.

Stirling, Andy, "A general framework for analysing diversity in science, technology and society," *Journal of The Royal Society Interface*, 2007, *4* (15), 707–719.

Strubell, Emma, Ananya Ganesh, and Andrew McCallum, "Energy and Policy Considerations for Deep Learning in NLP," in "Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics" Association for Computational Linguistics 2019, pp. 3645–3650.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le, "Sequence to Sequence Learning with Neural Networks," arXiv preprint arXiv:1409.3215 2014.

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," arXiv preprint arXiv:1312.6199 2013.

Team, Gemini, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk

Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso

Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kępa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabi-

novitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Roopali Vij, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz

Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy,

Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki,

Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals, "Gemini: A Family of Highly Capable Multimodal Models," arXiv preprint arXiv:2312.11805 2023.

Tewari, A., O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner et al., "State of the Art on Neural Rendering," *Computer Graphics Forum*, 2020, *39* (2), 701–727.

Thompson, Neil C., Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso, "The Computational Limits of Deep Learning," arXiv preprint arXiv:2007.05558 2020.

Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas

Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom, "Llama 2: Open Foundation and Fine-Tuned Chat Models," arXiv preprint arXiv:2307.09288 2023.

_ , Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample, "LLaMA: Open and Efficient Foundation Language Models," arXiv preprint arXiv:2302.13971 2023.

Traag, V. A., L. Waltman, and N. J. van Eck, "From Louvain to Leiden: guaranteeing well-connected communities," *Scientific Reports*, 2019, *9* (1).

Tricco, Andrea C., Erin Lillie, Wasifa Zarin, Kelly K. O'Brien, Heather Colquhoun, Danielle Levac, David Moher, Micah D.J. Peters, Tanya Horsley, Laura Weeks, Susanne Hempel, Elie A. Akl, Christine Chang, Jessie McGowan, Lesley Stewart, Lisa Hartling, Adrian Aldcroft, Michael G. Wilson, Chantelle Garritty, Simon Lewin, Christina M. Godfrey, Marilyn T. Macdonald, Etienne V. Langlois, Karla Soares-Weiser, Jo Moriarty, Tammy Clifford, Özge Tunçalp, and Sharon E. Straus, "PRISMA Extension for Scoping Reviews (PRISMA-ScR): Checklist and Explanation," *Annals of Internal Medicine*, 2018, *169* (7), 467–473.

Truong, Charles, Laurent Oudre, and Nicolas Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, 2020, *167*, 107299.

Uzzi, Brian, Satyam Mukherjee, Michael Stringer, and Ben Jones, "Atypical combinations and scientific impact," *Science*, 2013, *342* (6157), 468–472.

Vardi, Moshe Y., "Conferences vs. journals in computing research," *Communications of the ACM*, 2009, *52* (5), 5–5.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention Is All You Need," arXiv preprint arXiv:1706.03762 2017.

Vinyals, Oriol, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, 2019, *575* (7782), 350–354.

Visser, Martijn, Nees Jan van Eck, and Ludo Waltman, "Large-scale comparison of bibliographic data sources: Scopus, Web of Science, Dimensions, Crossref, and Microsoft Academic," *Quantitative Science Studies*, 2021, *2* (1), 20–41.

Waltman, Ludo, "A review of the literature on citation impact indicators," *Journal of Informetrics*, 2016, *10* (2), 365–391.

_ **and Nees Jan van Eck**, "A new methodology for constructing a publication-level classification system of science," *Journal of the American Society for Information Science and Technology*, 2012, *63* (12), 2378–2392.

_ **and** _ , "Field Normalization of Scientometric Indicators," in "Springer Handbook of Science and Technology Indicators," Springer, 2019, pp. 281–300.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," arXiv preprint arXiv:1804.07461 2018.

_ **, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman**, "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems," arXiv preprint arXiv:1905.00537 2019.

Wang, Dashun and Albert-László Barabási, *The Science of Science*, Cambridge University Press, 2021.

Wang, Hanchen, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, Carla P. Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, Tie-Yan Liu, Arjun Manrai, Debora Marks, Bharath Ramsundar, Le Song, Jimeng Sun, Jian Tang, Petar Veličković, Max Welling, Linfeng Zhang, Connor W. Coley, Yoshua Bengio, and Marinka Zitnik, "Scientific discovery in the age of artificial intelligence," *Nature*, 2023, *620* (7972), 47–60.

Wang, Jian, "Citation time window choice for research impact evaluation," *Scientometrics*, 2013, *94*, 851–872.

_ , Reinhilde Veugelers, and Paula Stephan, "Bias against novelty in science: A cautionary tale for users of bibliometric indicators," *Research Policy*, 2017, *46* (8), 1416–1436.

Wang, Kuansan, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia, "Microsoft Academic Graph: When experts are not enough," *Quantitative Science Studies*, 2020, *1* (1), 396–413.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," arXiv preprint arXiv:2201.11903 2022.

_ , Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus, "Emergent Abilities of Large Language Models," arXiv preprint arXiv:2206.07682 2022.

Williams, Virginia Vassilevska, "Hardness of Easy Problems: Basing Hardness on Popular Conjectures such as the Strong Exponential Time Hypothesis," in "10th International Symposium on Parameterized and Exact Computation (IPEC 2015)" Schloss Dagstuhl – Leibniz-Zentrum für Informatik 2015, pp. 17–29.

WILLIAMS, VIRGINIA VASSILEVSKA, "ON SOME FINE-GRAINED QUESTIONS IN ALGORITHMS AND COMPLEXITY," in "Proceedings of the International Congress of Mathematicians (ICM 2018)" WORLD SCIENTIFIC 2019, pp. 3447–3487.

Wing, Jeannette M., "Computational thinking," *Communications of the ACM*, 2006, *49* (3), 33–35.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush, "Transformers: State-of-the-Art Natural Language Processing," in "Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations" Association for Computational Linguistics 2020, pp. 38–45.

Yang, Qian, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman, "Re-examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design," in "Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems" ACM 2020, pp. 1–13.

Yao, Shunyu, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao, "ReAct: Synergizing Reasoning and Acting in Language Models," arXiv preprint arXiv:2210.03629 2022.

Yao, Yi-Ching, "Estimating the number of change-points via Schwarz' criterion," *Statistics & Probability Letters*, 1988, *6* (3), 181–189.

Zaharia, Matei, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica, "Apache Spark: a unified engine for big data processing," *Communications of the ACM*, 2016, *59* (11), 56–65.

Zhang, Chiyuan, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, 2021, *64* (3), 107–115.

Zhang, Lin, Ronald Rousseau, and Wolfgang Glänzel, "Diversity of references as an indicator of the interdisciplinarity of journals: taking similarity between subject fields into account," *Journal of the Association for Information Science and Technology*, 2016, *67* (5), 1257–1265.

Zhang, Min-Ling and Zhi-Hua Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, 2014, *26* (8), 1819–1837.

Zhang, Susan, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin et al., "OPT: Open Pre-trained Transformer Language Models," arXiv preprint arXiv:2205.01068 2022.

Appendices

A. The Ontology of Computer-Science Research

This appendix documents the classification scheme that assigns every eligible work of the corpus to one or more of the forty research categories listed in the mandate (Section 17 describes the scoring rule and its validation design). The scheme is an *extensible hierarchical ontology* rather than a flat keyword list: forty leaf categories are grouped into eight areas, a small set of parent–child edges records where a leaf specialises another leaf, and a denser set of "related" edges records thematic adjacency. The mapping from bibliographic signals to leaves is data, not

prose. It lives in one Python module per area (`taxonomy/src/*.py`), is validated and serialised by `taxonomy/build_ontology.py` into the machine-readable file `taxonomy/ontology.yaml` (version 1.0, built on 2026-11-03 from the OpenAlex topic export of 2026-10-14 and the DBLP stream table of 2018–2026) and into an audit table of OpenAlex topics (`taxonomy/topic_map.csv`). Everything stated below is derived from those files; the tables were regenerated from the YAML file rather than transcribed, and the counts they contain are properties of the ontology itself (how many topics, venues and patterns point to a category), not statistics of the corpus. The agreement of the rules with independent raters on the stratified validation sample is reported in Part IX and is deliberately not anticipated here.

A.1. Structure: areas, leaves and edges

The ontology has three levels. The eight areas of Table 4 are an organisational device for the manuscript (they correspond to the disciplinary reviews of Parts V–VII and to the area-level flow statistics of Part VIII); classification operates on leaves only, and every leaf belongs to exactly one area. The forty leaves are numbered as in the mandate (Table 5). Seven `parent_of`/`child_of` edges record genuine specialisation: `foundation_models` → `llm` and `multimodal_ai`; `programming_languages` → `compilers`; `distributed_computing` → `cloud_computing` and `edge_computing`; `computer_architecture` → `ai_accelerators`; `privacy_preserving_computation` → `federated_learning`. These edges have one operational consequence, the tie-breaking rule for the primary category (a child wins over its parent at equal score), and one interpretive consequence: a work carrying both labels is not double-counted as evidence of convergence between distinct fields, since the two labels are nested by construction. The `related` edges (236 directed entries in the YAML file, 144 distinct unordered pairs) carry no weight in classification; they document which categories the ontology's author expected to share works, and they are used in Part IX only to separate expected from unexpected co-classification.

Two features of the grouping deserve comment because they are choices, not facts. First, `ai_safety` is placed under the AI area rather than under security and privacy, although adversarial robustness and backdoors are scored in both `ai_safety` and `cybersecurity`; the ontology treats safety as a property of learning systems, with security as the adjacent field. Second, `computer_graphics` sits in the perception area next to vision and HCI rather than in a visual-computing area of its own, and `federated_learning` sits under privacy rather than under distributed computing, which reflects the definitional emphasis of the leaf (training without centralising raw data) rather than its algorithmic kinship. Readers who would group the leaves differently can do so from the leaf-level tables of Part IX without re-running anything, because areas are never used in scoring.

Table 4. The eight areas and their leaves (`ontology.yaml`, `areas`).

Area id	Name	Leaves	Leaf identifiers
`ai_ml`	Artificial Intelligence and Machine Learning	9	`llm`, `foundation_models`, `multimodal_ai`, `generative_ai`, `ai_agents`, `reinforcement_learning`, `ml_theory`, `deep_learning_architectures`, `ai_safety`
`perception_interaction`	Perception, Language, Robotics and Interaction	6	`computer_vision`, `nlp`, `speech_audio`, `robotics`, `hci`, `computer_graphics`
`software_pl`	Software, Programming Languages and Formal Methods	4	`software_engineering`, `programming_languages`, `compilers`, `formal_verification`
`theory`	Theory of Computation	2	`algorithms`, `computational_complexity`
`data_distributed`	Data, Information and Distributed Systems	6	`database_systems`, `information_retrieval`, `distributed_computing`, `cloud_computing`, `edge_computing`, `computer_networks`
`security_privacy`	Security, Privacy and Trust	5	`cybersecurity`, `applied_cryptography`, `blockchain`, `privacy_preserving_computation`, `federated_learning`
`hardware_emerging`	Hardware, Architecture and Emerging Computing	5	`computer_architecture`, `ai_accelerators`, `operating_systems`, `quantum_computing`, `sustainable_computing`
`computational_science`	Computational Science and Discovery	3	`scientific_computing`, `computational_biology`, `autonomous_scientific_discovery`

A.2. The forty categories

Table 5 reproduces the definition of each leaf exactly as it is given to the classifier's author and to the validation raters (the raters receive these definitions and nothing else about the rules). Each definition states the object of the category, lists its typical sub-topics and names the neighbouring category to which excluded work belongs; the latter clause is what makes the definitions a partition rather than a list of keywords, and it is the part most often invoked in the ambiguity decisions of Section A.7. Identifiers are those of the YAML file; numbers are those of the mandate.

Table 5. The forty leaf categories: identifier, area and definition (`ontology.yaml`, `categories[].definition`). Parent/child edges are given under the identifier.

No.	Category	Definition
1	**Large Language Models** `llm` area `ai_ml`; child of `foundation_models`	Research whose object is a transformer-based language model trained at scale: pre-training, instruction tuning, prompting and in-context learning, reasoning, retrieval augmentation, efficient inference, evaluation and benchmarks of LLMs. Excludes classical NLP tasks solved with small task-specific models (nlp) and non-language foundation models (foundation_models).
2	**Foundation Models** `foundation_models` area `ai_ml`; parent of `llm`, `multimodal_ai`	Large pre-trained models intended for adaptation to many downstream tasks in any modality: scaling laws, pre-training objectives, self-supervised representation learning at scale, transfer, fine-tuning and adapter methods, model releases and their evaluation. Language-specific work is llm (a child); vision-language work is multimodal_ai (a child).
3	**Multimodal AI** `multimodal_ai` area `ai_ml`; child of `foundation_models`	Models and systems that jointly process two or more modalities (vision-language, audio-visual, video-text, 3D-language, sensors with language): vision-language models and multimodal LLMs, captioning, visual question answering, cross-modal retrieval and grounding. Unimodal vision is computer_vision; pure speech is speech_audio.
4	**Generative AI** `generative_ai` area `ai_ml`	Models that synthesise new content (images, video, audio, 3D, text, code, molecules) and their training, sampling, evaluation, detection and misuse: GANs, VAEs, diffusion and flow models, autoregressive generators, text-to-X synthesis, deepfakes and AI-generated-content detection. LLM text generation as such is llm unless the contribution concerns the generative model or its outputs' authenticity.
5	**AI Agents and Autonomous Systems** `ai_agents` area `ai_ml`	Software agents that perceive, decide and act with some autonomy: LLM-based agents and tool use, multi-agent systems, planning and automated reasoning for acting, agent architectures, negotiation, agentic workflows and benchmarks, embodied agents in simulated environments. Physical robots are robotics; RL algorithms as such are reinforcement_learning.

continued on next page

Table 5. The forty leaf categories (continued).

No.	Category	Definition
6	**Reinforcement Learning** `reinforcement_learning` area `ai_ml`	Learning to act from reward signals: value- and policy-based methods, model-based RL, offline and inverse RL, multi-agent RL, bandits, exploration, reward design, RL from human feedback, and RL theory. Applications to robot control are shared with robotics.
7	**Machine Learning Theory** `ml_theory` area `ai_ml`	Statistical and computational learning theory and principled ML methodology: generalisation and sample complexity, optimisation theory for learning, kernel and Bayesian methods, probabilistic models, causal inference, online learning, PAC learning, interpretability and fairness as formal problems. Architecture engineering is deep_learning_architectures.
8	**Deep Learning Architectures** `deep_learning_architectures` area `ai_ml`	Design, training and compression of neural network architectures as such: CNNs, RNNs, transformers and attention, graph neural networks, state-space models, normalisation and optimisers in practice, neural architecture search, pruning, quantisation and distillation, neuromorphic and spiking networks as models. Task-specific results go to the application category; hardware goes to ai_accelerators.
9	**Computer Vision** `computer_vision` area `perception_interaction`	Extracting information from images, video and 3D data: recognition, detection, segmentation, tracking, pose, depth and 3D reconstruction, image restoration and enhancement, medical and remote-sensing image analysis, visual representation learning. Image synthesis is generative_ai; vision-language is multimodal_ai; rendering is computer_graphics.
10	**Natural Language Processing** `nlp` area `perception_interaction`	Computational processing of human language: machine translation, parsing and tagging, information extraction, sentiment and text classification, summarisation, question answering, dialogue, text mining, computational linguistics, low-resource and multilingual NLP, NLP evaluation and resources. LLM-centred work is llm (also classified here when the task framing dominates).
11	**Speech and Audio Computing** `speech_audio` area `perception_interaction`	Processing and generation of speech, music and general audio: speech recognition and synthesis, speaker and language recognition, enhancement and separation, spoken language understanding, music information retrieval and generation, audio event detection, neural audio codecs. Text-only dialogue is nlp.
12	**Robotics** `robotics` area `perception_interaction`	Machines that sense and act in the physical world: manipulation, locomotion, motion and path planning, SLAM and localisation, robot learning, human-robot interaction, autonomous vehicles and drones as robotic systems, soft and swarm robotics, teleoperation. RL algorithms are reinforcement_learning; robotic process automation and surgical-outcome studies are excluded.

continued on next page

Table 5. The forty leaf categories (continued).

No.	Category	Definition
13	**Human-Computer Interaction** `hci` area `perception_interaction`	Design, evaluation and understanding of interactive systems and their users: user studies and usability, interaction techniques, VR/AR/XR, accessibility and assistive technology, CSCW and social computing, human-AI interaction, ubiquitous and wearable computing, affective computing. Rendering is computer_graphics; physical robots are robotics.
14	**Software Engineering** `software_engineering` area `software_pl`	How software is specified, built, tested, maintained and studied: requirements, architecture, testing and debugging, program repair, mining software repositories, empirical studies of developers, DevOps, model-driven engineering, open-source ecosystems, and AI-assisted software development. Language design is programming_languages; proofs of correctness are formal_verification.
15	**Programming Languages** `programming_languages` area `software_pl`; parent of `compilers`	Design, semantics, type systems and implementation principles of programming languages: type theory, operational and denotational semantics, functional and logic programming, effects, concurrency models, memory safety, domain-specific languages, program analysis as a language problem. Code generation and optimisation pipelines are compilers (a child); machine-checked correctness is formal_verification.
16	**Compilers** `compilers` area `software_pl`; child of `programming_languages`	Translation and optimisation of programs for execution: compiler construction, intermediate representations, optimisation passes, code generation, JIT compilation, auto-vectorisation and parallelisation, polyhedral methods, high-level synthesis, and compilers for machine-learning workloads and accelerators. A child of programming_languages.
17	**Formal Verification** `formal_verification` area `software_pl`	Mathematically rigorous establishment of system properties: model checking, theorem proving and proof assistants, SAT/SMT solving and automated reasoning, deductive and runtime verification, verified software and hardware, formal semantics as a verification tool, verification of neural networks, smart contracts and protocols, automata and temporal logics. Language semantics as design is programming_languages.
18	**Algorithms and Data Structures** `algorithms` area `theory`	Design and analysis of algorithms and data structures: graph, string, geometric, approximation, randomized, online, streaming, parallel and parameterized algorithms, combinatorial optimisation and metaheuristics, algorithmic game theory, algorithm engineering, and the combinatorics that underpins them. Lower bounds and hardness as such are computational_complexity; numerical algorithms are scientific_computing.
19	**Computational Complexity** `computational_complexity` area `theory`	The intrinsic difficulty of computational problems: complexity classes and their separations, hardness of approximation, circuit, proof, communication and query complexity, fine-grained and parameterized complexity, pseudorandomness and derandomization, computability and automata theory, algebraic complexity, and quantum complexity. Algorithm design is algorithms; cryptographic hardness assumptions are applied_cryptography.

continued on next page

Table 5. The forty leaf categories (continued).

No.	Category	Definition
20	**Database Systems** `database_systems` area `data_distributed`	Storage, querying and management of structured and semi-structured data: query processing and optimisation, transactions and concurrency control, indexing and storage engines, data integration, cleaning and provenance, big-data processing frameworks, data mining as a data-management problem, learned database components, vector and graph databases. Ranking of unstructured documents is information_retrieval.
21	**Information Retrieval** `information_retrieval` area `data_distributed`	Finding and ranking relevant information for a user need: search engines and ranking models, dense and sparse retrieval, recommender systems, query understanding, evaluation and test collections, web search and mining, scholarly and digital-library retrieval, retrieval components of RAG systems. Structured query processing is database_systems; text understanding tasks are nlp.
22	**Distributed Computing** `distributed_computing` area `data_distributed`; parent of `cloud_computing`, `edge_computing`	Computation across multiple processors or machines: distributed algorithms and consensus, fault tolerance and replication, parallel programming and HPC systems software, cluster and grid resource management, distributed storage and middleware, distributed and parallel ML training systems. Cloud service models are cloud_computing and edge placement is edge_computing (both children).
23	**Cloud Computing** `cloud_computing` area `data_distributed`; child of `distributed_computing`	Computing delivered as elastic, multi-tenant, on-demand services: IaaS/PaaS/SaaS and serverless, virtualisation and containers, resource provisioning and auto-scaling, SLAs and pricing, cloud-native architectures, data-centre resource management, cloud security and migration. A child of distributed_computing; placement at the network edge is edge_computing.
24	**Edge Computing** `edge_computing` area `data_distributed`; child of `distributed_computing`	Computation placed near data sources and users: edge/fog/MEC architectures, computation offloading, on-device and TinyML inference, edge-cloud collaboration and split computing, IoT platforms as computing substrates, latency-critical edge services, edge caching and orchestration. Radio access and protocols are computer_networks; model compression as such is deep_learning_architectures or ai_accelerators.
25	**Computer Networks** `computer_networks` area `data_distributed`	Design, protocols, measurement and performance of communication networks: Internet architecture and transport, routing, wireless and mobile networks, 5G/6G and software-defined networking, network management, sensor and vehicular networks, content delivery and caching, network coding and information theory applied to networking. Physical-layer signal processing is included only when framed as networking.

continued on next page

Table 5. The forty leaf categories (continued).

