



MDMC

Master in Data Management
and Curation

MASTER IN DATA MANAGEMENT AND
CURATION

Building a FAIR-aligned
data-processing pipeline for
investigating research artifact
availability in high-energy physics
studies

Supervisor(s):

Dr. Maria Teresa LEO

Candidate:

Jacopo ZUPPA

2025–2026





This course has been funded by the European Union – NextGenerationEU within the projects:

- *NFFA-DI cod. IR0000015 - Missione 4, “Istruzione e Ricerca” – Componente 2, “Dalla ricerca all’impresa” – Linea di investimento 3.1, “Fondo per la realizzazione di un sistema integrato di infrastrutture di ricerca e innovazione” – Azione 3.1.1, “Creazione di nuove IR o potenziamento di quelle esistenti che concorrono agli obiettivi di Eccellenza Scientifica di Horizon Europe e costituzione di reti” (CUP B53C22004310006)*

[NFFA-DI : Nano Foundries Fine Analysis - Digital Infrastructure](#)

- *EFC cod. SSU2024 – 00002 - Missione 4, “Istruzione e Ricerca” – Componente 1, “Potenziamento dell’offerta dei servizi all’istruzione: dagli asili nido all’ università” – Linea di investimento 3.4, “Didattica e competenze universitarie avanzate” – Sub-Investimento, “Rafforzamento delle scuole universitarie superiori” (CUP G97G24000100007)*

[EFC : Educating Future Citizen](#)



Author's Declaration

I, Jacopo Zuppa, declare that this thesis entitled, 'Building a FAIR-aligned data-processing pipeline for investigating research artifact availability in high-energy physics studies' and the work presented therein are my own.

I certify that:

- This work was performed wholly or principally during the MDMC internship at SISSA Medialab S.r.l.
- If any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have relied on the published work of others, this is always clearly attributed.
- Where the thesis is based on work I have done in collaboration with others, I have made clear exactly what was done by others and what I contributed.
- I acknowledge the use of ChatGPT (<https://chatgpt.com/>) and Claude (<https://claude.ai/>) to provide information for background research, for code optimization and debugging, to generate scripts for data visualization, and for some conceptual refinements in the text.
- Where I have quoted from the work of others, the source is always cited. Except for such quotations, this thesis is entirely my own work.
- I have acknowledged all major sources of help.

Acknowledgments

I would like to thank my supervisor Dr. Maria Teresa Leo for the support, professionalism and interest shown during the thesis project, alongside Dr. Cristiana Prever, the editorial office, the development team and all other people at SISSA Medialab for creating such a welcoming and nurturing environment during my internship.

I would also like to thank my coursemates for the lovely weeks spent together in class and the teaching staff of the Master for the work they do every day for offering such a great learning experience. It has been life-changing.

Finally, I would like to thank my family for the constant support over the years. I could have never even dreamed of the opportunities I have had in my life without them.

Abstract

In the current thesis project, a FAIR-aligned data-processing pipeline was developed to extract and classify Data Availability Statements (DAS) and Code Availability Statements (CAS) in high-energy physics scientific publications. The pipeline was designed as a reusable tool organized around five modules handling ingestion of publications, conversion to text format, extraction, normalization and classification of availability statements. In total, 17,185 documents ($n = 8,511$ for 2024, $n = 8,674$ for 2025) were processed. Out of these, 17,076 were scientific studies analyzed to verify the inclusion of availability statements and classify their content. Results showed that 63.5% of the publications analyzed included a DAS section, while only 31.1% included a CAS section. Nonetheless, year-specific results showed a marked increase in the inclusion of both DAS and CAS sections in 2025 publications, suggesting possible adaptation to newly introduced open science policies. Classification results showed that a majority of studies declared no data and no code associated with the manuscript. When looking at studies that clearly reported associated data or code, a majority of studies declared that such artifacts were only available upon request. Despite this, year differences in category shares showed an increase in studies reporting both data (+15.8 pp) and code (+16.2 pp) as openly accessible in 2025 publications, accompanied by a decrease in studies reporting these artifacts as only available upon request. This might suggest a general increase in open availability of research artifacts in 2025 compared to 2024. Furthermore, data exploration revealed possible ambiguities around what is considered the proper subject of DAS and CAS sections. For code in particular, authors seem to report availability and usage of different types of software artifacts in these sections. This led to the development of a taxonomy differentiating author-produced code from external software tools in order to adequately classify CAS sections. Methodological and policy-related implications of these findings are discussed.

Contents

1	Introduction	7
2	Methods	10
2.1	General Software Architecture	10
2.2	Data Management and FAIR Compliance	11
2.2.1	Registries	11
2.2.2	FAIR Compliance	13
2.3	Software Modules	14
2.3.1	Data ingestion	14
2.3.2	PDF conversion	15
2.3.3	Section extraction	15
2.3.4	Data normalization	16
2.3.5	Section classification	17
2.4	Schemata Development	18
2.4.1	DAS Classification Schema	19
2.4.2	CAS Classification Schemata	20
2.5	Data Source and Sampling	23
2.6	Statistical Validation	23
3	Results	25
3.1	Data Availability Statements	25
3.1.1	DAS Presence	25
3.1.2	DAS Classification	25
3.2	Code Availability Statements	28
3.2.1	CAS Presence	28
3.2.2	CAS Classification	28
3.3	Statistical Validation	32
4	Discussion	33
4.1	Analytical Results and Study Implications	33

4.2	Limitations	36
4.2.1	Software Architecture	36
4.2.2	Data Sampling and FAIR Limitations	37
4.3	Future Developments and Considerations	37
4.3.1	Software Refactoring	37
4.3.2	Availability Statements and Schemata	38
5	Conclusion	39
	Bibliography	40
A	Classification Schemata	44
A.1	Data Availability Statement Schema	44
A.2	CAS Manuscript Code Availability Schema	49
A.3	CAS External Tool Availability Schema	57
B	Registry Schema and Pipeline Metadata	61
B.1	Registry Overview	62
B.2	Provenance Links	62
B.3	Archive Paths	63
C	Additional Result Figures	64

Chapter 1

Introduction

Over the last decades, scientific research came under scrutiny due to a flood of reports testifying to the limited reproducibility and replicability of studies across research fields [1, 2, 3, 4], coupled with scathing criticisms of commonplace scientific practices [5]. This situation raised considerable concerns due to the importance given to reproducibility for scientific research, emphasized in the epistemological literature of the past and present [6, 7]. Although often used interchangeably, the two notions of “reproducibility” and “replicability” serve different purposes in scientific discourse. Reproducibility in particular refers to testing the reliability of prior findings by obtaining the same results using the same input data, computational steps, methods, code and conditions of analysis¹ [8, 9].

Given this definition, availability of research artifacts like data and code can be considered a fundamental prerequisite to reproduce scientific findings [9]. The lack of availability of these artifacts is therefore considered problematic and several findings show how the unavailability of data and code is an important barrier to making science transparent and reproducible [10, 11, 12, 13].

In light of the doubts around transparency, reproducibility and fairness, the concept of open science became increasingly prominent in discussions on needed changes in scientific research and practice. Open science can be defined as “transparent and accessible knowledge that is shared and developed through collaborative networks” [14, p. 7] and generally, as a set of principles and practices to make scientific research more accessible and the production of scientific knowledge more “inclusive, equitable and sustainable” [15]. Researchers and stakeholders within the open science movement have called for a change in scientific practices and have proposed guidelines to enhance

¹On the other hand, replication refers to testing the reliability of prior findings by obtaining consistent results using different data.

openness and reproducibility, greatly emphasizing the need for better data sharing [16, 17, 18]. Furthermore, major institutions and granting agencies enacted policies favoring a transition toward practices in line with the idea of open, transparent and reproducible science [19, 20, 21, 22].

One such major funding institution has been the Sponsoring Consortium for Open Access Publishing in Particle Physics (SCOAP³). The consortium was established in 2014 from a partnership that currently includes more than three thousand libraries, funding agencies and research centers and is headed by the European Organization for Nuclear Research (CERN). Its main objective is that of converting “key journals in the field of High-Energy Physics to Open Access” at no cost for authors [23]. In order to guarantee the open access publication of scientific articles, SCOAP³ funds major publishers in the field to cover Article Processing Charges. Funded articles are then stored and made freely available in the SCOAP³ Repository, which counts more than 80 thousand published articles at the time of writing.

For the contractual period 2025-2027, the consortium introduced a major policy change in their funding model. In order to incentivize the adoption of open science practices, SCOAP³ proposed a mechanism of adjustment of funds granted to publishers. Essentially, publishers are comparatively evaluated on the adoption of “Open Science Elements”, a set of pre-defined publishing practices central to an open science approach [24].² Two such elements identified by SCOAP³ are Dataset Linkage and Software Linkage. Ideally, these two elements would require data and code associated with a study funded by SCOAP³ to be made publicly available. In cases where this might not be possible, SCOAP³ introduced the minimum, mandatory requirement of including Data Availability Statements (DAS) and Code Availability Statements (CAS) in published articles. These two statements refer to short sections at the end of a manuscript where authors specify information about the availability of data and code associated with their manuscript, and that are also included in the publication metadata publishers share with scientific indexing and abstracting services.

In the context of these policy changes, the development of scalable methods for detecting and analyzing availability statements becomes an important technical and empirical task. An analysis of the presence of CAS and DAS could indicate whether minimum reporting requirements are being adopted in

²Publishers whose funding will be affected by this policy and their respective journals are: APS (Physical Review C, Physical Review D, and Physical Review Letters), Elsevier (Nuclear Physics B, Physics Letters B), IOP (Chinese Physics C), Jagiellonian University (Acta Physica Polonica B), Oxford University Press (Progress of Theoretical and Experimental Physics), Springer Nature (European Physical Journal C, Journal of High Energy Physics) and Wiley (Advances in High Energy Physics).

recently published work. Furthermore, these statements represent a valuable resource to offer insights into what are the most common declared conditions of data availability and code availability in particle physics. Such an analysis would then give a general idea of reported barriers to reproducibility in the field, by showing the degree to which data and code of recently published articles is reported as publicly accessible. Finally, large scale analysis of the content of these statements could serve an arguably even more important purpose. That is, investigating whether the current guidelines on how to describe availability of data and code are sufficiently adequate to ensure that information vital to reproducing a study is included in the manuscript.

To this end, in the current thesis, a FAIR-aligned data-processing pipeline was developed in order to conduct large-scale analysis of CAS and DAS sections in scientific studies. In particular, the pipeline was designed as a reusable software tool serving two main purposes: a) verify the presence of DAS and CAS sections on large batches of scientific studies; and b) flexibly classify their content according to the access conditions described for data or code.

Chapter 2

Methods

2.1 General Software Architecture

The software developed is a reproducible data-processing pipeline designed for extracting and classifying Data Availability Statement (DAS) and Code Availability Statement (CAS) sections from scientific studies. The pipeline takes PDF files as input and performs the following operations: converts the PDFs into text, extracts and normalizes the targeted sections, and produces a classification output following manually curated schemata.

The pipeline is organized around one data ingestion module and four higher-level modules corresponding to the different operations (i.e. PDF conversion, section extraction, section normalization and section classification). These modules are orchestrated by a main orchestration layer, which exposes a function to run the full pipeline.

In order to ensure full reproducibility and traceability, a system of registries is integrated with the data-processing modules to continuously generate and store metadata about all operations performed (Fig. 1). The registry system is structured around a general output registry, which records metadata common to all pipeline outputs (e.g. identifiers, software versions), and stage-specific registries, which store metadata relevant to particular output types (e.g. extraction stage, classification outputs). Registry files are directly integrated into the logic of the pipeline. The four higher-level modules read the registries to retrieve the paths of files that have to be processed and the DOIs of the articles associated with those files. The DOIs act as persistent identifiers to make each processing output traceable back to a publication. Once the input file is identified, a lower-level processing function is called in order to perform the operation relative to that module. The function performs the operation, generates a new pipeline output and stores meta-

data about the output in a registry entry. Once generated, each output is assigned a unique hexadecimal hash as a persistent identifier. Additionally, a skip-processing logic is implemented at each stage to use these persistent identifiers to avoid re-processing files.