No.	Category	Definition
26	**Cybersecurity** `cybersecurity` area `security_privacy`	Protecting computing systems, networks, software and users from attack: vulnerability discovery and exploitation, malware and intrusion detection, authentication and access control, system and hardware security, web and mobile security, digital forensics, security of AI systems, security operations and human factors of security. Cryptographic primitives and protocols are applied_cryptography; data-protection techniques are privacy_preserving_computation.
27	**Applied Cryptography** `applied_cryptography` area `security_privacy`	Design, analysis, implementation and deployment of cryptographic primitives and protocols: symmetric and public-key schemes, hash functions, signatures, key exchange, post-quantum cryptography, zero-knowledge proofs, secure multi-party computation and homomorphic encryption as primitives, cryptanalysis, side-channel-resistant implementations, cryptographic hardware, provable security, quantum key distribution. Deployment of these primitives for data protection is privacy_preserving_computation; ledgers are blockchain.
28	**Blockchain Systems** `blockchain` area `security_privacy`	Distributed ledgers and the systems built on them: consensus for permissioned and permissionless chains, smart contracts and their security, scalability (sharding, layer-2, rollups), cryptocurrencies and DeFi mechanisms, blockchain applications and governance, measurement of blockchain ecosystems. Generic BFT consensus theory is shared with distributed_computing; primitives are applied_cryptography.
29	**Computer Architecture** `computer_architecture` area `hardware_emerging`; parent of `ai_accelerators`	Organisation and design of processors, memory systems and digital hardware: microarchitecture, caches and memory hierarchies, interconnects and networks-on-chip, FPGAs and reconfigurable computing, embedded and real-time systems, hardware/software co-design, electronic design automation, VLSI and low-power design, processing-in-memory. Accelerators specialised for ML are ai_accelerators (a child); kernels are operating_systems.
30	**Accelerators and AI Hardware** `ai_accelerators` area `hardware_emerging`; child of `computer_architecture`	Hardware and hardware-aware software for efficient machine-learning computation: GPU/TPU/NPU design and programming, ASIC and FPGA DNN accelerators, neuromorphic and spiking hardware, in-memory and analog/photonic computing for AI, quantisation and sparsity exploited by hardware, efficient LLM inference and serving systems, ML compilers' hardware mapping, benchmarks such as MLPerf. A child of computer_architecture.
31	**Operating Systems** `operating_systems` area `hardware_emerging`	System software that manages hardware and provides execution abstractions: kernels and drivers, scheduling, memory management, file and storage systems, virtualisation and containers, OS security and isolation, I/O and networking stacks, real-time and embedded OS, systems support for new hardware (persistent memory, CXL, SmartNICs). Distributed coordination is distributed_computing; hardware design is computer_architecture.

continued on next page

Table 5. The forty leaf categories (continued).

No.	Category	Definition
32	**Quantum Computing** `quantum_computing` area `hardware_emerging`	Computation and information processing exploiting quantum mechanics: quantum algorithms and complexity, quantum error correction and fault tolerance, qubit hardware platforms and control, quantum software, compilers and simulators, variational and NISQ algorithms, quantum machine learning, quantum communication and key distribution, quantum networking. Quantum physics without a computational or information-processing framing is out of scope; post-quantum cryptography is applied_cryptography.
33	**Scientific Computing** `scientific_computing` area `computational_science`	Numerical and computational methods and software for mathematical models of science and engineering: numerical linear algebra, discretisation of differential equations, solvers and preconditioners, simulation frameworks, uncertainty quantification, reduced-order and surrogate models, scientific machine learning (PINNs, neural operators), high-performance scientific software and performance portability, mathematical software, computational physics/chemistry/engineering tooling. Domain biology is computational_biology; parallel systems software is distributed_computing.
34	**Computational Biology** `computational_biology` area `computational_science`	Computational methods for biological and biomedical data: bioinformatics and genomics, protein structure and design, molecular modelling and computational drug discovery, single-cell and multi-omics analysis, systems and synthetic biology, biomedical text mining, computational neuroscience, biomedical and health informatics methods, DNA/molecular computing. Medical image analysis is shared with computer_vision; numerical methods as such are scientific_computing.
35	**Computer Graphics** `computer_graphics` area `perception_interaction`	Synthesis and manipulation of visual content from models: rendering, geometry processing and modelling, animation and simulation for graphics, neural rendering and 3D representations, visualisation, procedural content, game and VR engines. Image understanding is computer_vision; pixel-space generative models are generative_ai; user studies of VR are hci.
36	**Privacy-Preserving Computation** `privacy_preserving_computation` area `security_privacy`; parent of `federated_learning`	Computing on, releasing or sharing data while limiting what is learned about individuals: differential privacy, homomorphic encryption, secure multi-party computation and TEEs applied to data analysis and ML, anonymisation and synthetic data, private information retrieval, privacy attacks on models and their defences, machine unlearning, privacy engineering and regulation (GDPR). Federated learning is a child; cryptographic constructions as such are applied_cryptography.

continued on next page

Table 5. The forty leaf categories (continued).

No.	Category	Definition
37	**Federated Learning** `federated_learning` area `security_privacy`; child of `privacy_preserving_computation`	Training models across multiple clients or institutions without centralising their raw data: federated averaging and optimisation under heterogeneity, communication efficiency, client selection, personalisation, secure aggregation and privacy of federated systems, robustness to malicious clients, vertical and cross-silo federated learning, federated fine-tuning of foundation models, decentralised/gossip learning. A child of privacy_preserving_computation; also linked to distributed_computing and edge_computing.
38	**AI Safety and Alignment** `ai_safety` area `ai_ml`	Making AI systems safe, aligned, robust, fair, interpretable and governable: alignment and preference learning framed as a safety problem, red-teaming and jailbreaks, adversarial robustness and backdoors, hallucination and truthfulness, bias and fairness, mechanistic interpretability, evaluation of dangerous capabilities, AI governance and regulation. Generic system security is cybersecurity; XAI as a statistical method only is ml_theory.
39	**Sustainable Computing** `sustainable_computing` area `hardware_emerging`	The energy, carbon, water and material footprint of computing and how to reduce it: energy-efficient and carbon-aware systems, data-centre energy and cooling, Green AI and the cost of training and inference, life-cycle and embodied carbon of hardware, green software engineering and networking, energy harvesting and intermittent computing, computing for sustainability when the object studied is the computing system. Generic low-power circuit design is computer_architecture; energy-efficient communication is computer_networks.
40	**Autonomous Scientific Discovery** `autonomous_scientific_discovery` area `computational_science`	AI and automation systems that perform or substantially drive the scientific process: AI scientist and research-agent systems, automated hypothesis generation and literature synthesis, self-driving laboratories and closed-loop experimentation, AI-driven discovery of materials, molecules, equations, algorithms or theorems, autonomous experimental design, benchmarks for research agents, and the epistemology and evaluation of machine-generated science. Using ML inside a scientific pipeline without autonomy is scientific_computing or computational_biology; LLM agents in general are ai_agents.

A.3. Signal types and the assignment rule

Every leaf lists weighted signals of five types, and the five types are the only evidence the classifier ever sees. (i) *OpenAlex topics*: the up-to-three topic identifiers that OpenAlex attaches to a work, drawn from any of the 26 OpenAlex fields and not only from field 17; a topic may appear under several leaves with different weights. (ii) *OpenAlex subfields*: a coarse fallback used by 21 leaf–subfield entries over eleven subfields (the ten field-17 subfields and subfield 1312, Molecular Biology, for computational biology), always with a small weight. (iii) *arXiv cate-*

gories: the primary category and cross-lists of the linked arXiv record, 162 entries over 49 distinct categories, namely `cs.AI`, `cs.AR`, `cs.CC`, `cs.CE`, `cs.CG`, `cs.CL`, `cs.CR`, `cs.CV`, `cs.CY`, `cs.DB`, `cs.DC`, `cs.DL`, `cs.DM`, `cs.DS`, `cs.ET`, `cs.FL`, `cs.GR`, `cs.GT`, `cs.HC`, `cs.IR`, `cs.IT`, `cs.LG`, `cs.LO`, `cs.MA`, `cs.MM`, `cs.MS`, `cs.NA`, `cs.NE`, `cs.NI`, `cs.OS`, `cs.PF`, `cs.PL`, `cs.RO`, `cs.SD`, `cs.SE`, `cs.SI`, `cs.SY`, `eess.AS`, `eess.IV`, `eess.SP`, `eess.SY`, `math.IT`, `math.OC`, `math.ST`, `physics.comp-ph`, `q-bio.GN`, `q-bio.QM`, `quant-ph`, `stat.ML`. (iv) *DBLP streams*: the venue stream (journal or conference series, the prefix of the DBLP key) of the linked DBLP record, 1,320 entries over 771 distinct streams. (v) *Keyword patterns*: 273 regular expressions matched, in the present version of the ontology without exception, on the concatenation of title and abstract (the YAML schema allows title-only and abstract-only patterns, but none is used); 24 exclusion patterns veto a leaf when they match the same text.

The rule itself is the one written out in equations (21) and (22): within each signal type the weights of the matched signals are summed and capped at 1.0; the capped values of the five types are summed; the leaf is assigned when the total reaches the threshold $\theta = 1.0$ and no exclusion pattern matches; the primary leaf is the argmax, ties going to the child before the parent and then to the lower category number. The weights are therefore to be read on a scale where 1.0 means "this signal alone suffices". Table 6 reports, for each leaf, how many signals of each type it lists and how many of them carry weight 1.0; Table 7 aggregates the same counts by area. Three regularities are visible. First, every one of the forty leaves has at least one keyword pattern of weight 1.0 (106 of the 273 patterns are at 1.0, the rest spread between 0.2 and 0.9), so that a title or abstract naming the object of the category assigns it without metadata support; the build script refuses to serialise an ontology in which this is false for the thirteen categories that the design singles out (the six AI categories that OpenAlex topics do not separate and seven small categories without a topic or venue of their own), and in version 1.0 the property happens to hold for all forty. Second, the number of topic signals varies by an order of magnitude across leaves, from 4 for `compilers`, `reinforcement_learning` and `federated_learning` to 45 for `computer_networks` and 44 for `computer_vision`; this mirrors the granularity of the OpenAlex topic model, which has many clusters for networking and vision and none for federated learning, and not the importance of the categories. Third, venue signals are dense (291 of the 1,320 stream entries are at weight 1.0, almost all for single-purpose venues) but concentrated on the systems, networking and theory leaves, where the DBLP venue structure is informative, and sparse for the post-2022 AI leaves, where the same few venues host everything.

Table 6. Signals per category in `ontology.yaml`: number of weighted OpenAlex topics (T), of which with weight 1.0 (T_1); OpenAlex subfields (F); arXiv categories (X); DBLP streams (D), of which with weight 1.0 (D_1); keyword patterns (K), of which with weight 1.0 (K_1); exclusion patterns (Z); related edges (R). Computed from the YAML file.

No.	Category	T	T_1	F	X	D	D_1	K	K_1	Z	R
1	`llm`	6	1	1	3	12	0	12	3	0	5
2	`foundation_models`	5	0	1	4	8	0	10	2	0	5
3	`multimodal_ai`	5	1	1	4	15	0	9	4	1	5
4	`generative_ai`	5	1	0	6	9	0	11	4	2	6
5	`ai_agents`	5	1	1	2	12	3	9	5	1	5
6	`reinforcement_learning`	4	1	0	5	8	0	9	3	1	5
7	`ml_theory`	20	1	1	4	51	2	11	1	0	6
8	`deep_learning_architectures`	11	2	1	4	21	0	12	1	1	6
9	`computer_vision`	44	9	1	2	58	13	10	3	0	6
10	`nlp`	22	4	1	2	29	13	8	2	0	5
11	`speech_audio`	10	3	1	4	16	5	5	3	0	4
12	`robotics`	23	6	0	3	41	27	4	1	2	5
13	`hci`	24	4	1	3	44	22	10	1	0	6
14	`software_engineering`	20	7	1	1	39	18	5	3	0	6
15	`programming_languages`	6	1	0	3	22	6	6	3	0	4
16	`compilers`	4	0	0	5	25	2	5	3	0	5
17	`formal_verification`	10	1	0	4	43	10	7	5	0	6
18	`algorithms`	35	2	1	7	68	9	10	2	0	6
19	`computational_complexity`	9	0	1	7	28	2	7	5	1	5
20	`database_systems`	21	1	1	1	36	9	5	2	1	5
21	`information_retrieval`	20	2	1	3	29	5	6	3	0	5
22	`distributed_computing`	16	2	1	4	51	15	7	3	1	8
23	`cloud_computing`	11	2	0	4	29	5	5	2	1	6
24	`edge_computing`	13	1	0	4	38	1	5	2	1	7
25	`computer_networks`	45	9	1	4	96	42	5	1	0	5
26	`cybersecurity`	34	7	0	1	29	16	6	2	1	8
27	`applied_cryptography`	14	3	0	4	29	9	6	3	0	6
28	`blockchain`	6	1	0	3	19	3	4	2	1	5
29	`computer_architecture`	21	4	1	5	59	6	4	2	1	7
30	`ai_accelerators`	11	1	0	5	44	1	5	3	1	7
31	`operating_systems`	12	0	0	5	39	6	4	2	1	6
32	`quantum_computing`	16	2	0	6	27	4	3	2	1	5
33	`scientific_computing`	41	4	1	6	57	10	4	2	0	8
34	`computational_biology`	34	3	1	3	60	12	3	2	0	7
35	`computer_graphics`	15	2	1	4	21	8	9	2	0	5
36	`privacy_preserving_computation`	13	1	0	4	26	1	5	3	1	7
37	`federated_learning`	4	0	0	4	15	0	5	2	1	6
38	`ai_safety`	8	1	0	5	13	4	14	7	1	7
39	`sustainable_computing`	11	2	0	6	28	2	4	2	1	7
40	`autonomous_scientific_discovery`	12	0	0	8	26	0	4	3	1	8
	Total (entries)	646	93	21	162	1320	291	273	106	24	236

Table 7. Signal entries aggregated by area (sums of the rows of Table 6; a topic or venue mapped to several leaves of the same area is counted once per leaf).

Area	Leaves	Topic entries	arXiv entries	DBLP entries	Keyword patterns
`ai_ml`	9	69	37	149	97
`perception_interaction`	6	138	18	209	46
`software_pl`	4	40	13	129	23
`theory`	2	44	14	96	17
`data_distributed`	6	126	20	279	33
`security_privacy`	5	71	16	118	26
`hardware_emerging`	5	71	27	197	20
`computational_science`	3	87	17	143	11

The weight scale has a second consequence that the reader of Part IX must keep in mind. A signal with weight below 1.0 can only contribute through corroboration by a signal of *another* type, because repetitions within a type are capped. The design relies on this in both directions: a general machine-learning venue at 0.3 and the `cs.LG` category at 0.2 never assign `ml_theory` by themselves, whereas a 0.6 topic and a 0.4 arXiv category do reach the threshold. One discrepancy between the documentation and the implementation of the within-type aggregation (maximum versus sum over matched topics of the same leaf) is recorded in Section 17.1; the implemented rule, the sum, is the one described here and the one fixed by the unit tests.

A.4. Coverage of the OpenAlex topic vocabulary

The OpenAlex export of 2026-10-14 contains 4,516 topics, 302 of them in field 17 (Computer Science). The ontology maps 240 of the 302 (79.5 %) to at least one leaf and leaves 62 unmapped with an explicit reason each (Section A.6). Table 8 breaks the field-17 coverage down by subfield. Coverage is complete for Computer Vision and Pattern Recognition, Hardware and Architecture, Human–Computer Interaction, Signal Processing and Software, and nearly complete for Computational Theory and Mathematics; it is lowest for Information Systems (45 of 78), the subfield into which OpenAlex places most of its educational-technology and media clusters, and for Artificial Intelligence (61 of 77), where the unmapped topics are physical-science, generic residual or empty clusters rather than AI research.

Topics outside field 17 matter because computer-science work is frequently assigned a primary topic from Engineering, Mathematics, Physics or the life sciences, and because several leaves (computational biology, scientific computing, quantum computing, speech and audio, robotics) have their natural OpenAlex home outside field 17. The ontology therefore maps 170 further topics from eighteen other fields (Table 9), Engineering (71) and Mathematics (20) being the largest contributors, for a total of 410 distinct mapped topics and 646 topic–leaf weights. These

Table 8. Coverage of the OpenAlex field 17 (Computer Science) topics by subfield: number of topics in the 2026-10-14 export, mapped to at least one category, and left unmapped with a stated reason (Table 12).

Subfield	Name	Topics	Mapped	Unmapped
1702	Artificial Intelligence	77	61	16
1703	Computational Theory and Mathematics	24	23	1
1704	Computer Graphics and Computer-Aided Design	4	3	1
1705	Computer Networks and Communications	43	35	8
1706	Computer Science Applications	10	7	3
1707	Computer Vision and Pattern Recognition	35	35	0
1708	Hardware and Architecture	8	8	0
1709	Human-Computer Interaction	8	8	0
1710	Information Systems	78	45	33
1711	Signal Processing	10	10	0
1712	Software	5	5	0
	Field 17 total	302	240	62

adjacent topics were selected by reading the OpenAlex keyword lists of about 120 candidate topics named in the area modules; the remaining roughly 4,000 topics of the export were not inspected, which is the main known gap of the mapping (Section A.8). Of the 410 mapped topics, 146 point to more than one leaf, up to six for the broadest clusters (for instance T10054, *Parallel Computing and Optimization Techniques*, scores in `computer_architecture` at 0.8, `compilers` and `distributed_computing` at 0.6, `scientific_computing` at 0.4, `ai_accelerators` at 0.3 and `operating_systems` at 0.2). The full audit list of multiply mapped topics is the last generated section of `ONTOLOGY.md`; it is the first place to look when a co-classification pattern in Part IX seems suspicious, because a topic mapped to two leaves at weight 1.0 produces that pair mechanically.

A.5. Coverage of the DBLP venue vocabulary

DBLP streams are the venue signal. The stream table from which the mapping was built lists every DBLP journal or conference series with at least one hundred records in 2018–2026 (3,569 streams); the ontology maps 771 of them. Measured on the DBLP records of 2018–2026 and excluding CoRR (the arXiv mirror stream, which carries no venue information), the mapped streams cover 2,017,910 of 3,319,884 records (60.8 %). Among the 599 largest streams, 496 are mapped (82.8 %); the 103 unmapped large streams (Table 10 lists the 25 largest) are of two kinds. Some are out of scope for the forty categories although DBLP indexes them: control theory (*IEEE Transactions on Automatic Control*, *Automatica*, *IEEE Control Systems Letters*), instrumentation and industrial electronics, geo-information, agriculture, production research, information-systems and engineering-education conferences (HICSS, AMCIS, ICIS, FIE). Others are too generic to be a category signal at any weight, the largest being *IEEE Access* with 108,838 records, followed by *Information*, *Complexity*,

Table 9. Distinct OpenAlex topics mapped to at least one category, by OpenAlex field of the topic (`topic_map.csv`).

Field	Name	Mapped topics
17	Computer Science	240
22	Engineering	71
26	Mathematics	20
13	Biochemistry, Genetics and Molecular Biology	18
31	Physics and Astronomy	15
33	Social Sciences	10
18	Decision Sciences	8
27	Medicine	6
12	Arts and Humanities	5
28	Neuroscience	4
23	Environmental Science	3
19	Earth and Planetary Sciences	2
32	Psychology	2
11	Agricultural and Biological Sciences	1
14	Business, Management and Accounting	1
20	Economics, Econometrics and Finance	1
21	Energy	1
25	Materials Science	1
36	Health Professions	1
	All fields	410

Systems and *PeerJ Computer Science*; works published there are classified from their topics, arXiv categories and text alone. A record in an unmapped stream is therefore not excluded from the corpus (scope is defined in Part III, not here); it simply receives no venue evidence. Because weights are assigned per stream, workshop volumes that share a stream key with a flagship conference inherit the flagship's weight, a known coarseness of the venue signal.

Table 10. The 25 largest DBLP streams (2018–2026 records, CoRR excluded) deliberately left unmapped, out of 103 unmapped streams among the 599 largest (`ONTOLOGY.md`, generated section "DBLP coverage").