2.2 Data Management and FAIR Compliance

2.2.1 Registries

As mentioned in the previous section, a system of CSV registries is directly integrated in the data-processing pipeline to continuously record metadata about pipeline inputs and outputs. The system counts a total of seven registries: a document registry, a base output registry, a raw files registry, an extraction registry, an embedding registry and two registries for classification outputs.

The document registry records metadata about individual publications. The registry includes both metadata directly harvested from the SCOAP³ repository (e.g. DOIs, URLs, titles, authors, journals, etc.) and metadata produced during some of the processing stages (e.g. CAS/DAS section inclusion, document type, etc.) Document DOIs act as primary keys and unique identifiers for all registered publications. The base output registry is a general-purpose registry that records metadata common to all processing stages (e.g. output hashes, software version, dependencies, etc.). The hexadecimal hashes (i.e. `output_sha`) act as primary keys for all pipeline artifacts produced during processing. The base registry also registers the DOI of the records associated with that specific artifact, which makes each artifact traceable back to a specific publication. The base output registry is what bridges document metadata to stage-specific metadata of artifacts.

The remaining five registries are stage-specific. The raw files registry records metadata about PDFs, XMLs and whole text files of the papers associated with the publications (e.g. file path, file size, etc.), the extraction registry records metadata about extracted sections and normalized sections (e.g. stage, section type, section text, etc.), and the embedding registry records information about embedding outputs (e.g. file path, model used). Finally, the two classification registries store metadata about classification outputs of the two section types, such as classification labels, levels of confidence, what model was used, and other data. All stage-specific registry entries include the artifacts' corresponding identifiers. These are used to connect them back to the base output registry, which in turn links them back to their associated publication in the document registry via DOIs.



Figure 1: General architecture of the data-processing pipeline and registry system.

2.2.2 FAIR Compliance

Findability

The whole data/metadata bundle is freely accessible in an online repository and is assigned with a globally resolvable, persistent identifier. Individual records and artifacts are assigned unique, persistent identifiers. These are not resolvable globally, but can be used to find and identify single artifacts once the local archive system is reproduced. The data bundle is packaged in order to reproduce the project workspace, and file paths in the registries use relative paths in order to preserve their meaning when the project workspace is reinstated. As evidenced by the registry system, all data artifacts (source files, pipeline artifacts, classification outputs) are described with abundant metadata, and each registry entry includes the specific identifier of the described entity (i.e. DOIs and hexadecimal shas). Metadata is indexed in a searchable resource only at the dataset level in the published bundle, while single metadata entries are indexed and findable at the local level.

Accessibility

The dataset bundle is retrievable through a standardized communication protocol (e.g. HTTPS) using a globally resolvable, persistent identifier (i.e. the Zenodo DOI associated with it). The protocol used to retrieve the data is free, open and universally implementable.

Data and metadata are published together in Zenodo under the same data bundle. Data is published under permissive licences and should remain accessible. Even if the access conditions of source files change, the associated metadata will remain available in its corresponding Zenodo page.

Interoperability

A subset of metadata and research artifacts of this study are formally expressed in a lightweight knowledge graph in RO-Crate using JSON-LD as linked data format. The dataset is defined as the RO-Crate root data entity and linked to a number of relevant sub-entities representing documentation, affiliations, software, registry files, analytical outputs and related data releases. The knowledge graph in question is relatively minimal, so the dataset is only expressed formally at a limited level of granularity. For instance, individual metadata entities in the pipeline are defined through controlled labels and local schemas/documentation, but are not expressed through formal ontologies or linked-data formats, and are therefore only partially machine-actionable at the present stage. Relationships among metadata entities are

also informally described in the documentation, but are only partly expressed as qualified references by linking the dataset entity with relevant objects and files.

Reusability

Metadata entries of data artifacts include a rich plurality of relevant attributes recorded in the different registries. Generally speaking, all data and metadata associated with this thesis is published under permissive licences to maximize reusability. Most of the data and metadata are released under CC-BY-4.0. Licence specifications and author information are included in the manifests published alongside the data in order to guarantee proper attribution. Metadata harvested from the SCOAP³ repository is published under CC0 to preserve the previously assigned licence. Data and metadata provenance is tracked through fields included in the registry system and via the software documentation. All artifacts are traceable back to specific publications and software versions, and include the persistent identifier of their dependencies (i.e. the artifact from which they were generated). Specific publications are traceable back through DOIs, and URLs of all downloaded files are included in the registries. Metadata about individual publications additionally include author names. The dataset bundle is released with a data notice specifying information about the author of the project to ensure provenance and attribution. Even though there is not one single metadata standard applied universally, the project uses community-recognized identifiers, open formats, documented labels and commonly used licences.

2.3 Software Modules

2.3.1 Data ingestion

The data ingestion module acts as a standalone function for harvesting publications from the SCOAP³ Repository. The module ingests individual records by making HTTP requests to the SCOAP³ Repository API. The API issues a JSON response containing metadata about individual records, which is then parsed into a Python object. From the Python object, metadata about the record is read and saved in the document registry. In the same processing step, URLs of XML and PDF files associated with the record are used to download them by making additional HTTP requests, while metadata about these files is saved in dedicated registries. The module also incorporates a DOI-based skipping mechanism to avoid duplicating records in the local sys-

tem. Additionally, the function supports filtering the harvest of records by a specific publication period through function arguments.

2.3.2 PDF conversion

This is the first module in the data-processing pipeline. The module converts article PDFs into text files by using functions present in the PDF manipulation library PyMuPDF. The PDF file is opened as a structured document, and the function sequentially iterates through pages, extracts embedded text content and writes it in a new text file. Once the text file is created locally, metadata about the file and the processing operation are saved in a dedicated registry. This module also implements a skipping mechanism based on verifying that the hexadecimal hash associated with the converted PDF does not already exist in a pre-computed list of dependency hashes from previous conversions.

2.3.3 Section extraction

The second module of the data-processing pipeline is dedicated to the extraction of DAS and CAS sections from text files. Candidate files are selected by reading the raw file registry and selecting only files with file type “txt”. Section extraction works by using regular expression patterns to match the start and end positions of sections. Start regular expression patterns correspond to strings that generally mark the section’s title (e.g. “Data Availability Statement”). End regular expression patterns correspond to strings that mark the title of the subsequent section (e.g. “Open Access”), or strings matching typically occurring patterns signaling elements not part of the section (e.g. “TABLE I”, “(a)”, “0\n0.2\n0.4 [...]”). These regular expression patterns are loaded from a separate configuration file. The list of regex patterns was developed through an iterative error analysis process, whereby the updated regex pattern list was continuously tested against extraction outputs. Extraction outputs were regularly inspected to investigate whether the existing regex pattern list would produce correctly extracted sections. Upon finding malformed sections, new regular expression patterns were conceived, added to the configuration files and tested against the text file of the previously malformed section.

The section extraction module also implements a mechanism to identify the document type of the processed file. If no start regex pattern is found, the module uses specific regular expression patterns to check whether the text file might be a different document type than a scientific study (e.g. an erratum). If a regex pattern corresponding to a different type of article is

found, the document type of that record is updated accordingly. If neither a start regex pattern nor regex patterns corresponding to a different document type are found, the module records the document as a “Scientific study” and the specific section being extracted as missing from the document.

Conversely, if a start regular expression is found, the absolute position of the regex match is stored internally and the module starts searching for the earliest occurring end regular expression pattern. Once the end regex is found, the absolute position of the match is recorded. The positions corresponding to the start and end regex patterns are then used for slicing and extracting the section. The extracted section is then written to a local text file and registered as a “pre-cleaning” extraction output. In case a start regex pattern is found without a corresponding end regex pattern, the extraction is marked as “failed” in the registries.¹

Similarly to other modules, the section extraction module also implements a skipping mechanism based on checking the hash of the processed files against a precomputed list of hashes. These hashes correspond to text files that have already been processed under the same version of the software/pipeline. The skipping mechanism can be avoided by configuring a specific function parameter to force re-processing of files. To avoid registry entry duplication when forcing re-processing, the hexadecimal hash of previously processed files is calculated in order to identify and remove their corresponding registry entries.

Furthermore, the section extraction module optionally includes a mechanism for “repairing” sections interrupted by a page break. The mechanism uses a rule-based helper function to detect possible section interruptions. If an interruption is detected, the module makes an API call to a large language model. The model is instructed to identify if an interruption actually occurred, repair the section and return it. The model response is then parsed to extract the text and the section is written to a local text file.

2.3.4 Data normalization

The third module of the data-processing pipeline handles normalization of extracted section texts. The purpose of this module is to normalize text such that sections with equivalent semantic content produce sufficiently similar text outputs. It takes as input pre-cleaning extracted sections and outputs cleaned extracted sections. The module selects candidate files by reading

¹Many DAS and CAS sections include a so-called “Author’s comment”, a bracketed part of the section where authors can give additional information on artifact availability. This part of the section is masked during the regex detection process in order to avoid detecting end regex patterns inside the comment.

paths from the extraction registry and applies the lower-level normalization function to each individual file. The normalization function applies a series of transformations to the text designed to correct artifacts resulting from PDF conversion and section extraction (e.g. broken hyperlinks, glued words, missing final periods) and standardize textual content (e.g. removing leading whitespace, replacing ligatures). Once these transformations are applied, the resulting text is written to a local file. Metadata about the newly created file and the processing operation is recorded in dedicated registries. Similarly to other modules, a skipping mechanism is implemented using dependency hashes scoped to software/pipeline versions.

2.3.5 Section classification

The final data-processing module classifies normalized sections according to manually curated schemata. Classification works by issuing API calls to a large language model. The model is provided with the section to be classified, task instructions, a list of admissible labels and schemata to guide classification.² DAS and CAS sections are classified following different classification schemata. DAS sections are classified along a single dimension, while CAS sections are classified along two orthogonal dimensions. In order to limit costs and processing time, task instructions and classification schemata are cached, while each section text is passed as the variable user message. The prompt instructs the model to apply the classification schemata to the section at hand and return a structured classification output. Additional instructions require the model to exclusively output the classification label(s) and confidence level(s) in JSON format. The model response is then parsed to extract the classification output. The output and metadata related to this processing step are then recorded and stored in dedicated registries. The classification module also implements a skipping mechanism scoped to software/pipeline version for sections that were already classified. As in other modules, the module can be configured to allow force re-processing of section files, implementing the same deduplication logic as other modules. Additionally, the module allows users to select which model should be used for classification through function parameters. In the implementation used for classifying data discussed in this thesis, the module used `claude-haiku-4-5-20251001` as the default model for classification.

²Initially, the classification mechanism was supposed to be prototype-based cosine similarity with contextual embeddings. Unfortunately, no off-the-shelf embedding model turned out to be sufficiently functional for the task at hand.

2.4 Schemata Development

Three schemata were developed in the course of this thesis project, one for DAS sections and two for CAS. DAS sections were classified along the dimension “Data Availability” and only required a single schema. CAS sections were classified along two orthogonal dimensions: “Manuscript Code Availability” and “External Tool Availability” using two separate schemata.