Stream	Records	Venue
`journals/access`	108,838	IEEE Access
`journals/tim`	13,470	IEEE Trans. Instrum. Meas.
`journals/tie`	10,844	IEEE Trans. Ind. Electron.
`conf/hicss`	7,816	HICSS
`conf/iecon`	7,727	IECON
`journals/cea`	7,291	Comput. Electron. Agric.
`conf/smc`	6,048	SMC
`journals/complexity`	5,786	Complex.
`journals/ress`	5,732	Reliab. Eng. Syst. Saf.
`journals/tac`	5,687	IEEE Trans. Autom. Control.
`journals/information`	5,621	Inf.
`journals/taes`	5,543	IEEE Trans. Aerosp. Electron. Syst.
`journals/jfi`	5,432	J. Frankl. Inst.
`journals/ijgi`	5,118	ISPRS Int. J. Geo Inf.
`journals/automatica`	4,727	Autom.
`conf/eucc`	4,160	ECC
`conf/iccsa`	3,817	ICCSA (4)
`conf/fie`	3,807	FIE
`conf/amcis`	3,798	AMCIS
`journals/peerj-cs`	3,768	PeerJ Comput. Sci.
`journals/systems`	3,667	Syst.
`journals/csysl`	3,647	IEEE Control. Syst. Lett.
`journals/aei`	3,628	Adv. Eng. Informatics
`journals/ijpr`	3,591	Int. J. Prod. Res.
`journals/sj`	3,558	IEEE Syst. J.

A.6. Unmapped field-17 topics and the reasons

OpenAlex places in field 17 a number of clusters that are not computer-science research. The ontology leaves 62 field-17 topics unmapped and records a reason for each; Table 11 groups the reasons into eight codes and Table 12 lists the topics. The dominant case, 29 topics, is education and e-learning research (pedagogy, learning outcomes, teacher development, educational technology studied as an intervention rather than as software); such work is not one of the forty categories,

and educational technology *as software* is still captured through software-engineering and HCI venues and keywords when the contribution is technical. Nine topics are physical-science or engineering clusters without computing content (sensor instrumentation, photovoltaics, geochemistry, seismology, nonlinear dynamics, control theory, transportation safety, chemical and environmental engineering). Five are generic residual labels ("Computer Science and Engineering", "Diverse Interdisciplinary Research Studies", "Mathematics, Computing, and Information Processing", "Statistical and Computational Modeling", "Advanced Technologies and Applied Computing") whose keyword lists carry no specific content; works bearing them reach categories through their other topics, venue, arXiv category and text. Four are media, communication, literary or philosophical clusters, four are out-of-scope disciplines (psychiatry, economic growth, cognitive science, pandemic health care), one is the history of computing (kept out of the research-category map but relevant to Part II), one is the small cognitive-informatics cluster that has no counterpart among the forty leaves and was not forced into `ai_agents`, and nine are residual topics with zero works in the export. The works column of Table 12 is the OpenAlex `works_count` of the topic over all years and fields; it is reported to show the scale of what is set aside (the education cluster alone exceeds one million works, the media cluster T13745 alone 204,789) and is not a count of corpus records. The hand-written design note of `ONTOLOGY.md` speaks of 63 unmapped clusters; the generated tables, which are authoritative, list 62.

Table 11. Reason codes for the unmapped field 17 topics (Table 12), with the number of topics and the sum of their OpenAlex `works_count`.

Code	Reason recorded in the ontology	Topics	Works (sum)
E	Education / e-learning research; not a CS research category; technical ed-tech captured elsewhere	29	1,027,591
P	Physical-science or engineering cluster without computing contribution, placed in field 17 by OpenAlex	9	352,761
Z	Residual OpenAlex topic with zero works in the export; nothing to map	9	0
G	Generic, uninformative residual label; works reach categories through other signals	5	30,206
M	Media, communication, cultural, literary or philosophy-of-science cluster; not CS research	4	275,159
O	Out-of-scope discipline (psychiatry, economics, cognitive science, pandemic health care)	4	33,228
H	History of computing technologies (history of science); background for Part II only	1	72,582
C	Cognitive informatics / brain informatics; idiosyncratic cluster with no counterpart among the leaves	1	13,053
	Total	62	1,804,580

Table 12. The 62 OpenAlex field 17 topics left unmapped, with the subfield, the number of works of the topic in the 2026-10-14 topic export (all years, all fields) and the reason code of Table 11.

Topic	Name	Subfield	Works	Reason
T13745	Media and Digital Communication	1705	204,789	M
T14496	Educational Technology and E-Learning	1710	163,032	E
T14516	English Language Learning and Teaching	1710	107,202	E
T14430	Higher Education and Teaching Methods	1710	103,711	E
T12797	History of Computing Technologies	1706	72,582	H
T12564	Sensor Technology and Measurement Systems	1705	71,005	P
T11276	Solar Radiation and Photovoltaics	1702	70,601	P
T14455	Technology and Education Systems	1705	70,588	E
T12157	Geochemistry and Geologic Mapping	1702	69,914	P
T12171	Open Education and E-Learning	1706	66,332	E
T14498	Educational Technology in Learning	1710	63,096	E
T14216	Multimedia Learning Systems	1710	62,005	E
T11187	Nonlinear Dynamics and Pattern Formation	1705	61,626	P
T14103	Educational Management and Quality	1710	57,729	E
T11370	Mobile Learning in Education	1710	55,961	E
T12408	Educational Innovations and Challenges	1710	53,673	E
T12395	Educational Innovations and Technology	1710	45,138	E
T14025	Educational Technology and Assessment	1710	37,519	E
T13602	Educational Methods and Media Use	1710	35,313	E
T13194	ICT in Developing Communities	1710	32,298	M
T13615	Transportation Systems and Safety	1704	26,785	P
T14215	Chemical and Environmental Engineering Research	1710	25,315	P
T13018	Seismology and Earthquake Studies	1702	23,104	P
T13081	Digital Education and Society	1710	22,995	E
T14029	Thoreau and American Literature	1710	19,567	M
T13386	Educational Methods and Teacher Development	1710	19,565	E
T14345	Scientific Research and Philosophical Inquiry	1710	18,505	M
T13016	Innovative Educational Technologies	1710	16,953	E
T13674	Computer Science and Engineering	1702	16,620	G
T13559	Edcuational Technology Systems	1702	16,507	E
T14381	Psychiatry, Mental Health, Neuroscience	1702	15,800	O
T14519	Innovations in Education and Learning Technologies	1710	14,509	E
T13062	Cognitive Computing and Networks	1702	13,053	C
T13867	Economic Growth and Development	1710	9,780	O
T13898	Diverse Interdisciplinary Research Studies	1702	6,495	G
T14246	E-Learning and Knowledge Management	1706	6,373	E
T13523	Mathematics, Computing, and Information Processing	1703	6,343	G
T12805	Cognitive Science and Mapping	1702	5,430	O
T13935	Mathematical Control Systems and Analysis	1702	4,409	P
T14409	Educational Systems and Policies	1710	3,650	E
T13565	Education and Learning Interventions	1710	2,867	E
T14071	Healthcare during COVID-19 Pandemic	1710	2,218	O
T14403	Educational Challenges and Innovations	1710	1,150	E
T13482	Engineering Education and Technology	1710	765	E
T14351	Statistical and Computational Modeling	1702	731	G
T13932	Education, Technology, and Economics	1710	577	E
T13873	Educational Methods and Technology	1710	329	E

continued on next page

Table 12. Unmapped field 17 topics (continued).

Topic	Name	Subfield	Works	Reason
T13676	Educational and Technological Research	1710	52	E
T13345	Advanced Technologies and Applied Computing	1705	17	G
T14387	Environmental Engineering and Cultural Studies	1710	2	P
T13514	Experience-Based Knowledge Management	1702	0	Z
T13567	AI and Multimedia in Education	1702	0	E
T13693	Smart Systems and Machine Learning	1705	0	Z
T13748	Advanced Statistical Modeling Techniques	1705	0	Z
T13836	Educational Research and Pedagogy	1705	0	E
T13964	Educational Technology and Optimization	1710	0	E
T14064	Organizational and Employee Performance	1702	0	Z
T14155	Advanced Technology in Applications	1710	0	Z
T14174	Artificial Intelligence and Decision Support Systems	1710	0	Z
T14175	Intuitionistic Fuzzy Systems Applications	1702	0	Z
T14413	Advanced Technologies in Various Fields	1702	0	Z
T14431	Diverse Research and Applications	1710	0	Z

A.7. Known ambiguities and the decisions taken

Forty categories drawn up for a mandate are not a partition of the literature, and several pairs or triples overlap by construction. The ontology's author resolved each overlap by a stated rule rather than by leaving it to the weights; the rules are recorded here verbatim in substance, because they determine how the multi-label statistics of Part IX must be read.

Large language models, foundation models, generative AI and NLP. `foundation_models` is the parent concept (any modality, adaptation-oriented); `llm` is its language child and the home of the ChatGPT/GPT/LLaMA-era literature; `generative_ai` concerns the generative model and its outputs (diffusion, GANs, text-to-X synthesis, deepfakes and detection), not text generation by LLMs as such; `nlp` keeps task-framed language work. Multi-label assignment is expected and intended: an instruction-tuning paper scores in `llm` and `nlp`, a diffusion language model in `llm` and `generative_ai`. The OpenAlex topic T10028, labelled "Topic Modeling", is in fact the cluster hosting BERT, pre-trained language models and retrieval-augmented generation and is weighted 1.0 for `llm`; a reader who consults the OpenAlex label alone would misjudge this choice.

Agents, reinforcement learning and robotics. Software autonomy (LLM agents, multi-agent systems, planning) is `ai_agents`; learning from reward is `reinforcement_learning`; physical embodiment is `robotics`. The word "agent" never counts on its own: the patterns require a computing qualifier ("LLM agent", "autonomous agent", "multi-agent system") and an exclusion vetoes chemical, medical, commercial and other non-computing senses (reagent, contrast agent, real-estate agent).

AI safety, cybersecurity and the statistical side of interpretability. Adversarial examples, backdoors and prompt injection score in both `ai_safety` and `cybersecurity` at 0.8, so that either a second signal or the text decides; fairness and governance belong to `ai_safety`; explainability as a statistical method is `ml_theory` (0.5), whereas interpretability of large models as a safety practice is `ai_safety` (1.0).

Cloud, distributed and edge computing. `distributed_computing` is the parent (algorithms, HPC systems software, consensus); `cloud_computing` is the service model (virtualisation, serverless, data-centre resource management); `edge_computing` is placement near users and devices (offloading, multi-access edge computing, on-device inference). Internet-of-things topics are shared between `edge_computing` and `computer_networks`.

Cryptography, security and privacy. Primitives and protocols are `applied_cryptography`; attacks, defences and systems hardening are `cybersecurity`; using cryptography or statistics to protect data is `privacy_preserving_computation`, whose child `federated_learning` has no OpenAlex topic and no venue of its own and is carried by keywords. The topic T10764 ("Privacy-Preserving Technologies in Data") hosts both the parent (1.0) and the child (0.7).

Architecture, accelerators and operating systems. Accelerators are a child of architecture and include efficient-inference *systems* (serving engines, attention kernels, quantisation for hardware); operating systems have no OpenAlex topic and are carried by the OSDI, SOSP, EuroSys, USENIX ATC and FAST streams, by `cs.OS` and by kernel and file-system keywords.

Quantum computing and quantum physics. Only computing and information-processing framings count. Physics topics on qubit platforms are weighted 0.2–0.3, and an anchored exclusion removes texts about quantum dots, quantum chemistry, quantum field theory and similar physical senses whenever no computing term (qubit, quantum algorithm, error correction, QKD, NISQ, . . .) appears anywhere in the title or abstract.

Scientific computing, computational biology and autonomous discovery. A numerical method is `scientific_computing`; a biological object is `computational_biology`; AI that drives the scientific process itself (AI-scientist systems, self-driving laboratories, automated hypothesis generation, AI-discovered algorithms or theorems) is `autonomous_scientific_discovery`, which is keyword-driven and vetoed when no AI or automation term is present.

Algorithms and complexity. STOC, FOCS and ICALP are shared; ECCC, *ToCT* and `cs.CC` are complexity-only. Metaheuristics and evolutionary computation are counted as `algorithms` at 0.5–0.6 because the corpus is expected to contain many such works and no other leaf claims them.

Education, media and other field-17 residues. As documented in Section A.6, the educational, media, physical-science and generic clusters that OpenAlex places in field 17 are left unmapped, with four exceptions mapped at low weight; this is a deliberate narrowing of "computer science" relative to the OpenAlex field, and its effect on corpus scope is examined in Part III, where scope is defined by DBLP membership and OpenAlex primary field rather than by the ontology.

A.8. Limitations of the mapping and the validation design

The weights are expert priors set by reading keyword lists and venue scopes, not parameters fitted to labelled data, and the ontology's own documentation says so. Three limitations follow and are repeated here because they bound what the classification can show. Venue weights are per DBLP stream, so a workshop sharing a stream key with its main conference inherits the main conference's weight. OpenAlex keyword lists were read in full for the 302 field-17 topics and for about 120 CS-adjacent topics named in the area modules, but not for the remaining roughly 4,000 topics; relevant clusters from Medicine or Materials Science may be missing and are to be added when the corpus reveals them, through an amendment. And keyword patterns on abstracts fire on mentions as well as on contributions, a failure mode discussed with the others in Section 17.

The ontology was validated, after the classification run, on a stratified sample whose design is fixed in `taxonomy/VALIDATION.md`: fifty eligible works with an abstract per rule-assigned primary category (forty strata) plus two hundred works to which the rules assigned no category, drawn with seed 44; two independent LLM raters that never see the rule-based assignment and receive only the definitions of Table 5 with the title, abstract, venue and year of each work; the principal investigator as third rater on a 200-work subsample (checkpoint H-2). The pre-specified decision rule is that a category with consensus $F_1 < 0.60$ triggers a mapping revision, logged as a protocol amendment, and a re-run of the classification. The agreement statistics, the per-category precision and recall, the list of categories that met or failed the decision rule, and any amendment that resulted are reported in Part IX; nothing about them is asserted in this appendix.

B. Search strategies, sources and acquisition record

This appendix is the operational counterpart of Part III. Part III explains why four bibliographic sources were combined and what each contributes; the present appendix records, source by source, the endpoint or bucket that was read, the release or version that was read, the filters that were

applied, the rate limits and budgets under which the reading took place, the licence of the material, the integrity checks performed on arrival, and what was sought but not obtained. It then gives the reporting items of the PRISMA-S extension for literature searches (Rethlefsen et al., 2021) in the form they take for a scope-filtered mapping study, the exclusion codes of the eligibility stage with their predicates, the type and platform lists on which those predicates depend, and the record-linkage cascade with its thresholds. Every statement below can be checked against the scripts in `ingestion/` and `systematic_review/`, the runbook (`reproducibility/RUNBOOK.md`) and the provenance note `data/raw/README.md`; where a count is a property of the corpus it is rendered by a macro of `tables/numbers.tex`, and where no macro exists the value is marked pending rather than typed.

B.1. Acquisition at a glance

Table 13 lists the five acquisitions in the order of the runbook. All were performed from the compute node `M3U96b` between the evening of 2 November and 3 November 2026 (Eastern Time), with no credential of any kind: the OpenAlex key, the Semantic Scholar key and the Kaggle credentials were tested as absent before the first download (`reports/PHASE0_FEASIBILITY.md`, §2). The absence of credentials is not incidental. It determined the choice of the snapshot over the API for OpenAlex (decision D1 of the plan), it excluded Semantic Scholar (D3), and it capped the post-snapshot increment at the public daily budget (Section B.4).

B.2. OpenAlex works snapshot

Bucket, release and manifest. The works entity of the OpenAlex snapshot (Priem et al., 2022) is distributed as gzip-compressed JSON Lines files in the public S3 bucket `s3://openalex/` under the prefix `data/jsonl/works/`, partitioned by the date at which the records were last updated (`updated_date=YYYY-MM-DD/part_NNNN.gz`). The release read is the one whose `manifest.json` carries the date 2026-10-14; the manifest lists 1,898 part files with, for each, its URL, its content length in bytes and its record count, together with the totals (464,765,485 works, 675,298,563,638 bytes). A copy of the manifest is kept as `data/raw/openalex/works_manifest_2026-10-14.json` and is the reference against which the ledger is checked. No AWS client was installed on the node; the bucket is public and each object was fetched over HTTPS from `https://openalex.s3.amazonaws.com/` with `curl` (`-L -retry 5 -retry-delay 3 -retry-all-errors -C -`, the last option resuming an interrupted transfer), up to five outer attempts per file with a linearly increasing pause, and the transfer was accepted only when the byte count on disk equalled the content length in the manifest. The measured single-stream throughput from the node was about 71 MB/s and the aggregate over eight streams about 110 MB/s, which fixed the worker count at eight and the wall

Table 13. Acquisitions performed for the WP44 corpus, in runbook order.

Source	Object read	Obtained	Integrity record
OpenAlex works	public snapshot, release 2026-10-14, `data/jsonl/works/`, 1,898 part files, 675,298,563,638 bytes gzip	2026-11-02 21:19 to 23:05 ET, HTTPS, 8 parallel streams	manifest copy; content length and SHA-256 per file in `works/ledger.sqlite`; 1,898/1,898 done, 0 malformed lines; record total equal to the manifest total
OpenAlex entities	snapshots of topics, subfields, fields, domains, sources, institutions, publishers, funders, concepts	2026-11-02	`entities_ledger.json`: content length and SHA-256 per file
OpenAlex API	works with any CS topic and publication date 2026-10-01 to 2026-11-02, by day and by primary field	2026-11-03 onward, within the public budget	`api_increment/ledger.json`: calls, cost, records and pages per slice; raw pages kept verbatim
DBLP	`dblp.xml.gz` dated 2026-11-02 22:39 UTC, 1,116,270,104 bytes, with `dblp.dtd` and `dblp.xml.gz.md5`	2026-11-02	published MD5 `ff1c424c3b4cef36 582f647c36df27c0` verified on the whole file before parsing; parse report with size, MD5 of the first GiB, counts by type and year
arXiv	OAI-PMH, set `cs`, metadata prefix `arXiv`, datestamps 2018-01-01 to 2026-11-02	2026-11-02 to 03	`harvest_state.json` (pages, records, seconds per window); 758 raw XML pages kept; `arxiv_parse_report.json`
ACL Anthology	`anthology+abstracts.bib.gz`, 46,723,382 bytes	2026-11-02	SHA-256 `44c3f7e907f221b740 bea09069818a3d7c0d fd88b2da3b750d9f02 955efc58b3`; `acl_parse_report.json`

time at one hour and forty-six minutes for the whole bucket (`ingestion/openalex_snapshot.py -workers 8`).

Filter applied on the stream (pass 1). Each part file was decompressed in memory and parsed line by line (`orjson`); a line that failed to parse was counted as malformed and skipped, and the count of such lines (zero for every file) is in the ledger. For every parsed work w the script evaluated the predicate

$$\mathrm{CS}(w) \;=\; \Big[\exists\,\theta \in \mathrm{topics}(w) : \mathrm{field}(\theta) = \texttt{fields/17}\Big] \;\vee\; \Big[\mathrm{field}\big(\mathrm{primary_topic}(w)\big) = \texttt{fields/17}\Big], \tag{62}$$

that is, whether any of the up to three topics attached to the record, or its primary topic, belongs to the OpenAlex field with identifier 17 (*Computer Science*; OpenAlex, 2024), and kept the *complete* JSON record when $\mathrm{CS}(w)$ held and `publication_year` lay in $[2018, 2026]$. Records flagged `is_xpac` (OpenAlex's expansion of coverage from non-Crossref sources) were kept at this stage with their flag, so that the exclusion could be applied later by the eligibility predicate rather than silently at ingestion. The retained records were re-compressed with Zstandard (level 6) into one file per part, `works/cs/updated_date=X__part_NNNN.jsonl.zst`, about 20 GB in total for 15,490,159 records. No text search of any kind was performed: the scope of the OpenAlex layer is defined entirely by the topic classifier's field assignment, which is the search strategy appropriate to a field-level mapping and whose consequences are discussed in Part III.

Slim table of every work. In the same pass the script wrote, for *every* work in the snapshot regardless of field or year, one row of a slim Parquet table with the following columns: OpenAlex identifier (as an integer), DOI (lower-cased, prefix removed), publication year and date, type, `cited_by_count`, primary topic, subfield, field and domain identifiers, the value of $\mathrm{CS}(w)$, the XPAC, open-access and retraction flags, language, primary source identifier, number of referenced works, number of authors, whether an abstract is present, and FWCI. The 1,898 per-part files were later concatenated into `compact/slim_all_works.parquet` (464,765,485 rows). The slim table is what allows a reference from a computer-science work to a work in another field to be typed and dated (Part IV), and it is also what made the second pass of Section B.3 possible without re-reading the bucket for a search.

Integrity and provenance. Before parsing, the SHA-256 digest of the gzip bytes of each part file was computed and written, with the file's URL, content length, manifest record count, parsed record count, retained record count, malformed-line count, download and total times, attempt count and timestamps, to a SQLite ledger (`works/ledger.sqlite`, table `files`); a second table (`runs`)

records the manifest date and totals, the worker count, the host name and the command line of every invocation. The ledger makes the job resumable (a file with status `done` is never re-read) and is the provenance record of the raw layer. The raw `.gz` files were deleted after a successful parse (decision D6 of the plan): the snapshot is a public, versioned release, the manifest and the per-file digests permit an independent party to re-download and verify any part, and keeping 675 GB of compressed JSON on the node would have precluded the rest of the pipeline. Three checks were made at the end of the pass and are recorded in `data/raw/README.md`: all 1,898 files have status `done`; the sum of parsed records over files equals the manifest total; and the sum of malformed lines is zero. The OpenAlex snapshot is released under CC0 1.0 (`LICENSE.txt` in the bucket, copied next to the manifest), which permits redistribution of derived record-level data.

Entity snapshots. The smaller entity snapshots were downloaded from the same bucket by `ingestion/openalex_entities.py`, using each entity's own manifest, with the same length check and SHA-256 ledger (`entities_ledger.json`). The manifests list 4,516 topics, 252 subfields, 26 fields, 4 domains, 268,629 sources, 143,455 institutions, 10,713 publishers, 41,799 funders and 65,026 concepts; the topics snapshot was flattened to `topics_flat.csv` (identifier, name, subfield, field, domain, works count, keywords, description) and is the vocabulary over which the ontology of Appendix A maps OpenAlex topics to categories. The authors entity was not downloaded, because the authorship blocks embedded in each work record already carry the author identifier, display name, ORCID, position and institutions; the keywords entity (1,881,723 entries in its manifest) was not downloaded either, the per-work keyword lists being embedded as well.

B.3. Second snapshot pass by identifier set

The first pass retains what OpenAlex calls computer science. The record linkage of Section B.11 showed that 1,077,407 DBLP records of 2018–2026 have a DOI that exists in the slim table of all works but not in the retained subset, that is, works indexed by DBLP that OpenAlex labels with no computer-science topic. Rather than accept OpenAlex's boundary, the pipeline reads the bucket a second time in *identifier-set mode*. The linkage script writes the list `data/metadata/needed_oa_ids.csv` of OpenAlex identifiers to recover (1,077,406 reached through a DBLP DOI and 39,200 through an arXiv DOI), and `openalex_snapshot.py -id-list` streams every part file again, keeping exactly the records whose identifier is in the set and writing no slim rows. The identifier set is loaded in each worker process (macOS uses the `spawn` start method, so module globals are rebuilt per worker), the same download, length and SHA-256 discipline applies, and the output lands in `works_pass2/cs/` with its own ledger. The recovered records are normalised with the suffix `_p2` and the linkage is rerun on the union of both passes (`ingestion/run_pass2_chain.sh`). The second pass costs the same bandwidth as the first—about one hour and fifty minutes of streaming—because an identifier

lookup cannot be pushed to the bucket; the price was accepted because the alternative, fetching the identifiers one by one through the API, would have exceeded the public budget by two orders of magnitude. The PRISMA counts of Part III were generated from the first pass (15,490,159 records loaded); the second pass adds its records to $\mathscr{W}_0$ when it completes and the counts are regenerated by the same scripts, so the identification total in the final PRISMA figure is [pending human review: PrismaIdentifiedOpenalexTotalRecordsLoaded after the second pass] rather than the first-pass value quoted here.

B.4. OpenAlex API increment

The snapshot ends on 14 October 2026 and the observation window on 2 November 2026. The gap was covered by the REST API (`https://api.openalex.org/works`) without a key. Two constraints of the unkeyed API shaped the harvest and are recorded here because they bound what the increment can contain. First, cursor paging requires a key, so the harvest used basic page numbering, which the API caps at 10,000 results per filter expression; the harvest is therefore *sliced*, one request series per publication day d and per primary-field class, with the filter

```
topics.field.id:17, from_publication_date:d, to_publication_date:d,
primary_topic.field.id:17   (slice ‘‘cs_primary’’)
topics.field.id:17, from_publication_date:d, to_publication_date:d,
primary_topic.field.id:!17   (slice ‘‘cs_secondary’’)
```

for every day from 2026-10-01 to 2026-11-02 (the start was placed two weeks before the snapshot date to absorb records that OpenAlex back-dates), 200 results per page, a `select` list of the fields used by the normaliser (identifiers, dates, type, language, locations, open access, authorships, topics, keywords, referenced works, citation counts, abstract inverted index, flags, FWCI, funders, indexed-in, full-text flag, APC), the polite-pool `mailto` parameter and a descriptive user agent. A slice whose reported `count` exceeds 10,000 is marked `truncated` in the ledger; the two-way split by primary field was chosen because a single day of computer-science output in 2026 can exceed the cap while each half rarely does. Second, the unkeyed API is metered: list-and-filter calls are charged at about $0.0001 each against a shared budget of $0.10 per day, and the script stops at $0.085 per day (a margin for the free single-record lookups used for reference verification), sleeps 0.12 s between calls, and halts on HTTP 429, 403 or any 5xx response so that the next day's run resumes from the ledger. Every page is stored verbatim as `api_increment/pages/<day>_<slice>/page_NNNN.json`; the ledger (`api_increment/ledger.json`) records per slice the pages, records, reported count, completion and truncation flags, and globally the number of calls and the cost. At the time of writing the harvest had used the budget of several consecutive days and was not complete; its totals ([pending human review: IncrementApiCalls], [pending human review: IncrementRecordsRetrieved],

[pending human review: IncrementTruncatedSlices]) will be reported from the ledger when the last slice closes. The conversion step (`ingest_api_increment.py`) rewrites the pages into the same `.jsonl.zst` part format as the snapshot filter (`works_pass3/`), skipping any work whose identifier is already in the pass-1 or pass-2 tables and writing an `increment_report.json` (pages, new works, works already present); the number of works added by the increment is [pending human review: IncrementNewWorks]. Two limitations follow from the design and are not recoverable from the ledger: a truncated slice loses the records beyond the 10,000th in the API's default ordering, and a record created after 2 November 2026 with a publication date inside the window is absent whatever its date. Both affect only the last weeks of a window that is already treated as right-censored (Part III), so no analysis depends on them.

B.5. DBLP

Dump and version. DBLP (Ley, 2002, 2009) publishes its whole database as a single XML file. The file read is `dblp.xml.gz` as served on 2 November 2026 (file timestamp 22:39 UTC), 1,116,270,104 bytes, together with the document type definition `dblp.dtd` and the published checksum file `dblp.xml.gz.md5`. The MD5 of the whole file, `ff1c424c3b4cef36582f647c36df27c0`, was computed and compared with the published value before parsing; the parse report additionally stores the file size and an MD5 of the first gibibyte as a quick identity check for later sessions. DBLP is released under the Open Data Commons Attribution licence (ODC-BY 1.0), which permits the redistribution of derived data with attribution. The download from `dblp.org` proceeded at about 12 MB/s, the slowest link of the whole acquisition.

Parsing rule. The DTD matters: DBLP titles use XML entities for accented characters, and a parser that does not resolve them against `dblp.dtd` corrupts titles and therefore the title-based linkage. The parser (`ingestion/dblp_parse.py`) is `lxml.etree.iterparse` over the gzip stream with `load_dtd=True`, `resolve_entities=True` and `huge_tree=True`, run from the directory containing the DTD because the dump references it relatively. It emits one row per record of the types *article*, *inproceedings*, *proceedings*, *book*, *incollection*, *phdthesis*, *mastersthesis* and *data*; person records (`www`) are consumed and discarded. For each record it keeps the key, type, modification date, publication-type attribute, title (concatenated text of the title element, so that embedded markup such as subscripts is flattened), year, author names and DBLP person identifiers, editors, journal, booktitle, volume, number, pages, publisher, series, the DOI (taken from the first electronic-edition URL beginning with `https://doi.org/`, lower-cased), the full list of electronic-edition URLs, the URL, the cross-reference to the proceedings record, and the *stream*, the part of the key before its last slash (`journals/tocs`, `conf/nips`), which is DBLP's venue identifier and the unit on which the ontology's venue signals are defined. All years are parsed; counts by type and by

year for the entire archive are written to `dblp_parse_report.json`, and the records of 2018–2026 (4,347,365 after the type restriction of the linkage step) are the ones that enter the corpus. The total number of publication records in the dump is [pending human review: DblpParsedRecords]. A second script (`dblp_streams.py`) lists the streams with at least one hundred records of type article or inproceedings in 2018–2026, with their first and last years and a sample journal or booktitle name; this list (`taxonomy/dblp_streams_2018_2026.csv`) is the input of the venue-to-category mapping and is not itself a corpus filter.

B.6. arXiv

Protocol and windows. arXiv (Ginsparg, 2011) metadata were harvested through its OAI-PMH endpoint `https://oaipmh.arxiv.org/oai` with the verb `ListRecords`, the set `cs` and the metadata prefix `arXiv`, which returns arXiv's own schema (identifier, dates, title, abstract, authors with forenames and keynames, categories, comments, journal reference, DOI, licence, report number, MSC and ACM classes). The harvest was sliced into calendar-month windows on the OAI `from`/`until` parameters from 2018-01-01 to 2026-11-02; each window was walked with the resumption token until exhausted, and every response page was written verbatim to `raw/<window>/page_NNNN.xml` before being parsed, so that the Parquet table can be rebuilt from the raw pages (`-parse-only`) without touching the network. The harvester waited 3.5 s between requests, honoured the `Retry-After` header on HTTP 503 (arXiv's flow-control response), waited thirty seconds on other errors, and gave up a window after twelve consecutive failures (none occurred). The state file `harvest_state.json` records, per window, the number of pages, of records and of seconds, and the harvest of the 107 windows produced 758 pages in about one hour.

Datestamp semantics and deduplication. A point about OAI-PMH that affects the counts must be stated. The `from`/`until` parameters select on the record's *datestamp*, which is the date of its last modification, not on its submission date. A monthly window therefore contains every record touched in that month, including old e-prints whose metadata were edited, and a record edited in two months appears in two windows. The parser keeps, for each arXiv identifier, the occurrence with the latest datestamp, skips records whose header carries `status="deleted"`, normalises whitespace in titles and abstracts, lower-cases the DOI, takes the first listed category as the primary category, and records the set specifications. The result is 915,115 distinct records, of which 890,852 have a `created` date (first submission) in 2018–2026; the difference consists of older e-prints modified during the window, which are kept in the table for linkage but do not belong to the observation window. The `created` date is the one used as $d_{\mathrm{arx}}(w)$ in Part IV. arXiv's metadata are offered for reuse under its metadata terms; the e-prints themselves were not downloaded in the acquisition described here.

B.7. ACL Anthology

The ACL Anthology (Radev et al., 2013b; Association for Computational Linguistics, 2026) distributes its complete bibliography as a BibTeX export. The variant with abstracts (`anthology+abstracts.bib.gz`) was downloaded on 2 November 2026 (46,723,382 bytes; SHA-256 in Table 13). BibTeX is not a regular language, so the parser (`ingestion/acl_parse.py`) walks the text by brace depth: it locates each `@type{` header, finds the matching closing brace, splits the body at its first comma into the citation key (which is the Anthology identifier, for instance `P19-1001` or `2023.acl-long.1`) and the field list, and reads each field as a braced, quoted or bare value with nested braces allowed. Fifteen fields are kept (title, author, editor, booktitle, journal, month, year, address, publisher, URL, DOI, pages, abstract, volume, number); braces inside values are removed and whitespace normalised. Entries of type `comment`, `string` and `preamble` are skipped. The export is released under CC BY 4.0. The number of entries parsed is [pending human review: AclParsedEntries]; the linkage step uses only the *inproceedings* and *article* entries (135,558), of which 85,625 fall in 2018–2026. The Anthology's role is a coverage check for natural-language processing venues rather than a corpus source in its own right; an ACL entry with no OpenAlex counterpart is counted in the linkage report but is never a work of the corpus.

B.8. What was sought and not obtained

The mandate lists sources beyond the four used, and the integrity rules require that inaccessible data be recorded rather than worked around. The record is as follows.

Semantic Scholar Academic Graph. An anonymous request to the endpoint `graph/v1/paper/` for `arXiv:2303.08774` returned HTTP 429 with the instruction to apply for a key, and the datasets endpoint requires a key outright (Kinney et al., 2023b). No key was requested because the project rule is not to assume credentials and because applying would have made the acquisition depend on an approval outside the project's control. The consequence is the absence of the influential-citation and citation-intent signals from every analysis, stated wherever it matters.

OpenAlex API key. A free key raises the daily budget from \$0.10 to \$1 and enables cursor paging. Obtaining it requires the principal investigator to confirm an e-mail; this was not done without instruction, and the increment of Section B.4 was run under the unkeyed budget. The snapshot route does not depend on the key.

Full texts. No full text was downloaded during corpus construction. Abstracts come from the OpenAlex inverted index (and from arXiv and the Anthology for their records). The plan reserves lawful full-text access (arXiv e-prints at one request per three seconds, open-access locations) for at most 10,000 high-priority works in a later phase; the knowledge-graph extraction of Part IV is abstract-based and labelled as such.

Crossref. The REST API was reached (`works?filter=from-pub-date:2022-11-30&rows=0` returned a count for all fields) and is used for the verification of every DOI cited in the manuscript and for sample cross-checks of types and citation counts; it was not used as a corpus source, OpenAlex already carrying its metadata.

Other sources listed in the mandate. Kaggle arXiv snapshot, Microsoft Academic Graph, Papers with Code, institutional repositories: no Kaggle credentials were configured and OAI-PMH made the snapshot unnecessary; the Microsoft Academic Graph was discontinued in 2021; benchmark and code repositories are scoped to the reproducibility analyses of a later phase and are not bibliographic sources; institutional repositories enter only through OpenAlex's indexing of them, which the generic-deposit rule of Section B.10 then restricts.

B.9. PRISMA-S search record

PRISMA-S (Rethlefsen et al., 2021) is written for keyword searches of bibliographic databases. A scope-filtered mapping has no search string, but its sixteen items can still be answered, and answering them makes explicit which items have no counterpart. Table 14 gives the record; the PRISMA 2020 flow itself (Page et al., 2021b; Haddaway et al., 2022) is Figure 1 and Table **??** of Part III.

B.10. Exclusion codes and their predicates

The eligibility stage (`systematic_review/define_corpus.py`) computes seven exclusion flags and two membership flags as columns of the table `corpus_works`, each a SQL expression over the normalised OpenAlex record and the linkage tables. Table 15 paraphrases the predicates. Three lists on which they depend are reproduced verbatim from the script, because their content is the substance of the type rule and of amendment A-002.

- T_{keep} (types kept without further condition): `article`, `conference-paper`, `preprint`, `book-chapter`, `book`, `dissertation`, `report`, `review`, `software-paper`.
- T_{hard} (types excluded even for DBLP-linked records): `editorial`, `erratum`, `retraction`, `peer-review`, `paratext`, `conference-abstract`, `reference-entry`, `dataset`, `software`, `supplementary-materials`, `libguides`, `letter`, `grant`, `standard`.
- $\mathscr{G}$ (generic deposit platforms, matched on the primary source's display name): "Zenodo (CERN European Organization for Nuclear Research)" and "Zenodo"; "Figshare"; "Institutional Repositories DataBase (IRDB)"; "LA Referencia (Red Federada de Repositorios Institucionales de Publicaciones Científicas)"; "Dialnet (Universidad de la Rioja)"; "DOAJ (DOAJ: Directory of Open Access Journals)"; "OSTI OAI (U.S. Department of Energy Office of Scientific and Technical Information)".

Table 14. PRISMA-S items answered for the WP44 corpus.

No.	Item	This study
1	Database names	OpenAlex (works and entity snapshots), DBLP, arXiv, ACL Anthology; Crossref for verification only.
2	Multi-database searching	Each source read in full under its own scope filter; no interface or vendor platform (direct bucket, dump and OAI access).
3	Study registries	Not applicable.
4	Online resources and browsing	None.
5	Citation searching	None for identification; OpenAlex reference lists are analysed but do not add records to the corpus.
6	Contacts	None.
7	Other methods	None.
8	Full search strategies	OpenAlex: predicate (62) with publication year 2018–2026, then identifier-set recovery of DBLP- and arXiv-linked works (Section B.3), then the API filter of Section B.4. DBLP: whole dump, record types listed in Section B.5, years 2018–2026 at linkage. arXiv: OAI set `cs`, datestamps 2018-01-01 to 2026-11-02, `created` in 2018–2026 at linkage. ACL: whole export, types inproceedings and article, years 2018–2026 at linkage.
9	Limits and restrictions	Year window only; no language, type or country restriction at search time (types are an eligibility criterion, Section B.10).
10	Search filters	None beyond the scope predicates; the OpenAlex topic classifier is the instrument that defines "computer science" on that layer and its properties are discussed in Part III.
11	Prior work	No search strategy was reused from a prior review.
12	Updates	One update by API increment to 2026-11-02; no further update is planned for this version.
13	Dates of searches	OpenAlex snapshot released 2026-10-14, read 2026-11-02; DBLP dump 2026-11-02; arXiv harvest 2026-11-02 to 03; ACL export 2026-11-02; API increment from 2026-11-03.
14	Peer review of the search	No librarian review; the strategy is preregistered in `protocol/PROTOCOL.md` (P3) and audited by the independent agents of Phase 8.
15	Total records	OpenAlex 15,490,159 (pass 1); DBLP 4,347,365; arXiv 890,852; ACL 85,625 (all 2018–2026).
16	Deduplication	Deterministic record linkage across sources (Section B.11) and preprint–publication merging within OpenAlex (Part III, Part IV); OpenAlex works are the counting unit; 114,115 preprint members of merged pairs are flagged E5.

Table 15. Exclusion and membership codes of `define_corpus.py`, with their predicates paraphrased from SQL. w is an OpenAlex record; *in_dblp*, *in_acl* and *arxiv_linked* are read from the link tables; t_0 = 2018-01-01, t_1 = 2026-11-02.

Code	Column	Predicate (true = flagged)	Marginal count
E1	`e1_outside_window`	`publication_date` is null, or $< t_0$, or $> t_1$	160
E2	`e2_type`	$\tau(w) \notin T_{\text{keep}}$ and not (*in_dblp* and $\tau(w) \notin T_{\text{hard}}$); a null type counts as outside T_{keep}	3,961,633
E3	`e3_xpac`	`is_xpac` and not *in_dblp*	3,950,876
—	`xpac_kept_dblp`	`is_xpac` and *in_dblp* (kept, flagged)	87,288
E4	`e4_no_title`	title null or empty after trimming	16,426
E5	`e5_duplicate`	w is the preprint member of a surviving preprint–published pair (Part IV, eq. (9))	114,115
E6	`e6_retracted`	`is_retracted` (flag only; not part of eligibility)	25,165
E7	`e7_generic_deposit`	primary source name $\in \mathcal{G}$ and not *in_dblp* and no linked arXiv record	[pending human review: PrismaExcludedESevenGenericDeposit]
core	`cs_core`	*in_dblp* or `cs_primary` or (`cs_any` and linked arXiv primary category begins with `cs.`)	6,403,238 eligible
pen.	`cs_penumbra`	`cs_any` and not `cs_core`	3,595,364 otherwise eligible
—	`eligible`	¬E1 ∧ ¬E2 ∧ ¬E3 ∧ ¬E4 ∧ ¬E5 ∧ ¬E7 ∧ core	6,403,238

Three further columns are derived in the same statement and are used throughout the manuscript. The first date, `first_date`, is the least of the publication date, the merged preprint's date and the arXiv creation date, except that a publication date falling on 1 January is treated as year-only precision and is overridden by either of the other two when available; `date_precision` is `year` when the date falls on 1 January and no finer date exists, `month` otherwise; and `period` is `pre`, `baseline`, `post` or `future` according to whether the first date precedes t_0, lies in $[t_0, 2022\text{-}11\text{-}30)$, in $[2022\text{-}11\text{-}30, t_1]$, or after t_1. The PRISMA counts are computed twice, marginally (the column of Table 15) and sequentially in the order E1, E2, E3, E4, E5, E7, and the script asserts that the loaded total equals the sum of the sequential exclusions and the after-exclusion count (9,998,602); the macro set of Part III reports both. The E6 and penumbra rows are not exclusions and do not enter the sum. The numbering conflict between the protocol's E7 (penumbra) and the code's E7 (generic deposit) is resolved by amendment A-005 in favour of the code.

B.11. Record-linkage cascade and thresholds

Part IV states the cascade in prose; Table 16 gives it as executed, step by step, with every threshold. The normalised title $\eta(\cdot)$ lower-cases and deletes every character outside [*a*–*z*0–9]; "unique" means that the normalised title occurs exactly once on that side among the records still unlinked at that step (for OpenAlex in the arXiv and ACL cascades, unique among all OpenAlex records, the exclusion of already linked identifiers being applied to the pair instead); the length condition is on the raw title for DBLP and OpenAlex and on the normalised title for arXiv and ACL. The DBLP title step adds a first-author check that the other cascades do not have: the last whitespace-separated token of the first author's name on each side must coincide when both are present. Each link carries its method label, which is the stratum of the precision sample of Part IV; the counts per method are in `data/metadata/linkage_report.json` and Table **??**.