Schema development resulted from a preliminary exploration of data through common machine learning techniques. This exploratory phase helped to identify latent candidate categories and grounded schema development directly in observed data. Two techniques in particular were used. First, large portions of extracted sections were represented via contextual embeddings. Contextual embeddings can be defined as high-dimensional vector representations of text, where the numerical representation of individual words or tokens depends on their surrounding context [25, 26]. In sentence-level embedding models, these contextual representations are aggregated into a single vector representing a larger text unit. These vectors are intended to reflect semantic features of a sentence, such that similar texts produce similar representations. Once section texts were embedded, a hierarchical density-based clustering algorithm (HDBSCAN) was employed in order to explore the structure of the embedding space and observe emergent categories. This algorithm identifies high-density regions in embedding space as separate clusters, while labeling observations not belonging to stable dense regions as noise [27]. Observed clusters were then used to inform the development of some of the schema categories by inspecting individual cluster members and interpreting their shared meaning. For example, observations expressing the idea of “no data associated with the study” would group together in the same cluster, while being separated from other clusters representing ideas such as “data available upon request” or “data publicly available”. Clusters themselves were not treated as final categories, but rather as exploratory groupings to guide category conceptualization. A number of categories were developed by observing sections labeled as noise and by conceptually deriving missing categories to represent all possible access conditions for data and code.

Classification schemata were defined in a YAML file and were structured as follows:

Category name Name of the category.

Category definition Brief definition of the category.

Category exemplars Examples of category members.

Category hard positives Pairs of linguistically dissimilar statements belonging to the category at hand.

Category hard negatives Pairs of linguistically similar statements, one belonging to the category at hand, the other to a different category.

Explicit decision rules - inclusion Criteria for including sections in the category.

Explicit decision rules - exclusion Criteria for excluding sections from the category.

Ambiguous cases notes Notes explaining how to handle complex or ambiguous cases.

2.4.1 DAS Classification Schema

The classification schema for DAS sections categorizes sections along a single dimension: reported availability of data elements. In this context, elements of data can correspond to many different data artifacts. Artifacts included in this list are datasets, measurements, tables, figures, files and generally any research output supporting a study's findings. The model is instructed to classify sections according to the access conditions reported for detectable data elements. Data exploration coupled with conceptual analysis led to the development of the following seven categories for access conditions of data:

no_data: DAS sections stating that no data was used, analyzed or produced during the study.

open_access: DAS sections stating that all elements of research data described can be accessed openly by anyone.

partial_access: DAS sections stating that only part of the elements of research data are openly available to anyone.

restricted_access: DAS sections stating that data is available from a source upon request.

no_access: DAS sections stating that data cannot be made available under any circumstances.

incorrect: DAS sections stating information not pertaining to data availability.

unclear: DAS sections stating ambiguous or unclear information about data availability.

Both in the schema and in the instruction prompt, the model was explicitly instructed to not treat code/software as research data. Therefore, DAS sections exclusively reporting code or software availability were assigned to the “incorrect” category. In case both information about research data and code was reported, the task instructions required the model to prioritize information about data for classification purposes. In the DAS classification schema, hard positives and hard negatives were used to represent difficult distinctions, such as statements where some data artifacts were reported as openly available while some were reported as only available upon request, or statements where code was reported as openly available but the data themselves were not.

2.4.2 CAS Classification Schemata

The classification schema for CAS sections classifies sections along two dimensions: Manuscript Code Availability (MCA) and External Tool Availability (ETA). The need for two distinct dimensions of classification arose from an important observation made in the exploratory phase of category development. By inspecting cases clustered as noise, it became apparent that authors did not exclusively report information about code or software they developed, but would also provide information about external software tools used in the course of the study. This resulted in apparently contradicting statements where authors would mention that “no code [was] associated with [their] study” but would also report that Python packages had been used for calculations. Therefore, a dichotomous taxonomy was conceived in order to adequately categorize statements including access conditions of different types of software artifacts. The taxonomy distinguished Author-Produced Software Artifacts (APSA) from External Software Tools (EST), classifying the former along the MCA dimension and the latter along the ETA dimension. In the instruction prompt, the model was instructed to identify all distinct elements of software, distinguish between APSA and EST, and only then classify according to access conditions. Also, the model was given explicit criteria to distinguish the two types of artifacts. In particular, APSAs were described as software artifacts shareable as part of the manuscript that were produced, adapted, curated or packaged by the authors for study-specific purposes. APSAs could include code, scripts, notebooks, macros, configuration files, custom packages, Docker images, and virtual environments. External Software Tools were instead described as pre-existing, third-party,

general-purpose tools not specifically developed by the authors as part of the study. These could include programming languages, libraries, simulation frameworks, event generators, and proprietary computational software. In order to properly assess the access conditions of ESTs, the general instruction prompt included a list of commonly used ESTs and their known access conditions. Although reporting EST usage could be interpreted as an indication that APSAs were created in the course of the study, the model was explicitly instructed to not deduce APSA existence from mentioned ESTs alone. For instance, despite mentioning usage of an external tool, the following statement was featured among exemplars of sections where no APSA is mentioned:

“This manuscript has no associated code/software. [author’s comment: this manuscript relies on the standard implementation of the FeynCalc package in Wolfram Mathematica for symbolic computations.]”

The Manuscript Code Availability dimension was organized around seven categories mirroring the access conditions described in the DAS classification schema:

no_code: CAS sections stating that no Author-Produced Software Artifacts (APSA) are associated with the manuscript/study.

open_access: CAS sections stating that all explicitly-mentioned Author-Produced Software Artifacts associated with the study can be accessed openly by anyone.

partial_access: CAS sections stating that only part of the Author-Produced Software Artifacts mentioned are openly available to anyone.

restricted_access: CAS sections stating that the Author-Produced Software Artifacts mentioned are available upon request.

no_access: CAS sections stating that Author-Produced Software Artifacts are not accessible under any access condition explicitly specified in the statement.

incorrect: CAS sections that do not provide information about availability, use or existence of identifiable elements of software, whether Author-Produced Software Artifacts or External Software Tools.

unclear: CAS sections stating ambiguous or unclear information about the availability of Author-Produced Software Artifacts.

External Tool Availability was instead organized around five, conceptually-derived categories:

no_tool: CAS sections not mentioning any External Software Tool.

open_tool: CAS sections only mentioning the use of open, free-to-use External Software Tools.

mixed_tool: CAS sections mentioning the use of a combination of publicly available tools, proprietary tools or tools whose access conditions are unknown.

proprietary_tool: CAS sections only mentioning the use of third-party, proprietary External Software Tools.

unknown_status_tool: CAS sections mentioning the use of External Software Tools whose access conditions are unknown.

On top of rules for correctly distinguishing between APSA and EST, the model was also provided with rules to correctly interpret claims about access conditions. In particular, special attention was given to common cases where the structure of the statement consisted of one generic boilerplate statement, followed by an author’s comment giving more detailed information about code availability. Specifically, rules were given for situations where: a) the access conditions specified in the initial boilerplate statement could be assigned to a subsequently mentioned APSA; b) a positive availability claim made in the initial boilerplate statement could be treated as evidence for the existence of an APSA even if not mentioned (e.g. “code/software will be made available on reasonable request. [authors’ comment: no additional remarks.]”); and c) a negative availability claim should not have been treated as decisive evidence that no APSA existed if later contradicted by the information in the author’s comment (e.g. “The manuscript has no associated code/software. [author’s comment: custom Mathematica code was used for calculations and can be provided upon reasonable request.]”). Examples of the corresponding schema rules are reported in [Appendix A](#).

In the two CAS classification schemata, exemplars, hard negatives and hard positives were not only used for guiding correct categorization of access conditions, but had the additional function of guiding correct APSA-EST differentiation.

2.5 Data Source and Sampling

The total corpus of analysed data consisted of 17,185 documents published on the SCOAP³ Repository ($n = 8,511$ for 2024, $n = 8,674$ for 2025). These amount to about 98.4% of total active records for 2024 ($n = 8,531$, 99.76%) and 2025 ($n = 8,929$, 97.14%) present in the repository at the time of writing, so the corpus analyzed should not be interpreted as fully exhaustive of the registry population. The discrepancy was the result of a mix of several factors, including: authentication denials while scraping records, missing downloads of some PDF files, and PDF-to-text conversion failures. The analyzed studies came from 11 journals in total: the Journal of High Energy Physics (JHEP, $n = 5,299$, 30.8%), Physical Review D (Phys. Rev. D, $n = 5,139$, 29.9%), European Physical Journal C (EPJC, $n = 2,803$, 16.3%), Physics Letters B (Phys. Let. B, $n = 1,732$, 10.08%), Nuclear Physics B (Nuc. Phys. B, $n = 865$, 5.0%), Physical Review Letters (Phys. Rev. Let., $n = 715$, 4.1%), Chinese Physics C (Chin. Phys. C, $n = 207$, 1.2%), Physical Review C ($n = 200$, 1.2%), Progress of Theoretical and Experimental Physics (PTEP, $n = 186$, 1.0%), Acta Physica Polonica B (APPB, $n = 27$, 0.16%), Advances in High Energy Physics (AHEP, $n = 12$, 0.07%).

Publications were harvested through the data ingestion module, which allows filtering by publication year. In order to be included in the analysis sample, records had to satisfy the following criteria: being a publication in the SCOAP³ Repository, being published in 2024 or 2025, being an article describing a scientific study, having an associated retrievable PDF file and having reached the extraction stage of the pipeline. The main units of analysis were publication records of scientific studies, each identified by a DOI. Out of all processed records, 99.4% ($n = 17,076$) were identified as scientific studies eligible for analysis, while the remainder was a mix of errata, one editorial expression of concern and one retraction note.

2.6 Statistical Validation

Validation of the classification output was done by using manually annotated samples. Two subsets of 500 DAS sections and 500 CAS sections were randomly drawn and manually annotated according to the classification schemata. Manually labeled samples were then used as gold sets to assess the quality of the model's classification output, by comparing the manual labels with the model's labels through measures of inter-rater agreement. In particular, Cohen's kappa was used to estimate overall agreement between manual classifications and model classifications [28]. On top of calculating Cohen's

kappa, disagreement tables and confusion matrices were also computed to inspect differences in the two classification outputs.

Chapter 3

Results

3.1 Data Availability Statements

3.1.1 DAS Presence

Out of the total sample of processed studies, around 63.5% ($n = 10,844$) included a DAS section. That puts it at about two thirds of the total sample. Excluding journals with relatively small samples of analyzed studies ($n < 186$), the three top journals with the highest proportions of DAS sections present were European Physical Journal C (99.6%, n present = 2730), Physics Letters B (98.4%, n present = 1,698) and Nuclear Physics B (97.56%, n present = 840). Year-specific results showed a marked change in the inclusion of DAS sections in 2025 scientific studies. Specifically, DAS sections were present in about 35.1% of studies in 2024 ($n = 2,964$), while this number increased considerably in 2025 at about 91.3% ($n = 7,880$) (Fig. 2). That amounted to an increase in DAS inclusion of 56.2 percentage points from one year to the next. Seven journals saw an increase in DAS section inclusion, two journals a slight decrease (Physics Letters B, -0.45 pp, Chinese Physics C, -0.65 pp), Acta Physica Polonica remained stable, while Advances in High Energy Physics had no 2025 studies among analyzed data. The Journal of High Energy Physics (+89.1 pp), Physical Review D (+81.6 pp) and Physical Review C (+ 67.7 pp) were the three top journals for percentage point increase in DAS inclusion in 2025 scientific studies.