The record types admitted on the DBLP side are *article*, *inproceedings*, *incollection*, *book*, *phdthesis* and *data* with year 2018–2026; on the ACL side *inproceedings* and *article*; on the arXiv side every record of the harvest, the year condition being applied through the `created` year in the title step only. The linkage runs inside one DuckDB database with 24 threads and a 60 GB memory limit and completes in about ten seconds on the compacted tables, which is why it was possible to rerun it after each amendment rather than patch its output. The precision sample of 500 pairs (250 DOI, 200 title–year, 50 arXiv, the last stratum empty for the reason given in Part IV) is drawn by `systematic_review/linkage_sample.py` with seed 44, and its automatic agreement checks (title similarity at least 0.9, year difference at most one, first-author surname agreement) are the interim figure until the principal investigator's review (checkpoint H-1).

Table 16. The deterministic linkage cascade of `ingestion/link_sources.py`. Steps within a cascade apply only to records unlinked by the previous steps.

Cascade	Step	Join condition and thresholds	Method label
DBLP–OA	1	equal lower-cased DOI; all matching pairs kept (many-to-many allowed)	`doi`
	2	equal arXiv identifier (DBLP: parsed from `ee` URLs by `arxiv.org/abs/([^ ]+)`; both sides stripped of `v[0-9]+$`); both records unlinked	`arxiv`
	3	equal η, raw title ≥ 25 characters, η unique on both sides, $\lvert y_{\text{DBLP}} - y_{\text{OA}} \rvert \leq 1$, first-author surnames equal when both present	`title_year`
arXiv–OA	1	equal arXiv identifier (OpenAlex side: `ids.arxiv`, else DOI prefix `10.48550/arxiv.`, else `arxiv.org/abs|pdf/` in any location URL); all pairs kept	`arxiv_id`
	2	equal lower-cased DOI; arXiv record unlinked	`doi`
	3	equal η, $\lvert\eta\rvert \geq 25$, unique on both sides, $\lvert y_{\text{created}} - y_{\text{OA}} \rvert \leq 2$, OpenAlex record unlinked	`title_year`
ACL–OA	1	equal lower-cased DOI	`doi`
	2	Anthology identifier equal to the segment after `aclanthology.org/` in any OpenAlex landing-page URL (with or without trailing slash); ACL record unlinked	`url`
	3	equal η, $\lvert\eta\rvert \geq 25$, unique on both sides, $\lvert y_{\text{ACL}} - y_{\text{OA}} \rvert \leq 1$, OpenAlex record unlinked	`title_year`

B.12. Reproducing the acquisition

The acquisition can be repeated from the runbook in six commands, in the order of Table 13, on a machine with Python 3.12 and the locked environment of `reproducibility/environment_M3U96b.txt`:

1. `openalex_snapshot.py` with the manifest copy (then once more with `-id-list`);
2. `openalex_entities.py`;
3. `dblp_parse.py`;
4. `arxiv_oai.py`;
5. `acl_parse.py`;
6. `openalex_api_increment.py`, followed by `ingest_api_increment.py`.

Two of these are not exactly repeatable and the reader should know which. The OpenAlex bucket retains only recent releases, so a later reader must either locate the 2026-10-14 release or accept that a newer release will differ in record counts and in the nightly-recomputed fields (citation counts, FWCI); the per-file digests in the ledger allow a reader with the same release to prove identity, and a reader with a different release to quantify the difference by record count. The arXiv OAI feed is likewise live: a re-harvest of the same datestamp windows returns records as they now are, and the raw pages archived with the project are the only way to recover the state of 2–3 November 2026. The DBLP dump and the ACL export are dated files whose hashes are published here, and a copy of each is kept with the project's raw layer.

C. Mathematical derivations

Part IV defines the statistics of this study and states their properties; this appendix supplies the derivations behind those statements, in the order in which Part IV uses them. Each subsection fixes the object as implemented, derives the property that the main text relies on, and records any point at which the implementation departs from the convention of the cited source. The derivations are elementary and self-contained; their purpose is not novelty but the assurance that the quantities tabulated in Part IX are what the text says they are. Throughout, n is the length of a monthly series, $y_t = \log(N_t + 1)$ its logarithm, and $\mathbf{1}[\cdot]$ the indicator function.

C.1. The interrupted time series as a linear regression

Design matrix. Equation (41) of Part IV is the linear model $y = X\beta + \varepsilon$ with $y \in \mathbb{R}^n$ and $X \in \mathbb{R}^{n \times 15}$ whose columns are, for $t = 0, \dots, n-1$ and event index t_e,

$$x_t \;=\; \big(1,\; t,\; \mathbf{1}[t \ge t_e],\; (t - t_e)\,\mathbf{1}[t \ge t_e],\; \mathbf{1}[\text{month}(t) = 2],\; \dots,\; \mathbf{1}[\text{month}(t) = 12]\big)^{\top}, \tag{63}$$

exactly the data frame built by `_design()` in `econometrics/structural_change.py` (columns `const`, `t`, `post`, `t_post`, `m2`, …, `m12`). The post-event trend is centred at t_e rather than at zero so that β_2 is the level change *at* the event month and not the extrapolated intercept difference at $t = 0$; with an uncentred post-trend the two parametrisations are related by $\beta_2^{\text{unc}} = \beta_2 - \beta_3 t_e$, and only the centred one has the interpretation used in the text. The eleven month dummies are linearly independent of the constant because January is omitted; the full-column-rank condition requires at least one observation in every month of the year, satisfied by any series of twelve months or more, and at least one post-event month with $t > t_e$, guaranteed by the driver's rule that twelve months follow every event. The ordinary least-squares estimator is $\hat\beta = (X^{\top}X)^{-1}X^{\top}y$ and the residuals $\hat\varepsilon = y - X\hat\beta$.

The HAC covariance estimator. Under the assumption that ε_t is a stationary, serially correlated and possibly heteroskedastic sequence with $\mathbb{E}[x_t\varepsilon_t] = 0$, the asymptotic covariance of $\sqrt{n}(\hat\beta - \beta)$ is $Q^{-1}\Omega Q^{-1}$ with $Q = \lim n^{-1}X^{\top}X$ and $\Omega = \sum_{\ell=-\infty}^{\infty}\Gamma_\ell$, where $\Gamma_\ell = \mathbb{E}[x_t\varepsilon_t\varepsilon_{t-\ell}x_{t-\ell}^{\top}]$ is the autocovariance of the score at lag ℓ. The estimator of Newey and West (1987) replaces the infinite sum by a kernel-weighted finite one,

$$\hat\Omega_L \;=\; \hat\Gamma_0 + \sum_{\ell=1}^{L} w_\ell\,\big(\hat\Gamma_\ell + \hat\Gamma_\ell^{\top}\big), \qquad \hat\Gamma_\ell = \frac{1}{n}\sum_{t=\ell}^{n-1}\hat\varepsilon_t\hat\varepsilon_{t-\ell}\,x_t x_{t-\ell}^{\top}, \qquad w_\ell = 1 - \frac{\ell}{L+1}, \tag{64}$$

and the variance estimate is $\hat{V} = n\,(X^\top X)^{-1}\hat{\Omega}_L(X^\top X)^{-1}$, which is the expression of Part IV once the factors of n are absorbed into the un-normalised $\hat{\Gamma}_\ell$. The Bartlett weights w_ℓ are what make $\hat{\Omega}_L$ positive semi-definite for every sample: $\hat{\Omega}_L$ can be written as $\frac{1}{n(L+1)}\sum_t \big(\sum_{s=t-L}^{t} \hat{u}_s\big)\big(\sum_{s=t-L}^{t} \hat{u}_s\big)^\top$ up to end effects, with $\hat{u}_s = x_s\hat{\varepsilon}_s$, a sum of outer products, whereas the truncated sum with $w_\ell \equiv 1$ is not guaranteed to be positive semi-definite. The implementation calls `statsmodels` with `cov_type="HAC"` and `maxlags=6`, so $L = 6$ and $w_\ell = 1 - \ell/7$ as stated in Part IV, without the small-sample degrees-of-freedom correction and with p-values from the normal approximation to the t-ratio, which is the library's default for robust covariances. With n of the order of one hundred and fourteen regressors, the difference between the normal and the t_{n-15} reference distribution is immaterial at the magnitudes of the t-statistics reported; what matters is the placebo rank of the next paragraph, which does not depend on any reference distribution.

The placebo p-value as a rank statistic. Let $\mathscr{T}' = \{t' : 24 \le t' \le n - 13\}$ be the candidate event indices (the implementation's `min_pre = 24` and `min_post = 12`) and, for each $t' \in \mathscr{T}'$, let $t^{(2)}_{t'} = \hat{\beta}_2(t')/\widehat{\mathrm{se}}(\hat{\beta}_2(t'))$ be the HAC t-ratio of the level change when the model (63) is fitted with $t_e = t'$. The placebo value of the true event $t_e \in \mathscr{T}'$ is

$$p^{\mathrm{plac}} \;=\; \frac{1}{|\mathscr{T}'|}\sum_{t'\in\mathscr{T}'} \mathbf{1}\big[\,|t^{(2)}_{t'}| \ge |t^{(2)}_{t_e}|\,\big] \;=\; \frac{\mathrm{rank}\big(|t^{(2)}_{t_e}|\big)}{|\mathscr{T}'|}, \tag{65}$$

where rank is the descending rank of the true event's absolute t-ratio among the $|\mathscr{T}'|$ values, ties counted as not better (`>=`). Three properties follow directly. Because $t_e \in \mathscr{T}'$, the sum contains the term $t' = t_e$, so $p^{\mathrm{plac}} \ge 1/|\mathscr{T}'|$, which for the default window ($n = 101$, hence $|\mathscr{T}'| = 65$) is about 0.015: a placebo rank cannot be smaller, however large the true statistic, and a reported value of $1/|\mathscr{T}'|$ means "largest of all candidate dates", not "significant at that level". Because the statistic is a rank, it is invariant to any monotone transformation of the $|t|$-ratios and in particular does not depend on the HAC bandwidth through anything but the ordering it induces. And if the candidate dates were exchangeable under a null hypothesis of no break anywhere, p^{plac} would be a valid permutation-type p-value with $P(p^{\mathrm{plac}} \le \alpha) \le \alpha$; they are not exchangeable, since all $|\mathscr{T}'|$ regressions share the same n observations and adjacent candidates give nearly identical fits, so the value is reported as a rank and read as such. The same construction gives the placebo rank of the slope change from $t^{(3)}_{t'}$. The driver additionally computes the rank restricted to the baseline candidates $\{t' < t_e\}$, which answers the protocol's original question (is the reference date special relative to the pre-period?) and is reported next to the full-range rank.

C.2. Bai–Perron multiple breaks by dynamic programming

Segment costs and additivity. For a segment $[i, j)$ of the series with $j - i \geq h$, define

$$S(i, j) \;=\; \min_{a,b\in\mathbb{R}} \sum_{t=i}^{j-1} (y_t - a - bt)^2, \tag{66}$$

the residual sum of squares of a linear trend fitted to the segment alone (the implementation solves the two-parameter least-squares problem for each segment with `numpy.linalg.lstsq`; with `trend=False` the regressor t is dropped and $S(i, j)$ is the centred sum of squares). The model with m breaks $0 < \tau_1 < \cdots < \tau_m < n$ assigns its own (a, b) to each of the $m+1$ segments $[\tau_{k-1}, \tau_k)$ with $\tau_0 = 0$ and $\tau_{m+1} = n$. Because the parameters of different segments are distinct and each observation enters exactly one segment, the residual sum of squares of the whole model separates,

$$\mathrm{SSR}(\tau_1, \ldots, \tau_m) \;=\; \sum_{k=1}^{m+1} S(\tau_{k-1}, \tau_k), \tag{67}$$

and the global minimisation over τ decomposes: the optimal partition of the first j observations into $m+1$ segments contains, as its first m segments, an optimal partition of the first τ_m observations into m segments. This is the principle of optimality, and it is the only property of the cost that the dynamic programme uses; it holds for any cost that is a sum over segments of a function of the segment alone, which is why the same recursion serves the piecewise-constant model of Section C.3. It fails for a model with parameters shared across segments (a common slope, or the seasonal dummies of the ITS), which is why the segmented regression carries no seasonal terms and the series is seasonally adjusted beforehand by subtracting the baseline-period month means (`run_structural_change.py`).

The recursion. Let $C_m(j)$ be the minimum of (67) over partitions of observations $0, \ldots, j-1$ into $m+1$ admissible segments. Then

$$C_0(j) = S(0, j), \qquad C_m(j) = \min_{mh \leq i \leq j-h} \big\{ C_{m-1}(i) + S(i, j) \big\}, \qquad j \geq (m+1)h, \tag{68}$$

the lower limit of i ensuring that m segments of length at least h fit before the last break and the upper limit that the last segment has length at least h. The argument minimiser $i^*_m(j)$ is stored, and the breaks of the m-break solution for the full series are recovered backward: $\tau_m = i^*_m(n)$, $\tau_{m-1} = i^*_{m-1}(\tau_m)$, and so on, which is what the dictionary `B[m][j]` of the implementation accumulates forward instead. This is the dynamic programme that Bai and Perron (2003) propose for the global minimiser of Bai and Perron (1998); the earlier literature used it for the piecewise-constant case, and

its correctness requires nothing beyond (67) and the induction on m: if $C_{m-1}(i)$ is the true minimum for every i, then any partition of $[0, j)$ into $m+1$ segments has a last segment $[i, j)$ and a prefix partition of cost at least $C_{m-1}(i)$, so its cost is at least the right-hand side of (68), which is attained.

Complexity. The table of segment costs has one entry per admissible pair (i, j), that is $\binom{n-h+1}{2} + O(n)$ entries, of order $n^2/2$. The recursion (68) evaluates, for each $m \le M$ and each j, a minimum over at most n candidates, so its cost is $O(n^2 M)$ operations; the back-tracking costs $O(M)$. The implementation fills the cost table by solving each segment's least-squares problem independently, at a cost proportional to the segment length, so the table costs $O(n^3)$ rather than the $O(n^2)$ obtainable by updating running sums as in Bai and Perron (2003); for n of the order of one hundred, as here, the difference is a fraction of a second and the simpler code was preferred for its transparency. Trimming enters only through the admissibility condition $j - i \ge h$; with $h = \max\{8, \lfloor 0.15\, n \rfloor\}$ and $M = 3$, the default window of $n = 101$ months gives $h = 15$ and a feasible region $(m+1)h \le n$ for every $m \le 3$, so all four candidate models exist.

Information criteria as implemented. For $m = 0, \ldots, M$ let $\mathrm{SSR}_m = C_m(n)$ and let $p_m = k(m+1) + m$ be the parameter count, with $k = 2$ for the trend model ($k = 1$ for the mean model): one intercept and one slope per segment, plus the m break positions, counted as parameters by convention. The two criteria computed by the code are

$$\mathrm{BIC}(m) = n \log \frac{\mathrm{SSR}_m}{n} + p_m \log n, \tag{69}$$

$$\mathrm{LWZ}(m) = n \log \frac{\mathrm{SSR}_m}{n - p_m} + p_m\, c_0\, (\log n)^{2.1}, \qquad c_0 = 0.299, \tag{70}$$

and the selected number of breaks is $\hat{m} = \arg\min_m$ of each. The BIC form (69) is the Gaussian log-likelihood, maximised over the error variance, in which $-2 \log \hat{L} = n \log(\mathrm{SSR}/n) + \text{const}$, plus the Schwarz penalty $p \log n$ (Schwarz, 1978); the two differences from a textbook BIC are that the break positions are counted among the parameters and that the constant and the segment variances are treated as common, both as in Bai and Perron (2003). The LWZ form (70) is the modified Schwarz criterion of Liu et al. (1997), whose penalty grows as $(\log n)^{2+\delta_0}$ with $\delta_0 = 0.1$ and whose variance estimate divides by $n - p_m$; the constant $c_0 = 0.299$ is the value they recommend and the one used in the simulations of Bai and Perron (2003), which find that LWZ selects fewer spurious breaks than BIC under serial correlation at the cost of some power. The comparison across m is legitimate because every SSR_m is a global minimum over the same observations and the same admissible region; a sequential procedure that fixed earlier breaks before adding the next would not have this property and would need the sup-F machinery that is not implemented. Neither criterion yields a confidence set for the break dates; the text reports the agreement of the two selections and

the sensitivity to the truncation month instead.

C.3. PELT: cost, penalty and pruning

Objective. With the ℓ_2 cost $\mathscr{C}(y_{i:j}) = \sum_{t=i}^{j-1}(y_t - \bar{y}_{i:j})^2$ for a segment $[i, j)$ and a penalty $\lambda > 0$ per changepoint, the problem of equation (48) of Part IV is

$$F(n) \;=\; \min_{m,\ \tau_1<\cdots<\tau_m} \Big\{ \sum_{k=1}^{m+1} \mathscr{C}(y_{\tau_{k-1}:\tau_k}) + \lambda m \Big\}, \tag{71}$$

and, by the same additivity argument as in Section C.2, it satisfies the optimal-partitioning recursion of Jackson et al. (2005),

$$F(0) = -\lambda, \qquad F(j) \;=\; \min_{0\le i<j} \big\{ F(i) + \mathscr{C}(y_{i:j}) + \lambda \big\}, \tag{72}$$

where the number of breaks is not fixed in advance but chosen by the penalty; the convention $F(0) = -\lambda$ cancels the penalty charged to the first segment. Solving (72) for all j costs $O(n^2)$ cost evaluations.

The pruning condition. Killick et al. (2012) observe that if there exists a constant K such that for all $i < j < l$

$$\mathscr{C}(y_{i:j}) + \mathscr{C}(y_{j:l}) + K \;\le\; \mathscr{C}(y_{i:l}), \tag{73}$$

then any candidate i for which

$$F(i) + \mathscr{C}(y_{i:j}) + K \;\ge\; F(j) \tag{74}$$

can never be the last changepoint before any later $l > j$ and may be discarded from the candidate set. The proof is two lines. For $l > j$, by (73) and (74), $F(i) + \mathscr{C}(y_{i:l}) + \lambda \ge F(i) + \mathscr{C}(y_{i:j}) + \mathscr{C}(y_{j:l}) + K + \lambda \ge F(j) + \mathscr{C}(y_{j:l}) + \lambda$, so the partition that places a changepoint at j after the optimal prefix of $[0, j)$ is at least as good as any that ends its last segment at i; hence i never needs to be examined again. For the ℓ_2 cost, (73) holds with $K = 0$, because adding a split point to a segment can only reduce its residual sum of squares (the split model nests the unsplit one). The recursion with pruning examines, at each j, only the candidates not yet discarded; under the regularity conditions of Killick et al. (changepoints whose number grows linearly with n) the expected number of surviving candidates is bounded and the total cost is linear in n, whence the name. For a short series with few changes, as here, the pruning saves little and the method is simply an exact solution of (71); the point of using it is exactness at a given penalty, not speed. The implementation (`ruptures.Pelt`, Truong et al., 2020) is called with `model="l2"`, `min_size=6` and `jump=1`, the last two restricting

segments to at least six months and allowing a changepoint at every index; the library returns the segment end points including n, which the wrapper drops.

The penalty $2\sigma^2 \log n$. Suppose the observations are independent Gaussian with piecewise-constant mean and a common variance σ^2. Twice the negative log-likelihood of a partition, up to a constant, is $\sigma^{-2}\sum_k \mathscr{C}(y_{\tau_{k-1}:\tau_k})$. Each additional changepoint adds two parameters, a segment mean and a position, so the Schwarz criterion in this scale adds $2\log n$ per changepoint. Multiplying through by σ^2 to express the criterion in the units of the cost gives

$$\sum_k \mathscr{C}(y_{\tau_{k-1}:\tau_k}) + 2\sigma^2 \log n \cdot m, \tag{75}$$

which is (71) with $\lambda = 2\sigma^2 \log n$. The implementation replaces σ^2 by the sample variance of the whole series, $\widehat{\mathrm{Var}}(y) = n^{-1}\sum_t (y_t - \bar{y})^2$ (`numpy.var`, population form). This is the conservative choice discussed in Part IV: in a series with a trend or with genuine level shifts, the overall variance exceeds the within-segment variance, so the penalty is larger than the within-segment Schwarz penalty and fewer changepoints are declared. Yao (1988) proves that the Schwarz criterion, with a known or consistently estimated σ^2, selects the true number of changes with probability tending to one in the piecewise-constant Gaussian model; the result does not extend to a series whose mean is a trend, for which a piecewise-constant fit must spend changepoints to track the slope, and the PELT dates of this study are read with that in mind.