3.1.2 DAS Classification

A total of 10,843 unique DAS sections were classified. One scientific study did not receive classification due to the DAS section not having any content (DOI: 10.1093/ptep/ptaf057). Out of all classified DAS sections, 66.7% declared

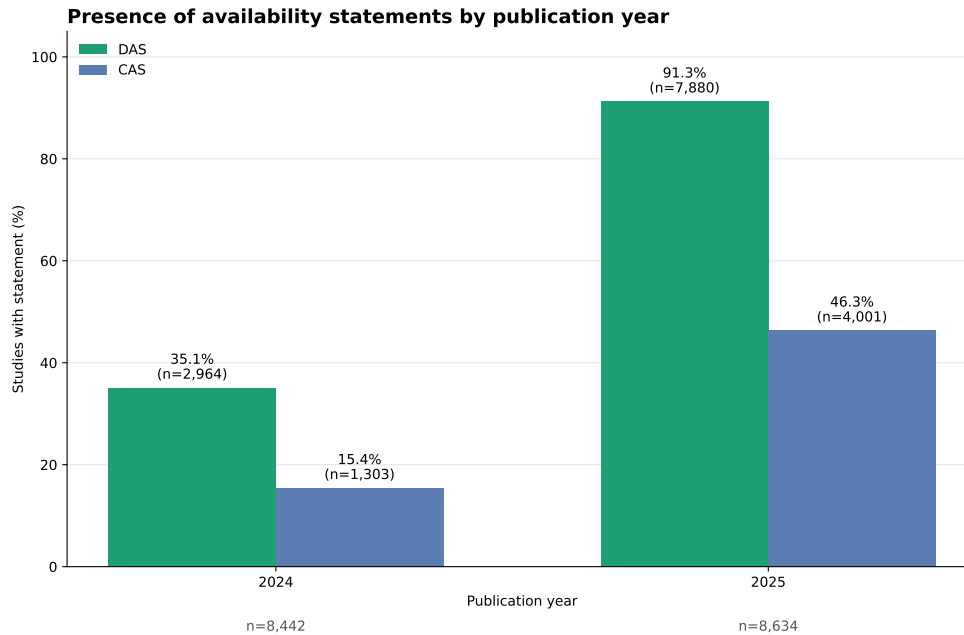


Figure 2: Presence of Data Availability Statements and Code Availability Statements by publication year.

that no data was associated with the study ($n = 7,237$), 2.4% of sections were judged as unclear following the instructions (unclear, $n = 255$) and 0.3% reported incorrect content ($n = 32$) (Fig. 3). Notably, European Physical Journal C accounted for 89.8% of all unclear DAS classifications ($n = 229$ of 255).

Among studies with associated data that reported access conditions clearly ($n = 3,319$), 59.5% reported data as only being available upon request (restricted_access, $n = 1,976$), 37.0% reported the data as openly available (open_access, $n = 1,228$), 2.4% reported mixed access conditions (partial_access, $n = 80$), and only 1.1% reported data as not being accessible (no_access, $n = 35$). Year differences in category shares for this specific subsample showed a visible 2025 increase in studies reporting openly available data (open_access, +15.8 pp) accompanied by decreases in studies reporting data as available upon request (restricted_access, -11.6 pp), studies reporting data as not available (no_access, -2.82 pp) and in studies reporting only part of the data as openly available (partial_access, -1.38 pp) (Fig. 4).

When considering only clearly reported access conditions of studies with associated data, journal-specific results showed that in 6 out of the 10 journals containing DAS sections, the most prevalent category was restricted_access. In 6 of these, more than half of studies were classified as restricted, with

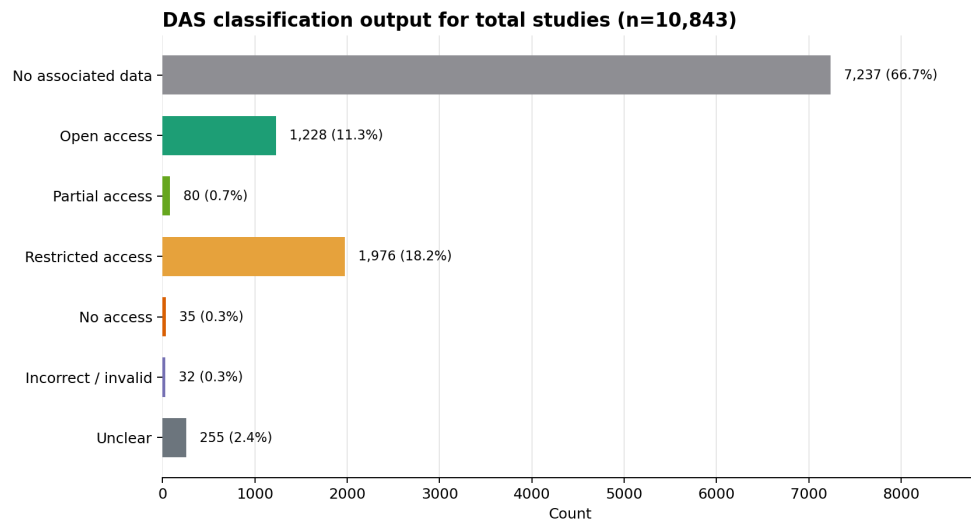


Figure 3: Classification distribution of Data Availability Statements.

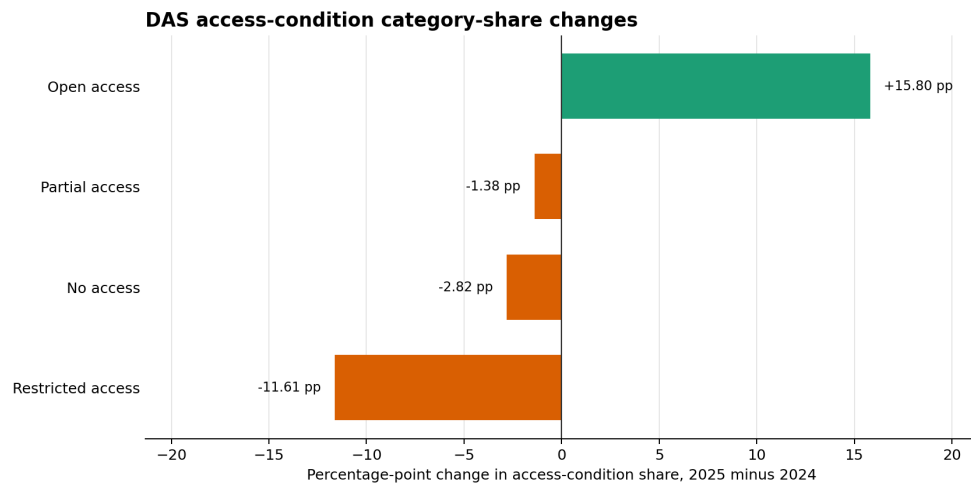


Figure 4: Percentage-point change in DAS access-condition categories between 2024 and 2025, considering only studies with associated data and clear access conditions.

three journals going over 70% (Nuc. Phys. B, 79%; Phys. Let B, 78.8%; PTEP, 77.8%). In the remaining four journals, the most prevalent category was `open_access` (Ch. Phys. C, 100%; JHEP, 96.7%; AHEP, 66.7%; Phys. Rev. Let., 58.1%).¹

3.2 Code Availability Statements

3.2.1 CAS Presence

Out of the total sample of processed studies, around 31.1% ($n = 5,304$) included a CAS section. Therefore, only about one in three studies included a Code Availability Statement. Notably, only two journals accounted for almost all CAS sections detected (JHEP, $n = 2866$, 54%; EPJC, $n = 2432$, 45.8%), while two other journals had a very limited number of sections detected (Phys. Rev. D, $n = 3$; Physics Letters B, $n = 3$). The remaining 7 journals had no detected CAS sections (Fig. 5).

CAS section inclusion also visibly increased in 2025 ($n = 4,001$, 46.3%) compared to 2024 ($n = 1,303$, 15.4%) for a total percentage point increase of 30.9 (Fig. 2). The largest journal-level increase in CAS inclusion was from the Journal of High Energy Physics (+89.1 pp), while European Physical Journal C also showed an increase but to a lesser extent (+22.8 pp).

3.2.2 CAS Classification

A total of 5,303 unique CAS sections were classified. One scientific study did not receive classification due to an unresolved extraction error (DOI: 10.1016/j.physletb.2024.138800).

Manuscript Code Availability

Out of classified CAS sections, 79.3% reported no Author-Produced Software Artifacts associated with the study (`no_code`, $n = 4,207$), 1.7% had sections judged as unclear under the schema rules ($n = 92$) and only in 0.7% of sections the content was judged as incorrect ($n = 35$) (Fig. 6). Among studies with author-produced code that clearly reported access conditions ($n = 969$), 60.2% reported code as available upon request (`restricted_access`, $n = 583$), 35.1% reported code as openly available (`open_access`, $n = 340$), 2.9% reported code as not accessible (`no_access`, $n = 28$) and 1.9% as only partially

¹Although it should be noted that the number of studies with associated data and a clear section were only 4 for Chinese Physics C and 9 for Advances in High Energy Physics.

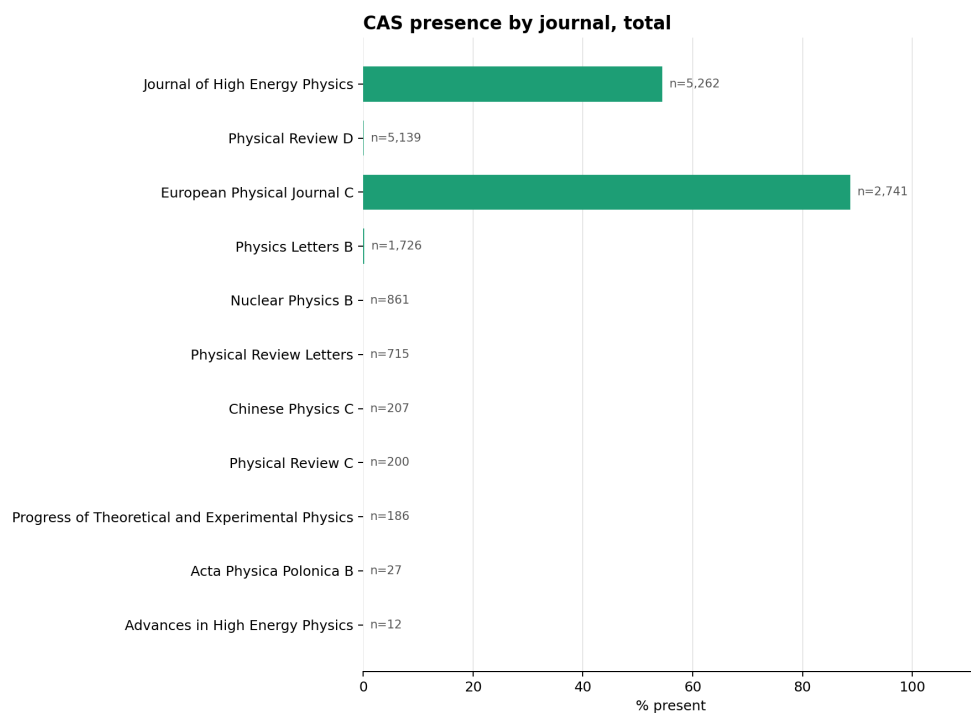


Figure 5: Presence of Code Availability Statements by journal in the full analyzed sample.

available (partial_access, $n = 18$). For the MCA dimension, year differences in category shares of sections clearly reporting access conditions of associated code showed an increase in studies with openly available author-produced code (open_access, +16.2 pp), little to no changes in studies reporting partially accessible code (partial_access, +0.18 pp) and studies reporting code as not accessible (no_access, -0.01 pp), and a clear decrease in the share of studies that reported code as available upon request (restricted_access, -16.4 pp) (Fig. 7).