C.4. Bayesian online changepoint detection

Conjugate updates. Within a run the observations are modelled as $y \mid \mu, \sigma^2 \sim \mathcal{N}(\mu, \sigma^2)$ with the Normal–Inverse-Gamma prior $\mu \mid \sigma^2 \sim \mathcal{N}(\mu_0, \sigma^2/\kappa_0)$, $\sigma^2 \sim \mathrm{IG}(\alpha_0, \beta_0)$, whose joint density is proportional to

$$(\sigma^2)^{-\alpha_0 - 3/2} \exp\Big\{ -\frac{\beta_0}{\sigma^2} - \frac{\kappa_0(\mu - \mu_0)^2}{2\sigma^2} \Big\}. \tag{76}$$

Multiplying by the likelihood of one observation y, proportional to $(\sigma^2)^{-1/2}\exp\{-(y-\mu)^2/(2\sigma^2)\}$, and completing the square in μ,

$$\kappa_0(\mu - \mu_0)^2 + (y - \mu)^2 \;=\; (\kappa_0 + 1)\Big(\mu - \frac{\kappa_0\mu_0 + y}{\kappa_0 + 1}\Big)^2 + \frac{\kappa_0(y - \mu_0)^2}{\kappa_0 + 1}, \tag{77}$$

shows that the posterior is again Normal–Inverse-Gamma with

$$\mu_1 = \frac{\kappa_0\mu_0 + y}{\kappa_0 + 1}, \quad \kappa_1 = \kappa_0 + 1, \quad \alpha_1 = \alpha_0 + \tfrac{1}{2}, \quad \beta_1 = \beta_0 + \frac{\kappa_0(y - \mu_0)^2}{2(\kappa_0 + 1)}, \tag{78}$$

which are the four one-step updates of the implementation (`mu_new`, `kappa_new`, `alpha_new`, `beta_new`). Applying (78) r times and collecting terms gives the batch form quoted in Part IV, with $\kappa_r = \kappa_0 + r$, $\mu_r = (\kappa_0\mu_0 + r\bar{y})/\kappa_r$, $\alpha_r = \alpha_0 + r/2$ and $\beta_r = \beta_0 + \frac{1}{2}\sum_{i\le r}(y_i - \bar{y})^2 + \kappa_0 r(\bar{y} - \mu_0)^2/(2\kappa_r)$; the equality of the two forms is a matter of algebra (induction on r) and is the reason the online recursion can be trusted to reproduce the batch posterior.

Student-t predictive. The predictive density of the next observation integrates the Gaussian likelihood against the current Normal–Inverse-Gamma posterior. Integrating μ first gives $y \mid \sigma^2 \sim \mathcal{N}(\mu_r, \sigma^2(1 + 1/\kappa_r))$; integrating $\sigma^2 \sim \mathrm{IG}(\alpha_r, \beta_r)$ against this scale mixture of normals gives

$$p(y \mid r) \;=\; \int \mathcal{N}\!\left(y;\, \mu_r,\, \sigma^2\frac{\kappa_r + 1}{\kappa_r}\right) \mathrm{IG}(\sigma^2; \alpha_r, \beta_r)\, d\sigma^2 \;=\; t_{2\alpha_r}\!\left(y;\, \mu_r,\, \frac{\beta_r(\kappa_r + 1)}{\alpha_r \kappa_r}\right), \tag{79}$$

a Student-t density with $2\alpha_r$ degrees of freedom, location μ_r and squared scale $\beta_r(\kappa_r + 1)/(\alpha_r\kappa_r)$: the standard result that a normal with inverse-gamma mixing on the variance is a t with degrees of freedom twice the shape and scale equal to the rate over the shape times the normal's variance factor. The code evaluates (79) through `scipy.stats.t.pdf` with `df = 2*alpha` and `scale = sqrt(beta*(kappa+1)/(alpha*kappa))`, vectorised over the run lengths.

Run-length recursion and the mass on $r = 0$. Let $R_t(r) = P(r_t = r \mid y_{1:t})$ be the filtered run-length posterior after t observations, with $R_0(0) = 1$, and let $\pi_t(r) = p(y_{t+1} \mid r_t = r, y_{1:t})$ be the predictive (79) evaluated with the hyperparameters of the run of length r ending at t. With a constant hazard H (a geometric prior on run lengths), the joint recursion of Adams and MacKay (2007) is, for the growth and the change moves,

$$\tilde{R}_{t+1}(r+1) = R_t(r)\,\pi_t(r)\,(1 - H), \qquad \tilde{R}_{t+1}(0) = \sum_{r=0}^{t} R_t(r)\,\pi_t(r)\,H, \tag{80}$$

followed by normalisation, $R_{t+1}(r) = \tilde{R}_{t+1}(r)/\sum_{r'}\tilde{R}_{t+1}(r')$. Write $Z_t = \sum_r R_t(r)\pi_t(r)$ for the marginal predictive likelihood of y_{t+1}. Then $\sum_{r\ge 1}\tilde{R}_{t+1}(r) = (1 - H)Z_t$ and $\tilde{R}_{t+1}(0) = HZ_t$, so the normaliser is Z_t and

$$R_{t+1}(0) \;=\; \frac{HZ_t}{HZ_t + (1 - H)Z_t} \;=\; H \qquad \text{for every } t. \tag{81}$$

The filtered probability that a changepoint occurs *at* the current step therefore equals the prior hazard identically, whatever the data: the run-length posterior carries information about where the last change was only through the relative masses at $r \ge 1$, which are reweighted by the predictive likelihoods, and not through the mass at $r = 0$, which the constant hazard fixes. This is why the

detection statistics of Part IV are not $R_t(0)$. The first, $P(r_t \le 3 \mid y_{1:t}) = \sum_{r=0}^{3} R_t(r)$, is H plus the mass on runs of length one to three, that is, on changes that occurred one to three months earlier and have since accumulated evidence; its value exceeds H only when the recent observations are better explained by a fresh run than by the continuation of the old one. The second is the maximum-a-posteriori run length $\hat{r}_t = \arg\max_r R_t(r)$, whose resets from a value above twelve to a value at most two are the online detector's changepoints. The implementation keeps the full $(n+1) \times (n+1)$ matrix R, adds 10^{-300} to the normaliser against underflow, and initialises μ_0 and β_0 from the first six observations with $\kappa_0 = \alpha_0 = 1$; the prior mean run length $1/H = 60$ months is the only tuning constant. The recursion is the filtered one; the smoothed posterior of Fearnhead (2006) would require a backward pass and answers the retrospective question that the offline methods of Sections C.2 and C.3 already address.

D. Data Dictionary

This appendix defines every column of every table that the analyses of Parts VIII–IX read, in the order in which the pipeline creates them (Appendix E gives the commands). The dictionary was derived from the code rather than from the data: the Arrow schemas declared in the ingestion scripts and the `SELECT` lists of the DuckDB statements are the sources, and a column is described as the code computes it, including the conventions that the raw sources do not share (integer identifiers, lower-cased DOIs, booleans that are never null). A reader who wants to recompute an indicator of Part IX should be able to do so from this appendix and the released aggregate tables without reading the scripts.

D.1. Conventions

Identifiers. OpenAlex identifiers are stored as integers with the entity letter removed: `https://openalex.org/W2741809807` becomes the `int64` value 2741809807, and the same rule applies to authors (A), institutions (I), sources (S), topics (T), subfields, fields, domains and funders (function `oid` in `build_corpus.py`; a tail that is not numeric yields a null). DBLP records are identified by their DBLP key (`conf/nips/VaswaniSPUJGKP17`), arXiv records by the bare arXiv identifier without version suffix and in lower case (`2303.08774`, `cs/0112017`), ACL Anthology entries by their anthology identifier (`2020.acl-main.207`). DOIs are stored without the `https://doi.org/` prefix and in lower case, on every side of every linkage.

Types. Types are Apache Arrow types as declared in the schemas (`int64`, `int32`, `int16`, `float32`, `bool`, `string`, `list<string>`, `list<int64>`); DuckDB-created tables use the corresponding DuckDB types (`BIGINT`, `INTEGER`, `DOUBLE`, `BOOLEAN`, `VARCHAR`). Dates are ISO-8601

strings (YYYY-MM-DD); a missing date is null, never a sentinel. Booleans produced by build_corpus.py are coerced with bool() and are therefore never null (a missing flag reads as false); booleans produced by DuckDB expressions can be null only where the definition says so.

Files and views. The normalised OpenAlex tables are written as one Zstandard-compressed Parquet file per snapshot part (data/processed/tables/<table>/<part>.parquet), then compacted into one file per table (data/processed/compact/<table>.parquet) because DuckDB could not open 1,898 files per table within the macOS file-descriptor limit (incident I-2, Appendix E). The second snapshot pass writes the same tables with the suffix _p2; link_sources.py exposes each table as a DuckDB view oa_<table> that unions the two passes by column name. The raw-source tables are exposed as dblp_raw, arxiv_raw, acl_raw and oa_slim. The database file is data/processed/wp44.duckdb; it is a derived artefact and is not redistributed, whereas every table it contains can be rebuilt from the Parquet files.

Reading the tables. Each dictionary table lists the column, its type, its definition and its source: an OpenAlex JSON path (for instance primary_topic.subfield.id) for fields copied from the snapshot, a DBLP, arXiv-OAI or BibTeX element for the other sources, or the derivation for computed columns. Columns are listed in schema order.

D.2. Raw-source tables

The four raw sources are parsed once each into a Parquet table that keeps the source's own vocabulary; nothing is harmonised at this stage except the identifier conventions above. The OpenAlex *slim* table is the only table that covers every work of the snapshot, not only computer science; it exists so that references to works outside the corpus can be resolved to a publication year, type and field (Section D.7).

Table 17. slim (data/raw/openalex/works/slim/*.parquet, compacted to compact/slim_all_works.parquet; DuckDB view oa_slim): one row for every work of the OpenAlex snapshot (openalex_snapshot.py, SLIM_SCHEMA).

Column	Type	Definition	Source
id	int64	OpenAlex work identifier (letter stripped)	id
doi	string	DOI, lower case, prefix removed; null if absent	doi
publication_year	int32	Publication year	publication_year
publication_date	string	Publication date, YYYY-MM-DD	publication_date
type	string	OpenAlex work type (article, preprint, conference-paper, ...)	type

continued on next page

Table 17. `slim` (continued).

Column	Type	Definition	Source
`cited_by_count`	int32	Citations received, all years, as of the snapshot	`cited_by_count`
`primary_topic_id`	int32	Primary topic	`primary_topic.id`
`primary_subfield_id`	int32	Subfield of the primary topic	`primary_topic.subfield.id`
`primary_field_id`	int32	Field of the primary topic (17 = Computer Science)	`primary_topic.field.id`
`primary_domain_id`	int32	Domain of the primary topic	`primary_topic.domain.id`
`cs_any_topic`	bool	True if any of the work's topics, or its primary topic, is in field 17	`topics[].field.id`, `primary_topic.field.id`
`is_xpac`	bool	OpenAlex "extended publication and citation" flag (records of lower provenance)	`is_xpac`
`is_oa`	bool	Open-access flag	`open_access.is_oa`
`is_retracted`	bool	Retraction flag	`is_retracted`
`language`	string	ISO language code	`language`
`source_id`	int64	Source (venue) of the primary location	`primary_location.source.id`
`n_refs`	int32	Number of referenced works	length of `referenced_works`
`n_authors`	int32	Number of authors	`authors_count`, else length of `authorships`
`has_abstract`	bool	True if an inverted-index abstract is present	`abstract_inverted_index` $\neq$ null
`fwci`	float32	Field-weighted citation impact as computed by OpenAlex	`fwci`

Table 18. `dblp_publications` (`data/raw/dblp/dblp_publications.parquet`; view `dblp_raw`): one row per DBLP publication record of every year (`dblp_parse.py`). Record types `www` are skipped.

Column	Type	Definition	Source
`key`	string	DBLP record key, unique	element attribute `key`
`rec_type`	string	Record type: article, inproceedings, proceedings, book, incollection, phdthesis, mastersthesis	element name
`mdate`	string	Last modification date of the record	attribute `mdate`
`publtype`	string	Publication sub-type (e.g. informal, survey, withdrawn); null if absent	attribute `publtype`
`title`	string	Title, inner markup flattened to text	`<title>`
`year`	int32	Year	`<year>`
`authors`	list<string>	Author names in order	`<author>`
`author_pids`	list<string>	DBLP person identifiers of the authors	`<author pid=...>`
`editors`	list<string>	Editor names	`<editor>`
`journal`	string	Journal name (articles)	`<journal>`
`booktitle`	string	Proceedings or book title	`<booktitle>`
`volume`, `number`, `pages`	string	Bibliographic locators	`<volume>`, `<number>`, `<pages>`
`publisher`, `series`	string	Publisher and series	`<publisher>`, `<series>`

continued on next page

Table 18. `dblp_publications` (continued).

Column	Type	Definition	Source
`doi`	string	DOI when DBLP records one	`<ee>` with a doi.org URL
`ee`	list<string>	Electronic-edition URLs (DOI, arXiv, publisher)	`<ee>`
`url`	string	DBLP-internal URL (table of contents)	`<url>`
`crossref`	string	Key of the enclosing proceedings or book	`<crossref>`
`stream`	string	Venue stream: the key with its last path component removed (`conf/nips`, `journals/corr`); null if the key has no slash	derived from `key`

Table 19. `arxiv_cs` (`data/raw/arxiv/arxiv_cs.parquet`; view `arxiv_raw`): one row per arXiv record of set `cs` harvested by OAI-PMH with the `arXiv` metadata format (`arxiv_oai.py`); records returned in several monthly windows are kept once, the latest datestamp winning.

Column	Type	Definition	Source
`arxiv_id`	string	arXiv identifier without version	`<id>`
`datestamp`	string	OAI datestamp (last update of the record)	header `<datestamp>`
`created`	string	Date of the first submitted version	`<created>`
`updated`	string	Date of the latest version	`<updated>`
`title`, `abstract`	string	Title and abstract	`<title>`, `<abstract>`
`authors`	list<string>	Author names	`<authors>/<author>`
`categories`	list<string>	All categories (primary first, then cross-lists)	`<categories>`
`primary_category`	string	First category	first token of `<categories>`
`comments`, `journal_ref`	string	Author comments and journal reference	`<comments>`, `<journal-ref>`
`doi`	string	DOI declared by the authors	`<doi>`
`license`	string	Licence URL of the e-print	`<license>`
`report_no`, `msc_class`, `acm_class`	string	Report number, MSC and ACM classes	corresponding elements
`set_specs`	list<string>	OAI set memberships	header `<setSpec>`

Table 20. `acl_anthology` (`data/raw/acl/acl_anthology.parquet`; view `acl_raw`): one row per entry of the ACL Anthology BibTeX export with abstracts (`acl_parse.py`). All fields are strings copied from the BibTeX entry; braces are removed.

Column	Type	Definition	Source
`acl_id`	string	Anthology identifier (BibTeX citation key)	entry key
`entry_type`	string	BibTeX entry type (inproceedings, article, proceedings, ...), lower case	entry type
`title`, `author`, `editor`	string	Title, author list, editor list as written in BibTeX	fields
`booktitle`, `journal`	string	Venue	fields
`month`, `year`	string	Month and year (year is cast to integer in `acl_keys`)	fields

continued on next page

Table 20. `acl_anthology` (continued).

Column	Type	Definition	Source
`address`, `publisher`	string	Place and publisher	fields
`url`, `doi`	string	Anthology URL and DOI	fields
`pages`, `volume`, `number`	string	Locators	fields
`abstract`	string	Abstract when the export carries one	field

D.3. Normalised OpenAlex tables

`build_corpus.py` reads the full JSON record of every work retained by the snapshot filter (any topic in field 17 and publication year 2018–2026, or, in the second pass, membership of the list of DBLP- and arXiv-linked identifiers) and writes nine relational tables. The `works` table has one row per work; the eight others are one-to-many and are keyed by `work_id`. Nested JSON is flattened without loss of the fields used downstream; fields not listed here (for instance `sustainable_development_goals`, `mesh`, `concepts`) are not extracted.

Table 21. `works` (`compact/works.parquet`; view `oa_works`): one row per OpenAlex work of the CS extraction (`build_corpus.py`, schema `S["works"]`).

Column	Type	Definition	Source
`id`	int64	Work identifier	`id`
`doi`	string	DOI, lower case, prefix removed; null if empty	`doi`
`title`	string	Title; `display_name` when `title` is absent	`title`, `display_name`
`publication_year`	int32	Publication year	`publication_year`
`publication_date`	string	Publication date; OpenAlex sets `YYYY-01-01` when only the year is known (see `date_precision`, Table 27)	`publication_date`
`created_date`	string	Date the record entered OpenAlex (used for indexing-lag diagnostics)	`created_date`
`updated_date`	string	Last update, date part only	`updated_date[:10]`
`type`	string	OpenAlex work type	`type`
`type_crossref`	string	Crossref type when the record has one	`type_crossref`
`language`	string	ISO language code	`language`
`cited_by_count`	int32	Citations received, all years, at snapshot date	`cited_by_count`
`fwci`	float32	OpenAlex field-weighted citation impact	`fwci`
`citation_percentile`	float32	Citation-normalised percentile	`citation_normalized_percentile.value`
`is_top1`, `is_top10`	bool	Top 1 % / top 10 % flags of that percentile	`...is_in_top_1_percent`, `...is_in_top_10_percent`
`is_oa`	bool	Open-access flag	`open_access.is_oa`

continued on next page

Table 21. works (continued).

Column	Type	Definition	Source
oa_status	string	gold, green, hybrid, bronze, diamond, closed	open_access.oa_status
license	string	Licence of the best OA location, else of the primary location	best_oa_location.license, primary_location.license
is_retracted	bool	Retraction flag	is_retracted
is_paratext	bool	Paratext flag (front matter, tables of contents)	is_paratext
is_xpac	bool	Extended-provenance flag	is_xpac
n_refs	int32	Number of referenced works in the record	length of referenced_works
n_authors	int32	Number of authors	authors_count, else length of authorships
n_institutions	int32	Number of distinct institution identifiers over all authorships	authorships[].institutions[].id
n_countries	int32	Number of distinct country codes over all authorships	authorships[].countries
countries	list<string>	Sorted distinct country codes	authorships[].countries
institution_types	list<string>	Sorted distinct institution types (education, company, government, ...)	authorships[].institutions[].type
primary_topic_id	int32	Primary topic	primary_topic.id
primary_subfield_id	int32	Subfield of the primary topic	primary_topic.subfield.id
primary_field_id	int32	Field of the primary topic	primary_topic.field.id
primary_domain_id	int32	Domain of the primary topic	primary_topic.domain.id
cs_primary	bool	Primary topic in field 17	primary_topic.field.id
cs_any	bool	Any topic in field 17	topics[].field.id
source_id	int64	Venue of the primary location	primary_location.source.id
source_name	string	Venue name	primary_location.source.display_name
source_type	string	journal, conference, repository, ebook platform, ...	primary_location.source.type
host_organization	string	Publisher or host organisation of the venue	primary_location.source.host_organization_name
is_in_doaj	bool	Venue listed in DOAJ	primary_location.source.is_in_doaj
arxiv_id	string	arXiv identifier: from ids.arxiv; else from a DOI of the form 10.48550/arxiv.<id>; else from an arxiv.org/abs or /pdf URL among the locations, version suffix removed	ids.arxiv, doi, locations[].landing_page_url, locations[].pdf_url
pmid	string	PubMed identifier (tail of the URL)	ids.pmid
mag	string	Microsoft Academic Graph identifier	ids.mag
has_abstract	bool	An abstract could be reconstructed	abstract_inverted_index
has_fulltext	bool	OpenAlex has indexed full text	has_fulltext or has_content
n_locations	int32	Number of locations (hosting copies)	length of locations
indexed_in	list<string>	Indexes that list the work (crossref, pubmed, doaj, arxiv, ...)	indexed_in

continued on next page

Table 21. works (continued).

Column	Type	Definition	Source
referenced_works_count	int32	Reference count declared by OpenAlex; falls back to n_refs	referenced_works_count
n_grants	int32	Number of grant entries	length of grants
apc_paid_usd	int32	Article-processing charge paid, USD, when known	apc_paid.value_usd
is_authors_truncated	bool	Author list truncated by OpenAlex	is_authors_truncated
part	string	Snapshot part file of origin (updated_date=..._part_NNNN)	file name

Table 22. authorships (view oa_authorships): one row per (work, author position).

Column	Type	Definition	Source
work_id	int64	Work	id of the work
position	int32	Zero-based index in the authorship list	list index
author_position	string	first, middle or last	authorships[].author_position
author_id	int64	OpenAlex author	authorships[].author.id
author_name	string	Author display name	authorships[].author.display_name
orcid	string	ORCID URL when known	authorships[].author.orcid
is_corresponding	bool	Corresponding-author flag	authorships[].is_corresponding
institution_ids	list<int64>	Institutions of this authorship, in order	authorships[].institutions[].id
institution_names	list<string>	Their display names	...institutions[].display_name
institution_types	list<string>	Their types	...institutions[].type
countries	list<string>	Country codes of this authorship	authorships[].countries
n_raw_affiliations	int32	Number of raw affiliation strings	length of authorships[].raw_affiliation_strings

Table 23. Small one-to-many tables of the CS extraction: work_topics, work_keywords, references, abstracts, grants, counts_by_year (views oa_<table>). Every table carries work_id (int64) as its first column.

Column	Type	Definition	Source
work_topics: one row per topic of the work, at most three			
rank	int16	Zero-based position in the topic list (0 = primary)	list index
topic_id	int32	Topic	topics[].id
score	float32	OpenAlex topic score	topics[].score

continued on next page

Table 23. Small one-to-many tables (continued).

Column	Type	Definition	Source
subfield_id, field_id, domain_id	int32	Hierarchy of the topic	topics[].subfield.id, .field.id, .domain.id
work_keywords: one row per OpenAlex keyword			
keyword	string	Keyword display name	keywords[].display_name
score	float32	Keyword score	keywords[].score
references: one row per referenced work (the citation graph)			
ref_id	int64	Cited work identifier; may lie outside the CS extraction	referenced_works[]
abstracts: one row per work with an abstract			
abstract	string	Abstract text rebuilt by sorting the inverted index by position and joining with spaces	abstract_inverted_index
n_words	int32	Number of whitespace-separated tokens	derived
grants: one row per grant entry			
funder_id	int64	Funder	grants[].funder
funder_name	string	Funder name	grants[].funder_display_name
award_id	string	Award identifier	grants[].award_id
counts_by_year: one row per (work, calendar year) with citations received			
year	int16	Calendar year of the citing works	counts_by_year[].year
cited_by_count	int32	Citations received in that year	counts_by_year[].cited_by_count

Table 24. locations (view oa_locations): one row per hosting location of the work.

Column	Type	Definition	Source
work_id	int64	Work	id of the work
rank	int16	Zero-based position in the location list	list index
is_primary	bool	Landing-page and PDF URLs equal those of primary_location	derived
is_best_oa	bool	Landing-page URL equals that of best_oa_location	derived
source_id, source_name, source_type	int64, string	Venue of the location	locations[].source.id, .display_name, .type
landing_page_url, pdf_url	string	URLs	locations[].landing_page_url, .pdf_url
is_oa	bool	Location is open access	locations[].is_oa
license	string	Licence of this copy	locations[].license
version	string	submittedVersion, acceptedVersion or publishedVersion	locations[].version
is_accepted, is_published	bool	Version flags	locations[].is_accepted, .is_published

D.4. Key tables and linkage tables

`link_sources.py` derives from each source a *key table* that carries only the identifiers used for matching, in the shared conventions of Section D.1, and then builds three linkage tables by the cascade of Section 16.1. In every key table `ntitle` is the title in lower case with every run of characters other than `a-z` and `0-9` removed, and `tlen` is the length of the original title; title matching requires `tlen` ≥ 25 and a normalised title that is unique on both sides.

Table 25. Key tables (DuckDB tables `oa_keys`, `dblp_keys`, `arxiv_keys`, `acl_keys`).

Column	Type	Definition	Source
`oa_keys`: every work of `oa_works` (both passes)			
`oa_id`	BIGINT	Work identifier	`oa_works.id`
`doi`	VARCHAR	DOI	`oa_works.doi`
`arxiv_id`	VARCHAR	arXiv identifier, lower case, version suffix removed	`oa_works.arxiv_id`
`year`	INTEGER	Publication year	`oa_works.publication_year`
`ntitle`, `tlen`	VARCHAR, INTEGER	Normalised title and original title length	`oa_works.title`
`type`	VARCHAR	OpenAlex type	`oa_works.type`
`dblp_keys`: DBLP records with year 2018–2026 and type article, inproceedings, incollection, book, phdthesis or data			
`dblp_key`	VARCHAR	DBLP key	`dblp_raw.key`
`rec_type`, `year`	VARCHAR, INTEGER	Record type and year	`dblp_raw`
`doi`	VARCHAR	DOI	`dblp_raw.doi`
`arxiv_id`	VARCHAR	arXiv identifier extracted from an `arxiv.org/abs/` URL among `ee`, version removed; null when none	`dblp_raw.ee`
`ntitle`, `tlen`	VARCHAR, INTEGER	Normalised title and length	`dblp_raw.title`
`stream`	VARCHAR	Venue stream	`dblp_raw.stream`
`arxiv_keys`: every record of `arxiv_raw`			
`arxiv_id`	VARCHAR	Identifier, lower case	`arxiv_raw.arxiv_id`
`created`, `created_year`	VARCHAR, INTEGER	First-submission date and its year	`arxiv_raw.created`
`doi`	VARCHAR	Declared DOI	`arxiv_raw.doi`
`primary_category`, `categories`	VARCHAR, list	Categories	`arxiv_raw`
`ntitle`	VARCHAR	Normalised title	`arxiv_raw.title`
`acl_keys`: entries of type inproceedings or article			
`acl_id`	VARCHAR	Anthology identifier	`acl_raw.acl_id`
`year`	INTEGER	Year (`try_cast`; null if not numeric)	`acl_raw.year`
`doi`	VARCHAR	DOI, lower case	`acl_raw.doi`
`url`, `ntitle`, `entry_type`	VARCHAR	URL, normalised title, entry type	`acl_raw`

Table 26. Linkage tables and the second-pass identifier list. Each linkage table has one row per matched pair; the cascade inserts a pair only when neither side is already linked by an earlier method (for `link_arxiv_oa` and `link_acl_oa`, when the source-side key is not already linked and, for title matches, the OpenAlex side is not either).

Column	Type	Definition	Source
link_dblp_oa			
dblp_key, oa_id	VARCHAR, BIGINT	Matched DBLP record and OpenAlex work	key tables
method	VARCHAR	doi (equal DOIs); arxiv (equal arXiv identifiers, both sides unlinked); title_year (unique normalised titles, $\lvert\Delta\text{year}\rvert \leq 1$, and equal first-author surname when both sides have one)	cascade
link_arxiv_oa			
arxiv_id, oa_id	VARCHAR, BIGINT	Matched arXiv record and OpenAlex work	key tables
method	VARCHAR	arxiv_id (equal identifiers); doi; title_year (unique titles, $\lvert\Delta\text{year}\rvert \leq 2$ between arXiv creation year and OpenAlex year)	cascade
link_acl_oa			
acl_id, oa_id	VARCHAR, BIGINT	Matched ACL entry and OpenAlex work	key tables
method	VARCHAR	doi; url (an OpenAlex location whose landing page is aclanthology.org/<acl_id>); title_year (unique titles, $\lvert\Delta\text{year}\rvert \leq 1$)	cascade, oa_locations
needed_oa_ids: identifiers extracted in the second snapshot pass			
oa_id	BIGINT	Work present in oa_slim but absent from the CS extraction	oa_slim.id
why	VARCHAR	dblp_doi (DOI of an unlinked DBLP record) or arxiv_doi (DOI of an unlinked arXiv record)	derivation

The linkage report (`data/metadata/linkage_report.json`) records, for each source, the number of keys, the number of keys with a DOI or an arXiv identifier, the links by method, the number of distinct linked keys, the unlinked records by type, the unlinked DBLP records whose DOI nevertheless resolves in the slim table (the trigger for the second pass), the identifiers needed for the second pass by reason, the size of the slim table and the number of slim rows with a CS topic in 2018–2026. These are the quantities quoted in Part III through the `\Link*` macros.

D.5. The corpus table

`define_corpus.py` creates `corpus_works` (also exported to `data/processed/corpus/corpus_works.parquet`) from `oa_works`, adding the membership, dating, duplicate and eligibility columns of Table 27. The constants are those of the protocol: the observation window is 2018-01-01 to

2026-11-02 with the boundary 2022-11-30; the kept types are article, conference-paper, preprint, book-chapter, book, dissertation, report, review and software-paper; the hard-excluded types, which exclude even a DBLP-linked record, are editorial, erratum, retraction, peer-review, paratext, conference-abstract, reference-entry, dataset, software, supplementary-materials, libguides, letter, grant and standard; the generic deposit platforms of amendment A-002 are Zenodo, Figshare, IRDB, LA Referencia, Dialnet, DOAJ-as-source and OSTI. Every row of `oa_works` is present, including excluded ones, so that PRISMA counts can be recomputed from the flags.

Table 27. `corpus_works`: columns added to those of `works` (Table 21).

Column	Type	Definition	Source
`ntitle`	VARCHAR	Normalised title	`title`
`in_dblp`	BOOLEAN	The work is linked to a DBLP record	`link_dblp_oa`
`in_acl`	BOOLEAN	The work is linked to an ACL Anthology entry	`link_acl_oa`
`arxiv_id_linked`	VARCHAR	arXiv identifier of a linked arXiv record (any one, if several)	`link_arxiv_oa`, `arxiv_keys`
`arxiv_created`	VARCHAR	Earliest first-submission date among linked arXiv records	`arxiv_keys.created`
`arxiv_primary`	VARCHAR	Primary arXiv category of a linked record	`arxiv_keys.primary_category`
`arxiv_cs_primary`	BOOLEAN	`arxiv_primary` starts with `cs.`; null when no linked record	derived
`is_arxiv_doi`	BOOLEAN	The DOI is an arXiv DOI (`10.48550/arxiv.`)	`doi`
`duplicate_of`	BIGINT	For a preprint-like record that has a published twin: identifier of the twin; null otherwise	`dup_pairs`
`preprint_work_id`	BIGINT	For a published record that has a preprint twin: identifier of the preprint	`dup_pairs`
`first_date`	VARCHAR	First public appearance: the earliest of the publication date, the twin preprint's date and the arXiv creation date; when the publication date is year-only (`-01-01`) and a preprint or arXiv date exists, the earliest of those two instead	derived (eq. 3)
`version_relationship`	VARCHAR	On the published twin: `title_year` or `arxiv_id`, the relation by which the pair was found	`dup_pairs.relation`
`e1_outside_window`	BOOLEAN	Publication date null, before 2018-01-01 or after 2026-11-02	`publication_date`
`e2_type`	BOOLEAN	Type not in the kept list, unless DBLP-linked and not hard-excluded	`type`, `in_dblp`
`e3_xpac`	BOOLEAN	XPAC record not linked to DBLP	`is_xpac`, `in_dblp`
`xpac_kept_dblp`	BOOLEAN	XPAC record kept because DBLP-linked (A-001)	`is_xpac`, `in_dblp`
`e7_generic_deposit`	BOOLEAN	Primary source is a generic deposit platform and the record is neither DBLP- nor arXiv-linked (A-002)	`source_name`
`date_precision`	VARCHAR	`year` when the publication date is `YYYY-01-01` and no preprint or arXiv date exists; `month` otherwise	derived
`e4_no_title`	BOOLEAN	Title null or blank	`title`

continued on next page

Table 27. corpus_works (continued).

Column	Type	Definition	Source
e5_duplicate	BOOLEAN	The record is the preprint member of a duplicate pair	duplicate_of
e6_retracted	BOOLEAN	Retracted (flag, not an exclusion: retracted works stay eligible and are excluded from citation indicators)	is_retracted
cs_core	BOOLEAN	DBLP-linked, or CS primary topic, or CS topic with a cs. arXiv primary category	derived
cs_penumbra	BOOLEAN	Any CS topic but not cs_core	derived
eligible	BOOLEAN	Not E1–E5, not E7, and cs_core	derived (eq. 19)
period	VARCHAR	pre (before 2018-01-01), baseline (before 2022-11-30), post (to 2026-11-02), future, by first_date	derived
first_month	VARCHAR	YYYY-MM of first_date	derived

Two auxiliary tables accompany it. dup_pairs (columns preprint_id, published_id, preprint_date, published_date, relation) holds the one-to-one preprint-to-published pairs after the ranking columns rn_pub and rn_pre have been used to keep, for each published work, the earliest preprint, and for each preprint, the earliest published twin; a candidate pair is a preprint-like record (type preprint or arXiv DOI) and a non-preprint record in the window that passes E1–E4, with equal normalised titles of at least 25 characters and years at most two apart (title_year), or with the same arXiv identifier (arxiv_id). dblp_only_counts (year, rec_type, n, with_doi) counts the DBLP records of 2018–2026 that have no OpenAlex record after both passes. The PRISMA file prisma_counts.json stores the identified records per source, the links, each exclusion flag counted alone and sequentially in the order E1, E2, E3, E4, E5, E7, the remainder, the eligible counts by period, year, type, membership and date precision, and the strict core (DBLP-linked and CS primary); its keys are the \Prisma* macros of the manuscript.

D.6. Citation windows

bibliometrics/03_citation_impact.py builds work_citation_windows (also data/processed/corpus/work_citation_windows.parquet) for the eligible, non-retracted works. The yearly citation counts of a published work and of its merged preprint twin are summed, so that a citation to either version counts for the canonical node (Section 18.2); fixed windows are then formed from calendar years, and each window is normalised by the mean of the same window over works of the same publication year, type and primary subfield.

Table 28. `work_citation_windows`: one row per eligible, non-retracted work.

Column	Type	Definition	Source
`work_id`	BIGINT	Work	`corpus_works.id`
`year`	INTEGER	Publication year y	`publication_year`
`type`	VARCHAR	OpenAlex type, `unknown` when null	`type`
`sf`	INTEGER	Primary subfield, -1 when null	`primary_subfield_id`
`cited_by_count`	INTEGER	All-time citations at snapshot date (the canonical record only)	`cited_by_count`
`fwci`	DOUBLE	OpenAlex FWCI (secondary indicator)	`fwci`
`c12`	BIGINT	Citations received in years $\leq y+1$ (publication year and the following year), work plus merged preprint	`oa_counts_by_year`
`c36`	BIGINT	Citations received in years $\leq y+2$	idem
`c48`	BIGINT	Citations received in years $\leq y+3$	idem
`c12_complete`, `c36_complete`, `c48_complete`	BOOLEAN	The window closes in a complete calendar year: $y+1 \leq 2025$, $y+2 \leq 2025$, $y+3 \leq 2025$	derived
`ncs12`, `ncs36`	DOUBLE	`c12`, `c36` divided by the mean of the same quantity over works with the same `year`, `type` and `sf`; null when that mean is zero	derived (eq. 27)

The accompanying result tables are `citation_windows_yearly.csv` (per year: n, completeness flags, means and median of the windows, shares uncited at 12 and 36 months, the 99th percentile and maximum of `c36`, Gini coefficients of `c36` and `c12`, mean and median FWCI), `top_cited_by_year.csv` (the fifty works with the largest `c36` per year, with title, venue, type and the window counts) and `fwci_summary_yearly.csv`.

D.7. Citation-graph edge tables

`knowledge_graph/01_citation_graph.py` splits the references of eligible works into two edge tables according to whether the cited work is itself eligible. External edges are resolved through the slim table, which is why the slim table covers the whole snapshot: a reference to a work outside computer science is attributed a year, a type and a field without that work ever being extracted in full. References whose target is in neither table (identifiers absent from the snapshot) appear in neither edge table and are counted only through `n_refs`.

Table 29. Edge tables `cit_edges_internal` and `cit_edges_external` (`data/processed/graph/`).

Column	Type	Definition	Source
`cit_edges_internal`: both ends eligible			
`citing_id`, `cited_id`	BIGINT	Citing and cited work	`oa_references`

continued on next page

Table 29. Edge tables (continued).

Column	Type	Definition	Source
citing_year, cited_year	INTEGER	Publication years of the two ends	corpus_works.publication_year
cit_edges_external: citing work eligible, cited work not eligible but present in the snapshot			
citing_id, cited_id	BIGINT	Citing and cited work	oa_references
citing_year	INTEGER	Year of the citing work	corpus_works
cited_year	INTEGER	Year of the cited work	oa_slim.publication_year
cited_field_id	INTEGER	Primary field of the cited work	oa_slim.primary_field_id
cited_type	VARCHAR	Type of the cited work	oa_slim.type
cited_cs_any	BOOLEAN	The cited work has a CS topic (it is then a non-eligible CS record, e.g. outside the window)	oa_slim.cs_any_topic

The derived result tables are field_flows_yearly.csv (citing_year, kind internal or external, cited_field_id, n), cited_age_yearly.csv (citing_year, age = citing year minus cited year, n, over both tables) and reference_coverage_yearly.csv (year, n_works, n_refs_declared, n_internal, n_external). Category-level flows, co-classification and modularity tables (knowledge_graph/02_category_flows.py) join the internal edges with the classification tables below; the semantic edges of 04_semantic_edges.py (work_id, neighbour_id, cosine_similarity, rank) are computed from the embeddings of Section D.9; and the extracted method, dataset and claim nodes and typed edges of 03_extract_entities.py carry, for every edge, the work, the quoted span from the abstract, a model-reported confidence and a verification status (unverified, machine-verified when the span is found verbatim, human-verified after review), as the mandate requires.

D.8. Classification tables

taxonomy/classify.py first exports its input (-prepare) and then scores it (-run), with the ontology of Appendix A and the rule of equations (21)–(22). The population is the eligible corpus together with the penumbra, so that sensitivity analyses on the wider set need no second run.

Table 30. Classification input and output tables (data/processed/classification/).

Column	Type	Definition	Source
input/chunk=N/.parquet: one row per work with eligible or cs_penumbra*			
id	BIGINT	Work	corpus_works.id
year	INTEGER	Publication year	publication_year
title, abstract	VARCHAR	Text scored by the keyword rules (abstract null when absent)	corpus_works, oa_abstracts

continued on next page

Table 30. Classification tables (continued).

Column	Type	Definition	Source
topic_ids	list<int>	Topic identifiers in rank order	oa_work_topics
subfield_ids	list<int>	Subfield identifiers in rank order	oa_work_topics
arxiv_cats	list<string>	All categories of all linked arXiv records, flattened	link_arxiv_oa, arxiv_keys.categories
dblp_streams	list<string>	Distinct streams of the linked DBLP records	link_dblp_oa, dblp_keys.stream
chunk	INTEGER	Chunk index (default 100,000 rows per chunk)	derived
work_categories (cats/.parquet): one row per (work, assigned category)*			
work_id	int64	Work	input
category_id	string	Leaf identifier (Table 5)	ontology
score	float32	Total score $s_k(w)$, rounded to three decimals	eq. (21)
s_topic, s_subfield, s_arxiv, s_dblp, s_kw	float32	Capped contribution of each signal type (each in $[0,1]$)	eq. (21)
work_primary (primary/.parquet): one row per work with at least one category*			
work_id	int64	Work	input
primary_category	string	Argmax category with the tie-breaking of Section 17.1	derived
n_categories	int16	Number of assigned categories $\lvert C(w)\rvert$	derived
primary_score	float32	Score of the primary category	derived

A work with no category has no row in either output table; the number of such works is the difference between the works and classified fields of classification_report.json, which also records the ontology version and the timestamp of the run. The validation tables (taxonomy/validation/sample.parquet, ratings_<model>.jsonl, metrics.json, disagreements.csv) carry, per sampled work, the rule categories and primary, and per rater the label set, primary, confidence and rationale returned under the JSON schema of VALIDATION.md.

D.9. Embedding shards

embeddings/embed_corpus.py exports the text to embed (-prepare) and encodes it (-run) with a sentence-embedding model, by default SPECTER2 (Cohan et al., 2020; Singh et al., 2023) as served through sentence-transformers (Reimers and Gurevych, 2019), in shards that can be distributed over nodes.

Table 31. Embedding input and output files (data/processed/embeddings/).

File or column	Type	Definition	Source
input/shard=K/.parquet: eligible works with an abstract, ordered by identifier, 50,000 per shard by default*			
work_id	BIGINT	Work	corpus_works.id
text	VARCHAR	Title, a full stop and a space, then the first 2,000 characters of the abstract	title, oa_abstracts.abstract

continued on next page

Table 31. Embedding files (continued).

File or column	Type	Definition	Source
`shard`	INTEGER	Shard index K	derived
`<model_tag>/shard_KKKK.npy`			
array	float16, $n \times d$	One L2-normalised embedding per text, in input order; d is the model's output dimension; maximum sequence length 256 tokens, batch 128 by default	model output
`<model_tag>/shard_KKKK.ids.npy`			
array	int64, n	The `work_id` of each row of the embedding array	input
`<model_tag>/manifest_<host>.json`			
`model`, `max_seq_length`, `device`, `host`	–	Model name, sequence limit, `mps` or `cpu`, host name	run parameters
`shards[K]`	–	Per shard: number of texts, seconds taken, timestamp	run

The model tag is the model name with `/` replaced by `__`; a second model produces a second directory with the same layout, which is how the comparison of embedding approaches required by the mandate is organised. Shards already present are skipped on resumption, so a shard computed on one node and copied to another is never recomputed.

E. Reproducibility

The mandate requires that the corpus be reproducible from immutable raw snapshots and that every quantitative claim trace to a script, an input artefact and a result file. This appendix records what a reader needs to re-run the pipeline: the computing environment and the locked package versions, the ordered steps with their commands, inputs, outputs and measured wall times, the mechanisms by which provenance is kept (manifests, hash ledgers, verbatim raw pages, API ledgers), the incidents that occurred during the first run and what they imply for a second one, the protocol amendments, and the licence of each source together with what is and is not redistributed. The runbook (`reproducibility/RUNBOOK.md`) is the operational version of this appendix and is updated whenever a step changes; where the two differ, the runbook is more recent.