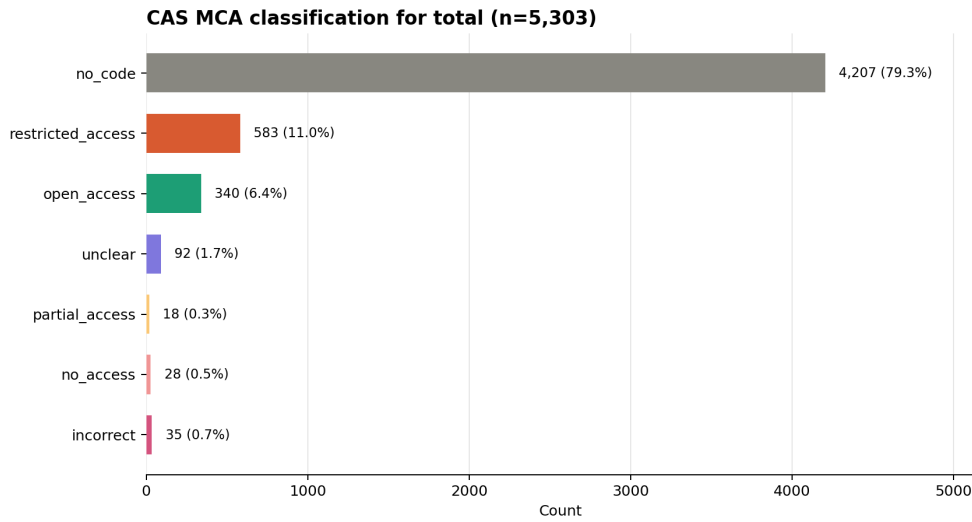


Figure 6: Classification distribution of Code Availability Statements along the Manuscript Code Availability dimension.

For the two journals that made most of observations for CAS sections, the category no_code was the most prevalent (JHEP, $n = 2,600$, 90.7%; EPJC, $n = 1,607$, 66.1%). When only sections of studies with associated code clearly reporting access conditions were included, open_access was the most prevalent category for the Journal of High Energy Physics ($n = 227$, 96.2%), while restricted_access was the most prevalent for European Physical Journal C ($n = 578$, 79.4%).

External Tool Availability

Out of classified CAS sections, only 3.9% reported usage of External Software Tools ($n = 206$). Among studies whose section reported use of external tools, 2 out of 3 sections reported using an open tool (open_tool, $n = 136$, 66.0%), 18.9% reported using a proprietary tool (proprietary_tool, $n = 39$), 14.1% reported using a mix of open tools and proprietary tools (mixed_tool, $n = 29$).

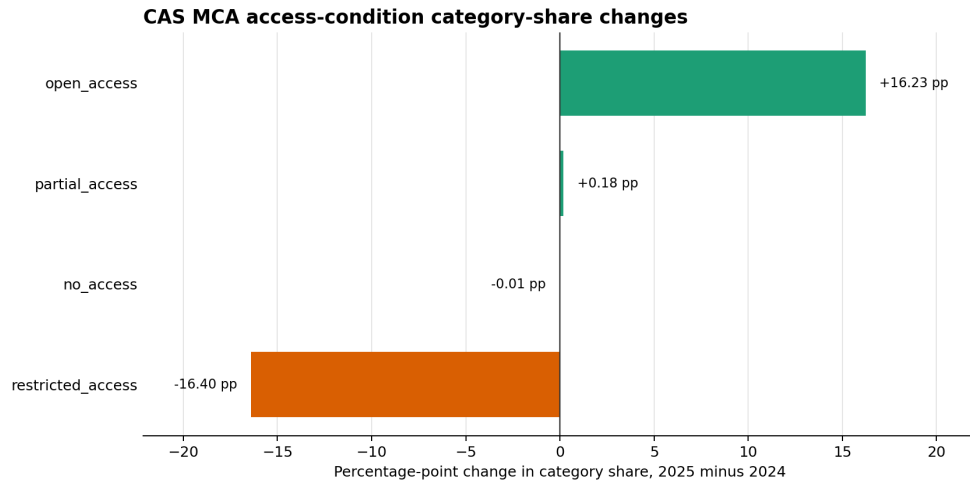


Figure 7: Percentage-point change in CAS-MCA access-condition categories between 2024 and 2025, considering only studies with author-produced code and clear access conditions.

and in 1.0% of cases, the access condition status of the tool was reported as unknown (`unknown_status_tool`, $n = 2$). Upon inspecting year differences in category shares, a small increase could be noticed in sections reporting no tool usage (`no_tool`, +2.69 pp).

MCA-ETA Combinations

Out of all classified CAS sections ($n = 5,303$), the most prevalent combination of the MCA and ETA categories was the reported absence of author-produced code and external tool usage (`no_code/no_tool`, $n = 4,094$, 77.2%) and generally, the first four were combinations involving no tool usage (`restricted_access/no_tool`, $n = 536$, 10.1%; `open_access/no_tool`, $n = 325$, 6.1%, `unclear/no_tool`, $n = 84$, 1.6%). Among category combinations excluding categories `no_code` and `no_tool` ($n = 93$), the three most prevalent combinations were 1) author-produced code available upon request and proprietary tool used (`restricted_access/proprietary_tool`, $n = 23$, 24.7%); 2) author-produced code available upon request and an open tool used (`restricted_access/open_tool`, $n = 21$, 22.6%) and 3) author-produced code not accessible and an open tool used (`no_access/open_tool`, $n = 18$, 19.4%). In 7.5% of cases, author-produced code was openly available and an open tool had been used (`open_access/open_tool`, $n = 7$, 7.5%).

Notably, whenever usage of an external tool was mentioned ($n = 206$), more than half of sections reported no author-produced software associated

with the study ($n = 113$, 54.9%). The three other most prevalent categories in this case were `restricted_access` ($n = 47$, 22.8%), `no_access` ($n = 19$, 9.2%) and `open_access` ($n = 15$, 7.3%).

3.3 Statistical Validation

Statistical validation measures showed very high levels of agreement when comparing the manually-labeled sample with the model-labeled sample from the latest available software version. In total, there were 30 disagreements out of 1,500 unique classification decisions along the DAS dimension and the two CAS dimensions. Specifically, DAS classifications had 6 disagreements ($n = 500$, 98.8% accuracy), CAS-MCA had 21 disagreements ($n = 500$, 95.8% accuracy) and CAS-ETA had 3 ($n = 500$, 99.4% accuracy). Across all three classifications, Cohen's kappa values observed were very high (DAS: $k = 0.9759$, 95% CI [0.9625, 0.9894]; CAS-MCA: $k = 0.8701$, 95% CI [0.8406, 0.8996], CAS-ETA: $k = 0.9243$, 95% CI [0.9011, 0.9475]).

Chapter 4

Discussion

4.1 Analytical Results and Study Implications

Generally speaking, the analytical results of this project testify to an increase in research artifact availability statements in 2025, especially for data. A large majority of 2025 studies did include a DAS section, with considerable increase compared to the previous year. Journal-specific results showed an increase among most journals, while one of the two journals where a slight decrease was detected had already very high inclusion of DAS sections in 2024 (Phys. Let. B, 2024 DAS presence = 98.4%). These results could be interpreted positively in the context of SCOAP³'s newly introduced policies on availability statements, and widespread increase in the inclusion of DAS sections could be a signal of publishers adapting to these policies.

Although only 1 out of 3 studies included a CAS section, a general increase of inclusion was also observed for CAS in 2025 publications. Differently from DAS, this increase was not observed across many journals though. Almost all studies with detected CAS sections came from only two journals (JHEP and EPJC), while a majority of journals seem to not have included CAS sections either in 2024 or in 2025 publications, signaling that a possible policy adoption did not occur globally. Even though the absence of CAS cannot be decisively interpreted as absence of reported code, in the context of the data analyzed, reporting of artifact availability seemed to be worse for code than for data.

Classification of DAS sections revealed that about two in three studies reported no data associated with the study. This may partly reflect the idea of particle physics being a field where many studies are theoretical in nature and do not involve experimental work. Nonetheless, a large proportion of studies reported as not having associated data might raise questions on what

authors interpret as data associated with their study. Arguably, the notion of “data” in high-energy physics is very broad and encompasses a large variety of research artifacts. In their review of data preservation in high-energy physics, Basaglia et al. [29, Sect. 2] describe data as “the multiverse of all specific inputs, outputs and digital tools used by a group of researchers in order to obtain a novel result” and report a long, comprehensive list of artifacts that could be considered data in high-energy physics. Therefore, it is possible that at times, authors might interpret the idea of data in a more narrow sense (e.g. only as raw/process data on empirical observations) and might not report research artifacts that could be considered as data in availability statements.

A closer inspection of DAS sections classified as “unclear” showed cases of logical inconsistencies between the initial boilerplate statement and the author’s comment. Three types of unclear sections turned out to be particularly prevalent. The first type are sections where the initial boilerplate explicitly mentions “no data associated with the manuscript” but the author’s comment instead reports associated data with corresponding access conditions. This might also suggest a possible interpretation error, whereby the notion of “associated with the manuscript” might be interpreted as “deposited with the manuscript record” rather than “produced during the study”. The second type involves cases where the initial statement describes data as available upon request, while the author’s comment reports no data associated with the study. These first two types were particularly visible in EPJC publications. The third type refers to cases of model errors. Arguably, a small subset of sections classified as unclear could have belonged to other categories.

When looking at sections that clearly reported access conditions for studies with associated data, the most common access condition was restricted access. Nonetheless, there was a visible improvement in the availability of data in 2025, as testified by category share changes across the two years. Compared to 2024, in 2025 there was a higher proportion of reported open data, accompanied by a decrease in data available upon request and data not accessible. This improvement in data availability could also be tentatively interpreted as a possible adaptation to the criteria set by SCOAP³’s new Open Science Elements policy [24], in particular those in Dataset Linkage.

CAS classification results should be interpreted with caution considering they almost exclusively refer to two journals. In almost 4 studies out of 5, CAS sections reported the study had no associated code. Similarly to data, this could also raise questions on how the notion of code is interpreted by the author, although the bounded notion of to-be-reported code as author-produced, original, study-specific code (i.e. Author-Produced Software Artifact in the classification schema) could be consistent with a small

proportion of studies reporting associated code. Nonetheless, journal-specific biases might be particularly relevant in CAS classification results, and large proportions of studies reporting no associated code might rather speak to these journals mostly including studies that did not require using specific author-produced software. Unclear classifications for CAS sections showed the same types of inconsistencies visible in DAS classifications. Also for CAS, the section would report either data not associated or data available upon request in the initial statement, only to then contradict it with new information in the author's comment.

When only looking at sections that clearly reported access conditions for associated code, a similar situation to the one for data availability could be observed. Despite 60% of sections reporting code as only available upon request in the total sample, a visible improvement in 2025 could be seen also for code availability. Specifically, the proportion of studies reporting openly available code increased, while that of studies reporting code as available upon request decreased.

As for CAS classification along the External Software Tool dimension, only a small proportion of sections reported using External Software Tools. Rather than reflecting EST usage, it is possible that such a small proportion of sections reporting the use of ESTs can be explained by a majority of authors not considering ESTs as the normative target of CAS sections. The observed proportion can be reasonably thought of as being an underestimation of actual EST usage. Since the definition of EST is very broad and includes even basic programming languages and computational environments, it is sensible to expect the proportion of actual EST usage to be at least as high as the proportion of studies reporting associated author-produced code. Nonetheless, among those that were reported, around 2 out of 3 tools were identified as free, openly accessible tools. Notably, a majority of studies did not report author-produced code associated with the study when a tool was mentioned. This observation could be considered counterintuitive since one might expect a minimal code base whenever an EST is used. Still, a majority of studies using tools but not reporting associated code might be an indication of a narrow interpretation by authors of to-be-reported code as only author-produced, original and study-specific code.

In light of the considerations above, one takeaway of this study is that there seems to be some degree of confusion around what should be the subject of DAS and CAS sections, possibly boiling down to a lack of clearly established definitions of data and code when reporting their access conditions in these sections. Whereas for DAS this claim is quite speculative as it only rests from observing large shares of studies reporting no associated data, for CAS this consideration is grounded in the observed data. In partic-

ular, authors seem to report generation or usage of different types of software artifacts, and in a majority of cases only for one or another artifact. This is what ultimately set the requirement for a minimal, data-driven taxonomy of software artifacts in order to achieve proper classification of CAS sections. Generally speaking, this confusion can highlight the need for better-defined guidelines on what exactly should be the content of these sections, possibly grounded in academic literature on the proper definition of research artifacts in high-energy physics.