E.1. Environment

Everything described in the runbook was executed on one node of the MacLustr cluster, M3U96b (Mac Studio, Apple M3 Ultra, 28 cores, 96 GB of unified memory, 103,079,215,104 bytes as recorded by the environment capture), in the working directory `~/wp44_uqo`, under Python

3.12.15 in a virtual environment created with `uv` from `requirements.txt` and then frozen to `reproducibility/environment_M3U96b.txt` (92 lines: 88 pinned packages, the interpreter version, the operating-system version string, the processor and the memory size). The operating-system version string captured in that file is 26.4, whereas the runbook header states macOS 27.0; the discrepancy is recorded here rather than resolved, since the node was updated between captures and neither value affects the computation. Two further nodes were used after the incidents of Section E.4: M5M36 for the second snapshot pass and M4M64c and M3U96a for embedding shards; they run the same locked environment. Table 32 lists the packages on which results depend; the complete list is the lock file. No step requires credentials except the two LLM raters of the taxonomy validation and the knowledge-graph extraction (an Anthropic API key read from a file outside the repository, never stored in it).

Table 32. Key packages of the locked environment (`reproducibility/environment_M3U96b.txt`, Python 3.12.15).

Role	Package	Version
Analytical database	`duckdb` (Raasveldt and Mühleisen, 2019)	2.0.0
Columnar I/O	`pyarrow`	26.0.0
Data frames	`polars` / `pandas`	2.1.0 / 3.1.0
Numerics	`numpy` / `scipy`	2.5.4 / 1.18.1
Statistics, time series	`statsmodels` / `formulaic`	0.15.0 / 1.2.2
Change-point detection	`ruptures` (Truong et al., 2020)	1.1.10
Machine learning	`scikit-learn`	1.9.1
Embeddings	`torch` / `transformers` / `sentence-transformers` (Reimers and Gurevych, 2019)	2.15.0 / 5.22.1 / 6.2.0
Model hub client	`huggingface-hub`	2.4.0
Dimensionality reduction, clustering	`umap-learn` / `hdbscan` / `pynndescent`	0.5.12 / 0.8.44 / 0.7.0
Networks	`networkx` / `igraph`	3.7 / 1.0.0
Parsing	`lxml` / `orjson` / `zstandard` / `regex`	6.1.3 / 3.13.0 / 0.25.0 / 2026.9.29
Ontology file	`pyyaml`	6.0.3
HTTP	`requests` / `httpx2`	2.34.2 / 2.16.0
LLM raters, extraction	`anthropic`	2.4.0
Figures	`matplotlib`	3.11.2
Tests	`pytest`	9.2.0

E.2. Steps, commands and measured times

Table 33 reproduces the runbook's step table. Commands are given relative to the repository root with the virtual environment active; ... abbreviates arguments that are spelled out in the runbook. Wall times are those measured during the first run and are reported as orientation, not as guarantees: the snapshot download depends on network throughput (about 108 MB/s was sustained from the node with eight parallel streams), and the steps marked "to be measured" had not completed at the time the runbook was last updated because of the node freezes described below. Steps 1–5 are pure acquisition and can be run independently and in any order; steps 6–10 are sequential; step 9 is a second acquisition pass whose need is discovered at step 8 and which then repeats steps 6–8 and 10–11 with the suffix `_p2` (`ingestion/run_pass2_chain.sh`, which waits for a flag file written when the pass-2 outputs have been copied back to the node). The API increment of Table 36 (works published after the snapshot cut-off) is converted to the snapshot's part format by `ingestion/ingest_api_increment.py` and normalised by the same step 6; it is not yet a numbered step of the runbook.

Table 33. Pipeline steps (`reproducibility/RUNBOOK.md`), with the command, the inputs, the outputs and the wall time measured on M3U96b.

#	Step and command	Inputs	Outputs	Wall time
1	OpenAlex works snapshot, pass 1 (CS filter + slim table): `python ingestion/openalex_snapshot.py -out data/raw/openalex/works -manifest data/raw/openalex/works_manifest_2026-10-14.json -workers 8`	public S3 bucket, release 2026-10-14	`works/cs/*.jsonl.zst` (15,490,159 records), `works/slim/*.parquet` (464,765,485 rows), `ledger.sqlite`	1 h 46 min at ≈108 MB/s
2	OpenAlex entities: `python ingestion/openalex_entities.py -out data/raw/openalex`	S3	`topics/`, ..., `institutions/`; `entities_ledger.json`	5 min
3	DBLP: `python ingestion/dblp_parse.py -xml data/raw/dblp/dblp.xml.gz -out data/raw/dblp`	dump of 2026-11-02, MD5 verified	`dblp_publications.parquet` (8,840,866 records), parse report	4 min 45 s
4	arXiv OAI-PMH: `python ingestion/arxiv_oai.py -out data/raw/arxiv -from 2018-01-01 -until 2026-11-02`	OAI feed, set `cs`	raw XML pages (758), `arxiv_cs.parquet` (915,115 records)	≈1 h
5	ACL Anthology: `python ingestion/acl_parse.py -bib data/raw/acl/anthology+abstracts.bib.gz -out data/raw/acl`	BibTeX export	`acl_anthology.parquet` (138,796 entries)	10 s
6	Normalise: `python ingestion/build_corpus.py -cs-dir data/raw/openalex/works/cs -out data/processed -workers 12`	step 1	`tables/<table>/*.parquet`	2 min 10 s

continued on next page

Table 33. Pipeline steps (continued).

#	Step and command	Inputs	Outputs	Wall time
7	Compact: `ulimit -n 65536; python ingestion/compact_tables.py ~/wp44_uqo`	step 6	`compact/<table>.parquet`	30 s
8	Link: `python ingestion/link_sources.py -root ~/wp44_uqo`	steps 3–7	`wp44.duckdb` with `link_*`; `data/metadata/linkage_report.json`; `needed_oa_ids.csv`	10 s
9	Second pass (DBLP- and arXiv-linked works outside the CS-topic filter): `python ingestion/openalex_snapshot.py -out data/raw/openalex/works_pass2 -manifest ...-workers 8 -id-list data/metadata/needed_oa_ids.csv`, then `ingestion/run_pass2_chain.sh` (steps 6–8 with suffix `_p2`, then 10–11)	S3, step 8	`works_pass2/cs/*.jsonl.zst`, `compact/<table>_p2.parquet`	≈1 h 50 min
10	Corpus definition: `python systematic_review/define_corpus.py -root ~/wp44_uqo`	step 8	`corpus_works`, `data/metadata/prisma_counts.json`	2 min
11	Volume, collaboration, citations, graph: `bibliometrics/01_volume_series.py`, `02_collaboration.py`, `03_citation_impact.py`, `knowledge_graph/01_citation_graph.py`	step 10	`results/bibliometrics/*.csv`, `results/graph/*`, `data/processed/graph/*`	minutes
12	Classification: `python taxonomy/classify.py -root ...-ontology taxonomy/ontology.yaml -prepare -run` (in practice `taxonomy/run_classify_shards.sh`, see incident I-3)	steps 8, 10; `ontology.yaml`	`classification/cats/*.parquet`, `classification/primary/*.parquet`, report	to be measured
13	Validation sample and raters: `python taxonomy/validate_sample.py -root ...-draw; -rate -model claude-haiku-5-5; -rate -model claude-sonnet-5-5; -metrics`	step 12	`taxonomy/validation/*`	to be measured
14	Embeddings: `python embeddings/embed_corpus.py -root ...-prepare; -run -model allenai/specter2_base` (sharded over nodes)	step 10	`embeddings/<model>/shard_*.npy`	≈300 texts/s per M3 Ultra
15	Structural change, emerging detection, figures, tables: `econometrics/run_structural_change.py`, `experiments/emerging_detection.py`, `figures/make_figures.py`, `tables/make_numbers.py`	steps 11–14	`results/econometrics`, `results/emerging`, `figures/`, `manuscript/tables/numbers.tex`	minutes
16	Manuscript: `cd manuscript && make` (`latexmk`)	all	`manuscript/main.pdf`	—

The test suite (`python -m pytest -q tests/`) covers the structural-change toolkit and the

classification scoring rule (tests/test_structural_change.py, tests/test_classify_rules.py); the second fixes the implemented aggregation of equation (21) against the toy ontology it declares, so that a change in classify.py that altered the rule would fail before any corpus were classified.

E.3. Provenance mechanisms

Provenance is kept at the level of files and jobs, not of individual records, by six mechanisms.

Manifests and hash ledgers for the OpenAlex snapshot. The snapshot's own manifest (works_manifest_2026-10-14.json) lists every part file with its content length and record count. The ingestion script keeps a SQLite ledger (works/ledger.sqlite) with a files table (one row per part: URL, local name, expected length, manifest record count, status, SHA-256 of the gzip bytes, records parsed, records retained by the filter, lines that failed to parse, download and total time, number of attempts, start and end timestamps, error text) and a runs table (start time, manifest date, manifest record and byte totals, number of workers, host name and command line of every invocation). A part is downloaded with resume and retries until its size equals the manifest's, hashed, parsed, and only then marked done; outputs are written to a temporary name and renamed atomically, so an interrupted run leaves no partial output that a resumed run would mistake for a complete one. The raw gzip files are deleted after a successful parse unless -keep-raw is given: the snapshot is a public, versioned release, and the manifest together with the per-file hashes allows any later copy to be verified against what was ingested. For the first pass the ledger records 1,898 of 1,898 files done, 0 parse errors, and a sum of per-file record counts equal to the manifest total.

Entity ledger. The small entity snapshots (topics, subfields, fields, domains, sources, institutions, publishers, funders, concepts) are listed in entities_ledger.json with, per file, the entity, the local path, the byte count, the SHA-256 and the status.

DBLP checksum. The DBLP dump is distributed with an .md5 file; the downloaded dblp.xml.gz was verified against it (ff1c424c3b4cef36582f647c36df27c0). The parse report (dblp_parse_report.json) additionally records the input path, its size in bytes, an MD5 of its first GiB (the file is 1.12 GB, so this digest is not the published one and is a convenience for detecting a changed input), the parse timestamp, per-type and per-year record counts, and timings.

Verbatim OAI-PMH pages. The arXiv harvest stores every XML response page as received under data/raw/arxiv/raw/<window>/page_NNNN.xml (758 pages for the monthly windows from 2018-01 to 2026-11), together with a harvest state file that allows resumption by window and

a parse report; the Parquet table can be rebuilt from the pages alone with `-parse-only`. Because OAI selects by record *datestamp*, a record updated in several windows appears in several pages; the parser keeps the latest.

API ledger. The OpenAlex API increment (works with a CS topic published between 2026-10-01 and 2026-11-02, fetched after the snapshot) stores every JSON page verbatim under `api_increment/pages/` and a ledger (`api_increment/ledger.json`) with the slices requested (day $\times$ CS-primary or not), the number of calls and the cost charged against the public budget; without an API key the harvest is paged by day with at most 10,000 results per slice and spread over several days.

Step reports and generated numbers. Each processing step writes a machine-readable report: `linkage_report.json` (counts of keys, links by method, unlinked records by type, identifiers needed for the second pass), `prisma_counts.json` (every PRISMA stage, with an identity check that the records loaded equal the sum of sequential exclusions and the remainder), `classification_report.json` (ontology version, works scored, works classified, timestamp), one embedding manifest per host (model, maximum sequence length, device, and per shard the number of texts, the seconds taken and the timestamp), and the validation metrics. The macros of `manuscript/tables/numbers.tex` are generated from these files by `tables/make_numbers.py`; a number whose source file is missing yields an undefined macro and therefore a compilation error, which is intended: the prose cannot cite a number that no result file contains.

E.4. Incident log and its implications

Six incidents were recorded during the first run (`reports/INCIDENTS.md`); none caused data loss, because every step is resumable from its ledger or its shard files, but three of them changed how the pipeline must be run and are therefore part of the reproduction instructions.

Three implications follow for anyone reproducing the work. First, the parallelism in the scripts assumes the `spawn` start method of macOS: worker state must be built in initialisers, and the classifier should be run as independent shard processes rather than through a pool (`classify.py -shard i/N`); on Linux, where `fork` is the default, the pool mode may work but has not been exercised. Second, the per-part Parquet layout exceeds default file-descriptor limits; the compaction step is mandatory, not an optimisation. Third, the two freezes of the primary node occurred under concurrent download, parse, GPU and database load, and their cause was not established from the system logs; the operational rule adopted afterwards (one heavy job per node at a time) costs wall time but is the configuration under which the reported outputs were produced, and the measured times of Table 33 assume it. Because every job writes its outputs atomically and records what it

Table 34. Incident log of the first run (`reports/INCIDENTS.md`; times in Eastern Time, 2026-11-03).

#	When	What happened	Impact and recovery
I-1	≈15:45	Second snapshot pass with `-id-list`: worker processes had an empty identifier set, because the macOS `spawn` start method does not inherit module globals	68 part files marked in error, no data loss; a pool initialiser now loads the identifier set in each worker; ledger reset and rerun
I-2	≈15:50	DuckDB "too many open files" when reading 1,898 Parquet files per table	Linkage step failed; `compact_tables.py` (one file per table) and `ulimit -n 65536`
I-3	≈16:30	`multiprocessing.Pool` with pyarrow in spawned workers hung (0 % CPU) in the classifier	Test stalled 10 min; shard mode: independent processes started by `run_classify_shards.sh`
I-4	≈16:58	M3U96b froze (no ping, SSH "host is down") while running the second pass (8 download-and-parse workers), the SPECTER2 embedding job on the GPU and earlier DuckDB jobs concurrently	Second pass (≈1,400 of 1,898 files done), the chained corpus v2 and classification, and the node's embedding share interrupted; other embedding nodes unaffected; all jobs resumable; rule adopted: never run ingestion, embeddings and classification concurrently on one node
I-5	17:16	M3U96b froze again two minutes after the second pass was restarted alone on the rebooted node	Pass 2 moved to the idle node M5M36 (10 workers, ledger copied); M3U96b reserved for DuckDB and classification; the chain now waits for a flag raised when the pass-2 outputs are copied back; Apple diagnostics show performance auto-bug-capture reports, no kernel panic found
I-6	17:17	M4M64a froze a second time within two minutes of restarting its embedding share (batch 32); M3U96b rebooted in the same minute while running pass 2 alone	M4M64a embedding share lost again; M4M64a and M3U96b withdrawn from GPU and bulk-download work; shard ranges reassigned to M4M64c and M3U96a; pass 2 on M5M36

has completed, the state of the pipeline after an interruption is recoverable from the ledgers alone, and the outputs of a resumed run are identical to those of an uninterrupted one for all deterministic steps (ingestion, normalisation, linkage, corpus definition, classification). The embedding step is the exception: shards were computed on several nodes and devices, and the per-host manifests record for each shard where it was computed, so that any numerical difference between devices can be traced; the embeddings are stored in half precision and are not claimed to be bit-identical across hardware.

E.5. Protocol amendments

The protocol was frozen on 2026-11-02 before any analysis of the corpus, and every later change is an amendment with an identifier, a reason and an approver (`protocol/amendments.csv`). All six amendments of Table 35 are approved by the executing agent and pending the principal investigator's signature at the checkpoint of the corresponding phase; the manuscript marks the affected conclusions accordingly.

Table 35. Protocol amendments (`protocol/amendments.csv`).

Id	Date	Phase	Change	Reason
A-000	2026-11-02	0	§14 of the mandate completed from §1–§13	The pasted mandate was truncated by the terminal
A-001	2026-11-03	1	Eligibility type list corrected: conference-paper and software-paper are kept; DBLP-linked records are kept whatever their type unless hard-excluded; XPAC records kept when DBLP-linked (`xpac_kept_dblp`)	The first run excluded 2.5 M conference-paper records (1.09 M DBLP-linked) through an outdated type list
A-002	2026-11-03	1	Records whose primary source is a generic deposit platform are excluded unless DBLP- or arXiv-linked (E7); `date_precision` added, monthly series use month-precision records only	Zenodo alone contributed 138,533 otherwise eligible 2026 records (deposits of slides, datasets and reports typed as articles); 18–28 % of records carry a year-only date
A-003	2026-11-03	2	LLM rating budget for the taxonomy validation fixed at 25 USD (two raters, structured outputs)	Mandate §7 requires agent-based validation with human checkpoints; the PI rates 200 works
A-004	2026-11-03	3	Knowledge-graph extraction from abstracts of the high-priority set with `claude-sonnet-5-5`, provenance spans and machine verification of spans; budget 150 USD	Mandate §6.4 requires provenance, confidence and verification status on every inferred relationship
A-005	2026-11-03	1	Exclusion code E7 redefined as the generic-deposit exclusion; the CS penumbra is a membership flag, not an exclusion code	The Part IV writer found the protocol and code labels inconsistent

E.6. Sources, licences and what is redistributed

Table 36 lists each raw source as obtained, with its size, integrity record and licence. Nothing under `data/raw/` or `data/processed/` is committed to the repository (`.gitignore` excludes both, together with every Parquet and JSONL file, except the provenance `README` and the metadata directory); the copies live on the compute node. What the released repository contains is the code, the ontology and its audit table, the manifests, ledgers and hashes, the PRISMA counts and the linkage report, the aggregated result tables, the figures, the generated number macros and the manuscript (protocol §P8). Record-level derived tables may be released where the source licence permits: OpenAlex is CC0, DBLP is ODC-BY 1.0, arXiv metadata obtained through OAI-PMH is under the CC0 terms of that feed, and the ACL Anthology is CC BY 4.0; a derived record table therefore carries the attribution required by ODC-BY and CC BY for the DBLP and ACL fields it contains. Full texts are never redistributed and were never obtained through paywalls. The Semantic Scholar Academic Graph was not used: bulk access requires an API key that was not available and anonymous calls were refused, and the mandate's instruction to record inaccessible data explicitly is satisfied by this note; influential-citation and citation-intent signals derived from that graph are consequently absent from the analyses.

Table 36. Raw sources as obtained (`data/raw/README.md`), with integrity record and licence.

Source	What was obtained, when, size	Integrity	Licence
OpenAlex works snapshot, release 2026-10-14	`data/jsonl/works/`: 1,898 part files, 464,765,485 works, 675.3 GB gzip; streamed 2026-11-02 21:19–23:05 ET over HTTPS with 8 parallel streams; retained: CS subset 20 GB (15,490,159 records) and slim table 9.3 GB (all works)	manifest; per-file content length and SHA-256 in `works/ledger.sqlite` (1,898/1,898 ok, 0 parse errors, record total = manifest total)	CC0
OpenAlex entity snapshots	topics 4,516; subfields 252; fields 26; domains 4; sources 268,629; institutions 143,455; publishers 10,713; funders 41,799; concepts 65,026; 2026-11-02; 0.6 GB	`entities_ledger.json` (SHA-256 per file)	CC0
OpenAlex API increment	works with any CS topic, publication date 2026-10-01 to 2026-11-02, sliced by day and CS-primary status, basic paging $\leq$10,000 per slice; from 2026-11-03 (public budget 0.10 USD/day, several days)	raw JSON pages; `api_increment/ledger.json` (counts, cost)	CC0
DBLP	`dblp.xml.gz` (2026-11-02 22:39 UTC), `dblp.dtd`, `.md5`; 1.12 GB	MD5 `ff1c424c...27c0` verified; parse report (8,840,866 records)	ODC-BY 1.0
arXiv OAI-PMH	set `cs`, metadata format `arXiv`, datestamps 2018-01-01 to 2026-11-02 in monthly windows, raw XML pages kept; 2026-11-02/03	758 pages, 915,115 unique records; `harvest_state.json`, parse report	CC0 (metadata)
ACL Anthology	`anthology+abstracts.bib.gz`; 2026-11-02; 46.7 MB, 138,796 entries	SHA-256 `44c3f7e9...58e3`	CC BY 4.0

E.7. Minimal reproduction sequence

A reader with a macOS or Linux machine, disk for the retained outputs (the CS subset and the slim table occupy about 30 GB; the 675.3 GB of gzip parts are streamed and deleted after parsing unless `-keep-raw` is given), memory for the DuckDB steps as configured (the scripts set `memory_limit` to 60–70 GB, to be lowered on smaller machines at the cost of time) and no credentials can reproduce the corpus and the bibliometric tables by creating the environment (`uv venv -python 3.12 .venv && uv pip install -r requirements.txt`), downloading the manifest of the 2026-10-14 release, and running steps 1–11 of Table 33 in order, with the second pass of step 9 between steps 8 and 10. The snapshot release is versioned, so a later release will not reproduce the counts; the manifest date is the parameter that pins them. Steps 12 and 14 need no credentials but benefit from several nodes; step 13 and the knowledge-graph extraction of amendment A-004 need an Anthropic API key and a budget of 25 and 150 USD respectively, and their outputs (ratings, extracted edges) are released so that the downstream statistics can be recomputed without re-rating. Step 15 regenerates every figure, table and number macro from the result files, and step 16 compiles the manuscript; a missing result file surfaces as an undefined macro at compilation, which is the intended way of detecting that a number has no source.