4.2 Limitations

4.2.1 Software Architecture

The reusable software pipeline developed arguably has a number of limitations in its current state. Some conversions of PDF files into text failed and should be further investigated. Furthermore, the extraction module's regex-based design does not guarantee deterministically that all sections will be extracted if present. Naming conventions for sections might change over time and new sections might be included in scientific papers, which underscores the necessity of regularly monitoring the appropriateness of the software's configuration files against current standards.

An additional limitation of the extraction module was related to document type detection. Specifically, the document types detected by the module cannot be considered exhaustive of all possible document types published on the repository. This became clear upon finding a number of documents that turned out to be lecture notes in the final stages of the thesis. The cleaning module might also benefit from some refactoring. Specifically, the module has the unwanted side effect of incorrectly splitting some proper names, often those of digital objects. This is due to the word-splitting function not recognizing these as standalone words and incorrectly splitting them.

As for classification, a limitation of the current version of the software is that the model selected for classification is a cost-efficient, lightweight model. Despite passing formal validation checks, this model can have difficulties in properly implementing several different interpretation rules, and arguably still made some classification mistakes overall.

Although not impacting functionality, one further limitation of the software concerns the path construction for stored and generated artifacts. Currently, file storage is organized around two types of versioning: software version and pipeline version. The latter was initially developed with the idea of making substantial modifications to the general pipeline architecture by

adding or removing entire modules. The development of a complete, functional version of the data-processing pipeline did not require making module-level changes deviating from the initially planned pipeline structure. Hence, all files are stored under the same version of the pipeline, and indexing by pipeline version is not meaningful in terms of file system organization.

4.2.2 Data Sampling and FAIR Limitations

Results from this study should only be considered as representative of high-energy physics publications present in the SCOAP³ repository, not of all existing publications. Furthermore, as mentioned previously, the corpus of analyzed data cannot be considered fully exhaustive of the entire population of records published in the SCOAP³ repository from 2024 and 2025.

It is worth noting that this study only analyzed reported availability of artifacts in DAS and CAS sections, not actual accessibility or reusability. Additionally, the current study only analyzed publications from 2024 and 2025. Therefore, analytical outputs should not speak to long-term trends as for the constructs analyzed.

In terms of FAIR data management, one limitation to accessibility is the fact that presently, persistent identifiers assigned to research artifacts (i.e. hexadecimal hashes) are only resolvable in the context of the local archive structure rather than being globally and independently accessible. A number of limitations also concern the interoperability of the data. The dataset is only minimally expressed through a formal knowledge graph, while individual metadata entities are described in the documentation and only use project-specific controlled labels. Even though richly documented, metadata is therefore only partially machine-actionable and interoperable at this stage. Finally, an important limitation to reusability of the software is that the classification module is currently based on making API calls to an LLM provided by a subscription-based service.

4.3 Future Developments and Considerations

4.3.1 Software Refactoring

A number of changes will be implemented to the code base to optimize the pipeline and turn it into a more reusable tool producing machine-actionable data. In terms of general architecture, an interface will be developed for running the pipeline from the command line. Some changes will then be implemented at module level. For PDF failures, a solution that will be

tested will be a fallback logic in the lower-level processing function to use alternative parsers when the primary parser hangs, fails, or returns unusable output. The extraction module will be updated in order to include lecture notes among detectable document types. In order to resolve the problem of some proper names of digital objects being split, a masking mechanism will be implemented in the cleaning module using a pre-defined list of common digital objects and software names. In terms of interoperability and dataset linkage, future versions will also describe individual metadata entities and relationships through linked data formats in order to achieve interoperability at finer levels of granularity. Finally, an important refactoring that will be implemented will be rewiring the classification module to an openly accessible model, either locally installable or externally hosted and testing whether performance is comparable with proprietary models. That could avoid the need for an API key from a proprietary service to run classifications and significantly improve the accessibility of the tool.

4.3.2 Availability Statements and Schemata

As mentioned earlier, variability in reported software artifacts in the data required the development of a more complex taxonomy to accurately classify the content of CAS sections. This suggests the possible need to develop structured, community-level reporting guidelines for availability statements. Development of these guidelines should be guided by a comprehensive conceptual analysis of the notion of reproducibility in high-energy physics studies. Such an analysis should be grounded in existing metascientific literature, but could also benefit from sociological, qualitative investigations using expert interviews and focus groups. The aim of the analysis should be that of systematically identifying artifacts necessary to reproduce studies across different research areas. This could then help define new guidelines for structured reporting in availability statements. Development of these guidelines could fit well with the intrinsic flexibility of the classification mechanism of the tool, since schemata could be progressively adapted to handle different artifact taxonomies and requested outputs.

Chapter 5

Conclusion

This thesis project involved the development of a FAIR-aligned data-processing pipeline for analyzing DAS and CAS sections in scientific publications. The results of the analysis show a general increase in the inclusion of these sections in high-energy physics publications in 2025. Furthermore, the analysis revealed a general move toward open sharing of research artifacts in 2025, since both data and code reported as openly available increased substantially compared to 2024. The analysis also showed the existence of possible inconsistencies around what authors believe should be the subject of these sections. Such inconsistencies should not be passively assessed as simple interpretation mistakes on the part of authors. Instead, they should actively motivate the definition of explicit, comprehensive reporting guidelines. In an operational context tied to reproducibility, DAS and CAS sections could serve the crucial function of systematically listing all artifacts needed to reproduce a scientific study and their corresponding access conditions.

Data Availability Statement

The data and metadata associated with this thesis are openly available on Zenodo at <https://zenodo.org/records/20135209>. The data bundle includes:

- text files of the publications analyzed;
- pipeline artifacts produced during processing, including extracted sections and embeddings;
- metadata registries;
- manifests including attribution and provenance information;

- classification schemata;
- licensing information;
- descriptive statistics and graphs of the analytical outputs of this study;
- manual annotations used for validation measures.

File paths in the registries are relative, and files were packaged so as to reproduce the original workspace once reinstated. The PDFs of the publications analyzed were published separately on Zenodo at <https://zenodo.org/records/20137704>. The RO-Crate created for this project is also available on Zenodo at <https://zenodo.org/records/20202026>.

Code Availability Statement

The source code of the data-processing pipeline developed for this thesis is openly available at https://github.com/Appianibus/MDMC_Internship_Project_Zuppa. The repository includes:

- source code for running the pipeline under `/src`;
- configuration files implemented in the various modules;
- notebooks used for descriptive statistics and data visualization;
- documentation and licensing information.

Bibliography

- [1] C. G. Begley and L. M. Ellis, “Raise standards for preclinical cancer research,” *Nature* **483** (2012) 531–533.
- [2] A. C. Chang and P. Li, “Is economics research replicable? sixty published papers from thirteen journals say “usually not”,” Finance and Economics Discussion Series 2015-083, Board of Governors of the Federal Reserve System, 2015.
- [3] Open Science Collaboration, “Estimating the reproducibility of psychological science,” *Science* **349** no. 6251, (2015) aac4716.
- [4] J. H. Stagge, D. E. Rosenberg, A. M. Abdallah, *et al.*, “Assessing data availability and research reproducibility in hydrology and water resources,” *Scientific Data* **6** (2019) 190030.
- [5] J. P. A. Ioannidis, “Why most published research findings are false,” *PLOS Medicine* **2** no. 8, (2005) e124.
- [6] K. Popper, *The Logic of Scientific Discovery*. Basic Books / Hutchinson, 1959. <https://www.routledge.com/9780415278430>.
- [7] S. Schmidt, “Shall we really do it again? the powerful concept of replication is neglected in the social sciences,” *Review of General Psychology* **13** no. 2, (2009) 90–100.
- [8] National Academies of Sciences, Engineering, and Medicine, *Reproducibility and Replicability in Science*. National Academies Press, 2019.
- [9] B. A. Nosek, J. R. Spies, and M. Motyl, “Scientific utopia: Ii. restructuring incentives and practices to promote truth over publishability,” *Perspectives on Psychological Science* **7** no. 6, (2012) 615–631.

- [10] A. A. Alsheikh-Ali, W. Qureshi, M. H. Al-Mallah, and J. P. A. Ioannidis, “Public availability of published research data in high-impact journals,” *PLOS ONE* **6** no. 9, (2011) e24357.
- [11] T. E. Hardwicke, J. D. Wallach, M. C. Kidwell, T. Bendixen, S. Crüwell, and J. P. A. Ioannidis, “An empirical assessment of transparency and reproducibility-related research practices in the social sciences (2014–2017),” *Royal Society Open Science* **7** (2020) 190806.
- [12] D. G. Hamilton, M. J. Page, S. Finch, *et al.*, “How often do cancer researchers make their data and code available and what factors are associated with sharing?” *BMC Medicine* **20** (2022) 438.
- [13] H. Seibold, S. Czerny, S. Decke, R. Dieterle, T. Eder, S. Fohr, *et al.*, “A computational reproducibility study of PLOS ONE articles featuring longitudinal data analyses,” *PLOS ONE* **16** no. 6, (2021) e0251194.
- [14] R. Vicente-Saez and C. Martinez-Fuentes, “Open science now: A systematic literature review for an integrated definition,” *Journal of Business Research* **88** (2018) 428–436.
- [15] UNESCO, “UNESCO recommendation on open science,” 2023.
<https://www.unesco.org/en/open-science/about>.
- [16] Royal Society, “Science as an open enterprise,” 2012.
<https://royalsociety.org/topics-policy/projects/science-public-enterprise/report/>.
- [17] M. R. Munafò, B. A. Nosek, D. V. M. Bishop, *et al.*, “A manifesto for reproducible science,” *Nature Human Behaviour* **1** (2017) 0021.
- [18] B. A. Nosek, G. Alter, G. C. Banks, *et al.*, “Promoting an open research culture,” *Science* **348** no. 6242, (2015) 1422–1425.
- [19] cOAlition S, “Plan S principles,” n.d.
https://www.coalition-s.org/plan_s_principle.
- [20] European Commission, “Open science,” n.d.
https://research-and-innovation.ec.europa.eu/strategy/strategy-research-and-innovation/our-digital-future/open-science_en.
- [21] National Institutes of Health, “Final NIH policy for data management and sharing,” 2020.
<https://sharing.nih.gov/data-management-and-sharing-policy>.

- [22] A. Nelson and Office of Science and Technology Policy, “Ensuring free, immediate, and equitable access to federally funded research,” 2022.
<https://doi.org/10.21949/1528361>.
- [23] SCOAP³, “What is SCOAP³?,” n.d.
<https://scoap3.org/what-is-scoap3/>.
- [24] SCOAP³, “Open science elements,” n.d. <https://scoap3.org/journals-2025-2027/open-science-elements/>.
- [25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of NAACL-HLT 2019*, pp. 4171–4186. 2019.
- [26] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using siamese BERT-networks,” in *Proceedings of EMNLP-IJCNLP 2019*, pp. 3982–3992. 2019.
- [27] R. J. G. B. Campello, D. Moulavi, and J. Sander, “Density-based clustering based on hierarchical density estimates,” in *Lecture Notes in Computer Science*, vol. 7819, pp. 160–172. 2013.
- [28] J. Cohen, “A coefficient of agreement for nominal scales,” *Educational and Psychological Measurement* **20** no. 1, (1960) 37–46.
- [29] T. Basaglia, M. Bellis, J. Blomer, *et al.*, “Data preservation in high energy physics,” *European Physical Journal C* **83** (2023) 795.

Appendix A

Classification Schemata

This appendix reports shortened versions of the schemata used to classify Data Availability Statement (DAS) and Code Availability Statement (CAS) sections. For each category, the appendix includes three randomly selected exemplars, one hard-positive pair and one hard-negative pair, while preserving the remaining schema fields.

A.1 Data Availability Statement Schema

no_data

Definition. This category includes data availability statement sections stating that no data was used, analyzed or produced during the study.

Exemplars. (1) No data was used for the research described in the article (2) This manuscript has no associated data or the data will not be deposited (3) My study has no associated data

Hard positive. "This manuscript has no associated data or the data will not be deposited, as such there is no observational data related to this article where the necessary calculations and graphic discussion are already available in the manuscript" / This work does not involve any datasets. All results are derived from analytical and theoretical argument presented in the paper

Hard negative. This manuscript has no associated data / This manuscript has no associated code/software

Inclusion. A data availability section is included in this category when the content states that no data is associated, was used, generated or analyzed in the study described in the manuscript.

Exclusion. A data availability section is excluded from this category when the content states that there is data associated with this study regardless of the access mode (openly available, partially available, available upon request or not available).

Exclusion criteria. The section mentions there is data associated with the manuscript/study/paper; The section mentions there is data associated with the manuscript/study/paper openly available; The section mentions there is data associated with the manuscript/study/paper partially available; The section mentions there is data associated with the manuscript/study/paper available upon request; The section mentions there is data associated with the manuscript/study/paper which is not openly available.

Ambiguous cases. Sections that mention that the manuscript/study/paper has no associated code or software should not belong to this category. This category is exclusively associated to data. Sections mentioning information about availability of code/software belong to the category "incorrect". "This manuscript has no associated data or the data will not be deposited" is a specific set phrase used by the Journal of High Energy Physics. The expression is ambiguous at face value. Nonetheless, you should

ALWAYS classify statements using this phrase as "no_data". The last hard negative pair exemplifies the case where only implying or hinting at the fact that there might be no associated data with the study is not sufficient for a section to be classified as "no_data". The positive statement explicitly states that the work has no experimental data associated with it. Thus, it is classified as "no_data". The negative statement only implies that by saying the work is purely theoretical, but that is not sufficient to classify the section as "no_data".

open_access

Definition. This category includes data availability statement sections stating that all elements of research data can be accessed openly by anyone.

Exemplars. (1) The data used for this manuscript is available in ref. [71] (2) All data used from published material is contained in the article and duly referenced. (3) The PDF replicas for the JAMpol25 distributions produced in this analysis are available on LHAPDF [127]

Hard positive. Differential decay rate values are openly available. Raw correlator data is openly available. The code used to generate the graphs in Fig. 3 is available on GitHub / Differential decay rate values are openly available. Raw correlator data is openly available. The code used to generate the graphs in Fig. 3 is not publicly available

Hard negative. The data that support the findings of this article are openly available / "The data that support the findings of this article are openly available [60]. Gen 2 data can be found in Ref. [42]. Gen 2L ensembles are available on request. Details of the code and data presented can be found in Refs. [60,61]."

Inclusion. A data availability section is included in this study when the content states all research data used in the study is openly available. To be considered openly available, data is said to be available in the manuscript, references, supplementary material, online repositories, etc. Research data include every type of research output used, analyzed or generated during the study. That includes raw data, graphs, files, notebooks, numerical computations, etc.

Exclusion. A data availability section is excluded from this category when the content states that all or part of the research data is not openly available. A data availability section is excluded from this category when the content exclusively refers to code being openly available.

Exclusion criteria. The section mentions that no data was used or generated in this study; The section mentions that data is not openly available or is available upon request; The section mentions that part of the research data used, analyzed or generated is not openly available or available upon request; The section exclusively mentions that code is openly available, but not data.

Ambiguous cases. Sections that mention data AND code being openly available should belong to this category. As long as the section mentions that every element of research data is openly available, it should belong to this category. This even if the sections also mentions information about code availability. Information about data availability always takes precedence in classification. Sections that mention that at least one element of research data is not openly available should be excluded from this category. Every single element of research data mentioned must be openly available for a section to be included in this category, except for code. In the last hard positive pair, the two statements only differ as per the accessibility condition of code. As for elements of research data, both sections state that all elements are openly available. Since data takes precedence in the classification task, different access conditions for code do not determine different categorization of the two statements. Code only influence classification if no other element of research data is mentioned. Therefore, in this case both sections are classified as "open_access". The exemplar stating: "The data that support the findings of this article are openly available [18–20], embargo periods may apply." should be classified as part of this category. Arguably, the statement is ambiguous and could belong to other categories. Nonetheless, we are willingly deciding that a statement mentioning that data is openly available AND that an embargo period may apply should be classified as part of this category, especially if the embargo period is underspecified. Sections stating that the release and preservation of data is guided by the CMS data preservation, reuse and open access policy MUST be classified as "open_access".

partial_access

Definition. This category include data availability statements stating that only part of the elements of research data are openly available to anyone.

Exemplars. (1) "The data for this manuscript are not available. The values in the plots and tables as-

sociated to this article are stored in HEPDATA (<https://www.hepdata.net/record/157579/>)." (2) "Some of the data that support the findings of this article are openly available [14,65] . Energy levels for the isospin-0 pi-pi system by HadSpec are not publicly available because they are owned by a third party and the terms of use prevent public distribution. Results of the analyses are not publicly available. The data are available from the authors upon reasonable request" (3) "The DNN model is available from the GitHub repository [60]. The data that support the findings of this article are not publicly available. The data are available from the authors upon reasonable request"

Hard positive. "Tabulated results are provided in supplementary material. The DDN model is available in the GitHub repository. Gen2 data are available upon request. Three-point function data is openly available. Raw correlator data is publicly available in Zenodo" / Some of the data that support the findings of this article are openly available

Hard negative. "Gen 3 data is available upon request. Tabular data is available upon request. The baryon acoustic oscillations (BAO) dataset is publicly available." / "Tabulated results are available upon request. Energy levels for the isospin-0 pi-pi system are available upon request. The code used in the analysis is openly available in the GitHub repository. The PDF replicas for the JAMpol25 distributions is not available."

Inclusion. A data availability section is included in this category if the content states that at least one element of research data is openly available. That includes section stating that "some" data is openly available.

Exclusion. A data availability section is excluded from this category if the content states that all elements of research data are openly available, available upon request or not available.

Exclusion criteria. The section mentions that all data is openly available; The section mentions that all data is not available; The section mentions that all data is available upon reasonable request; The section mentions that some elements of research data are available upon request and some are not available.

Ambiguous cases. Sections mentioning that every element of research data is not openly available, but code used in the study is openly available, should NOT belong to this category. This category exclusively refers to cases where at least one element of research data (not code) is openly available. Information about code/software used to generate data should be considered information about code. Therefore it should received lower priority than information about data in the classification task. If a section exclusively mentions information about code used to generate data, it should be classified as "incorrect". The "negative" statements in the hard negative examples includes information about code being openly available. Since code/software is the only element of this studies that is indicated as openly available, these sections should not belong to the "partial_access" category. Information about code availability should not be considered. It should instead belong to the category "restricted_access" since at least one element of research data is mentioned as being available upon request. The examples provided as hard positives belong to the same category because in both cases, sections describe at least one element of research data as openly available. This regardless of the access condition of all other elements of research data listed. Sections that mention that some data is available upon request while some data is not available should NOT belong to this category. They should instead belong to the category "restricted_access". This is the case of the second example of hard negative pair, where the only difference between the two statements as for the accessibility conditions of elements of research data (not code) is that the first mentions the last element of research data discussed as being "available in Zenodo", whereas the second mention the last element of research data as being "available upon request".

restricted_access

Definition. This category includes data availability statements stating that data is available from a source upon request

Exemplars. (1) Data underlying the figures of this paper are available from the authors upon reasonable request. (2) "The data that support the findings of this article are not publicly available because they are owned by a third party and the terms of use prevent public distribution. The data are available from the authors upon reasonable request." (3) "The data that support the findings of this article are not publicly available because they contain sensitive personal information. The data are available from the authors upon reasonable request"

Hard positive. Data will be made available on reasonable request. / The full Belle II dataset can be made available upon contacting the High Energy Accelerator Research Organization

Hard negative. "The Wolfram Mathematica notebook are available upon reasonable request. The Gen 4 dataset is available from the authors upon reasonable request Three point function data will be made

available upon request" / "Tabulated results are available upon request. Gen 5 data are available upon request. The Hadron Collider data can be made available from the authors upon request. The DNN model is available from the GitHub repository [60]"

Inclusion. A data availability statement is included in this category when the content states that at least one element of research data is available from a source upon requesting it, while NOT stating that any of the elements of research data is openly available.

Exclusion. A data availability statement is excluded from this category if it states that at least one element of research data is openly available, all elements of research data are available, or no element of research data is available.

Exclusion criteria. The section mentions that at least one element of research data is openly available; The section mentions that all elements of research data are not available; The section mentions that data is not available while code is available either openly or upon request.

Ambiguous cases. Sections mentioning that data will be made available in a data repository should not be considered as restricted. Restricted access should apply to cases where data has to be requested from a source like authors, research groups or organizations. If a section states that even a single element of research data is openly available, it should not be included in this category regardless of the number of other elements of research data that are available upon request. This is imperative, one single element of research data being openly available is a sufficient condition for exclusion in this category. If at least one element of research data is described as available upon request and no other element is described as being openly available, the section should be classified in this category. As shown by the last hard negative pair, only stating that "data [...] are not publicly available" is not sufficient to classify a section as "restricted_access". The section must also state that data could be obtained by requesting it. A section only stating that data is not publicly available should instead be classified as "no_access".

no_access

Definition. This category includes data availability statements stating that data cannot be made available under any circumstances.

Exemplars. (1) The data used in this study cannot be made public (2) Energy levels for the isospin-0 pi-pi system by HadSpec are not publicly available because they are owned by a third party and the terms of use prevent public distribution. (3) The data used in this study have not been deposited

Hard positive. The data used in this study have not been deposited / The data analyzed in this manuscript cannot be made available because they include personal data

Hard negative. The data that has been used is confidential and it has not been deposited in an online repository / "The data that has been used is confidential and it has not been deposited in an online repository. Results of the analysis of data used can be made available upon reasonable request"

Inclusion. A data availability statement is included in this category when the content implies that data used, analyzed or generated in the study is not accessible under any circumstances. This criterion for inclusion is independent of a reason being specified for lack of accessibility and if specified, of the reason itself.

Exclusion. A data availability statement is excluded from this category if it states that data can be accessed, regardless of the accessibility conditions. This includes cases where all data can be accessed publicly, only part of data can be accessed publicly, all data can be accessed but under specific circumstances (e.g. upon request) or some data can be accessed under specific circumstances. As long as at least one element of research data can be accessed under some condition, data availability statements should be excluded from this category.

Exclusion criteria. The section mentions that all or part of the data is publicly available; The section mentions that all or part of the data is available under specific conditions (e.g. upon request); The section mentions that code cannot be made available, not data..

Ambiguous cases. Sections that mention that all elements of research data are not available except for one should not be included in this category. One single element of research data being available under some condition suffice to exclude a section from this category. The reason why data cannot be made accessible is irrelevant for classification into this category. Different sections might specify different reasons (e.g. confidentiality, terms of use, no reason) why data cannot be made accessible under any circumstances. As long as the content of a section states or implies that data is not accessible, the section should be classified as part of this category.

incorrect

Definition. This category includes data availability statements stating information not pertaining to data availability.

Exemplars. (1) This manuscript has no associated code/software. (2) This article is distributed under the terms of the Creative Commons Attribution License (CC-BY4.0), which permits any use, distribution and reproduction in any medium, provided the original author(s) and source are credited (3) This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology

Hard positive. The code is available at the GitHub repository link provided in the manuscript. [2https://github.com/salnikov-dmitry/lsw](https://github.com/salnikov-dmitry/lsw) / This article is distributed under the terms of the Creative Commons Attribution License (CC-BY4.0), which permits any use, distribution and reproduction in any medium, provided the original author(s) and source are credited

Hard negative. The code associated with this manuscript is openly available / The code used to generate data in this study is openly available. Raw analysis data can be made available upon request.

Inclusion. A data availability statement is included in this category when it contains information not pertaining to data availability. For instance, statements talking about code are included in this category if they exclusively state information about code. Statements mentioning information about an article's general accessibility conditions (e.g. license associated to the article) are also included in this category.

Exclusion. A data availability statement is excluded from this category when the content mentions any information about availability of an element of research data. That includes any element that might be considered as part of research data except for code (e.g. numerical values, measurements, tables, figures, images, etc.).

Exclusion criteria. The section mentions any information about availability of any element of research data..

Ambiguous cases. Sections that only mention information about code/software used to generate data should be considered sections about code. Therefore, they should be classified as part of this category. Sections that mention information that might refer to data availability (e.g. "Not applicable") should not belong to this category, but should rather be classified as part of the category "unclear". Sections that mention contradictory information about data availability (e.g. stating both that a study has no associated data AND that data can be made available) should not belong to this category, but should rather be classified as part of the category "unclear". Sections that mention any information about data availability are excluded from this category. This also includes sections that additionally mention information about code availability. In the third example of hard negatives, given the second sentence also mention information about data availability, that should take precedence. Therefore, even if the section mentions information about code availability, it should not be classified as "incorrect". Information about data availability should guide classification and the section should therefore be classified as "restricted _access" Exemplar number 8 mentions information about the files containing the code to reproduce the data. The subject of the section are files containing code, therefore the section should be classified as "incorrect". Exemplar number 9 mentions "Mathematica" files. Those are to be considered as code and therefore the section should be classified as "Incorrect". The same reasoning applies to sections mentioning Mathematica "notebooks".

unclear

Definition. This category includes data availability statements stating ambiguous or unclear information about data availability.

Exemplars. (1) Not applicable. (2) All relevant data generated or analyzed during this study have been included in the article. Data availability No data was used for the research described in the article. (3) "The data samples (HepMC and EventCalc-SHiP) generated during this study are available from the corresponding author on reasonable request: HepMC samples (M. Bahmani), EventCalc-SHiP samples (A. Reghunath). My manuscript has no associated data."

Hard positive. This study is purely theoretical / All relevant data generated or analyzed during this study have been included in the article. Data availability No data was used for the research described in the article.

Hard negative. This is a purely theoretical study / This is a purely theoretical study. Therefore, the study has no associated data

Inclusion. A data availability statement is included in this category when the content either states ambiguous/contradictory information about data availability or when the content implies information about data availability (e.g. "This is a purely theoretical study").

Exclusion. A data availability statement is excluded from this category when the content states information about data availability that clearly indicates the condition of accessibility of the elements of research data involved.

Exclusion criteria. Section mentions information about data availability that does not logically contradict information stated previously.; Section mentions information not pertaining to data availability..

Ambiguous cases. Sections that initially include information that only implies information about data availability, but later clarify data availability/access conditions should not go into this category. Those sections should go into the category consistent with the clarified data availability condition. For instance, a section stating that a study is purely theoretical and therefore has no associated data should go in the "no_data" category. Sections that exclusively mention ambiguous/contradictory information about code should not belong to this category, but should instead be classified under the category "incorrect" Sections that contain information about data availability that is apparently nonsensical or circular (e.g. "Data cannot be made available for reasons disclosed in the data availability statement") should belong to this category. The last hard negative pair showcases an example where the special statement "The data that support the findings of this article are openly available [18–20], embargo periods may apply." occurs. This statement is arguably ambiguous, but it should nonetheless be classified as "open_access".

A.2 CAS Manuscript Code Availability Schema

no_code

Definition. This category includes code availability statements stating that no Author-Produced Software Artifact (APSA) is associated with the manuscript/study

Exemplars. (1) "this manuscript has no associated code/software. [author's comment: code/software sharing not applicable to this article as no code/software was generated or analysed during the current study. The current work has already included a comprehensive analysis and the corresponding calculations.]." (2) this manuscript has no associated code/software. [author's comment: in this theoretical study, all analyses and derivations were performed using established theoretical frameworks and analytical methods.]. (3) this manuscript has no associated code/software. [author's comment: all codes and software used in this work are publicly available, and all parameters have been thoroughly documented in the manuscript to ensure full reproducibility.

Hard positive. this manuscript has no associated code/software. [author's comment: in this theoretical study, all analyses and derivations were performed using established theoretical frameworks and analytical methods.]. / "the manuscript has no associated code/software. [author's comment: only standard numerical methods available in widely used computer algebra systems were employed; no original code or software was generated.]."

Hard negative. this manuscript has no associated code/software. [author's comment: all figures have been created using mathematica software. / the manuscript has no associated code/software. [author's comment: all the figures are generated by the mathematica program, which can be requested from the authors.].

Inclusion. A code availability statement section is included in this category when the content states that no Author-Produced Software Artifact is associated, was generated, adapted or curated in the course of the study. Only and only if the information in the section does NOT state that even a single Author-Produced Software Artifact is presented with the manuscript, the section is included in this category.

Exclusion. A code availability statement section is excluded from this category when the content states that AT LEAST ONE Author-Produced Software Artifact is presented with the manuscript. This is true EVEN IF the first part of the section states that the study/manuscript has no associated code/software. As long as some other part of the section mentions an Author-Produced Software Artifact (APSA), the section is excluded from this category.

Ambiguous cases. At times, sections first state that there is no code/software associated with the manuscript BUT later mention either an Author-Produced Software Artifact, or External Software Tool. You must carefully check that no part of the section explicitly mentions an Author-Produced Software Artifact as being created, generated, or used for some aspect of the study. If a section mentions an

Author-Produced Software Artifact, it should NOT be categorized as part of the "no_code" category. If a section EXCLUSIVELY mentions an External Software Tool being used, it should be categorized as part of the "no_code" category when the first part mentions no code/software associated with the study. The following exemplar: "my manuscript has no associated code/ software. [author's comment: code/software sharing not applicable to this article as no code/software was generated during the current study. The numerical analysis was performed using the public open source software flavio.]," states that there is no code associated with the study. It then mentions that an open source software was used. This section should be classified as part of this category because an open source software tool should NOT be considered as an Author-Produced Software Artifact. An open source software tool should instead be considered as an External Software Tool. Therefore, information about the open source software tool should EXCLUSIVELY be used to classify sections along the dimension "external_tool_availability", not alongside the current dimension "manuscript_code_availability". Exemplar from 15 to 21 are similar cases among others. The negative example of the third hard negative pair should NOT belong to the no_code category, since "code necessary to recreate an analysis" should count as an APSA, as well as "code and configurations written in support of this analysis". This is because it can be inferred that these elements of code/software are something directly related to the manuscript, and therefore most likely artifacts that were produced and can be made available by the authors of the study. The "Atlas collaboration software" mentioned as open source is an EST and therefore its condition of access (i.e. open source) should be irrelevant for the classification in this dimension. The positive example of the same pair exclusively mentions an EST (i.e. fenicsx open Python package). Therefore, the section is classified as "no_code" along the "manuscript_code_availability" dimension. The fourth hard negative pair exemplifies the difference between Author-Produced Software Artifact and External Software Tool. The positive example mentions "mathematica software". This, by itself, counts as an EST, as evidenced by the use of the verbal expression "used to" and by prior knowledge of Wolfram Mathematica being a proprietary software. The negative case ALSO mentions a "Mathematica program" BUT also specifies that this can be "requested from the authors". The property of being requestable by the author therefore qualifies this as an APSA. Given an APSA is mentioned, this section is NOT classified as part of the "no_code" category in the dimension at hand. The following exemplar mentions "parameters" being documented in the manuscript: "this manuscript has no associated code/software. [author's comment: all codes and software used in this work are publicly available, and all parameters have been thoroughly documented in the manuscript to ensure full reproducibility." "Parameters" in this context cannot reasonably be considered as APSA, as they are not explicitly described as research software artifacts. Therefore, the section is classified as "no_code" since no APSAs are mentioned. The following exemplar is a VERY IMPORTANT case: "This manuscript has associated code/software in a data repository. [author's comment: the analysis in this study made use of publicly available cosmological codes: class (cosmic linear an isotropy solving system): https://github.com/lesgourg/class_public cobaya (code for bayesian analysis): <https://github.com/cobayasampler/cobaya>. No new proprietary software was developed in the course of this work.]" It shows how EVEN IF the first part of the code availability statement says there IS code/software associated with the study, this does not necessarily imply that there are APSAs associated with it. In this case, after the first part, only the availability of third-party software tools/packages is discussed. These tools are ESTs and should NOT influence classification in this dimension. The section is therefore classified as "no_code" since no APSAs are mentioned. The following exemplar is an ambiguous case: "My manuscript has associated code/software in a data repository. [author's comment: the neural network global fit of cffs used in this analysis is available in the partons framework [84]. The code of this framework is open source and can be found online at <https://drf-gitlab.cern.ch/partons/core/partons> on version 3 of the gpl (gplv 3).]" The phrase "neural network global fit" refers to a methodological component of the analysis, but there is no explicit indication that this corresponds to a software artifact implemented, curated, or produced by the authors. Therefore, no APSA is explicitly identified in the section. Although third-party software platforms are mentioned, such information is not relevant for the "manuscript_code_availability" dimension. The section is therefore classified as "no_code".

open_access

Definition. This category includes code availability statement sections stating that all explicitly-mentioned Author-Produced Software Artifact associated with the study can be accessed openly by anyone.

Exemplars. (1) "my manuscript has associated code/ software in a data repository. [author's comment: the code/software used for this study is openly accessible under the licenses apache 2. <https://gitlab.cern.ch/lhcb/allen> and gpl version 3 <https://gitlab.cern.ch/lhcb/rec>.]" (2) this article has associated code in a code repository. (3) this manuscript has no associated code/software. [authors' comment: the code/software used for the analysis is publicly available in the git hub repository, at the links

<https://github.com/alismw/aliroot> and <https://github.com/alismw/aliphysics/>].

Hard positive. this article has associated code in a code repository. The data on which this paper is based may be reproduced using the code and scripts found on github at <https://github.com/zgmou/sewedthimble>. / "code/software will be made available on reasonable request. [authors' comment: the code used for the event generation, the network architecture and its training can be found at https://github.com/amorandini/param_hunt. an example jupyter notebook which explains how to derive the plots and results in this paper is also provided.]"

Hard negative. the lattice generation and gf analysis presented in this work were performed using simulate qcd [49] and the milc code. The analysis code is openly available in [91, 93, 94]. / "software/code that is associated with this publication and that is publicly available is referenced within the publication content. Specific analysis software/code used to produce the results shown in the publication is preserved within the lhc collaboration internally and can be provided on reasonable request, provided it doesn't contain information that can be associated with unpublished results."

Inclusion. A code availability statement section is included in this category when all Author-Produced Software Artifacts that are mentioned are described as being openly available. A section is also included in this category if the first part mentions that there is no code/software "associated" with the manuscript BUT it then states that all APSAs mentioned are openly available. The access condition of External Software Tools mentioned in the section should NOT matter for classifying a section as "open_access". Only access conditions of APSAs are relevant for classification in this dimension.

Exclusion. A code availability statement section is excluded from this category when it mentions that all or part of the Author-Produced Software Artifacts mentioned are NOT openly available. If no APSA is explicitly or implicitly identifiable, the section MUST NOT be classified as open_access.

Exclusion criteria. The section ONLY states that there is no code/software associated with the study/-manuscript; The section states that only SOME APSAs are openly available; The section states that one or more APSAs are not openly available; The section states that one or more APSAs are not available under any circumstance.

Ambiguous cases. You MUST classify statements stating "this article has associated code in a code repository." as part of the "open_access" category. Even though access conditions are underspecified, we deliberately treat this statement as a marker of "open_access" if it appears alone. The statement "My manuscript has associated code/software in a data repository. [author's comment: all analytical computations reported in this manuscript have been rechecked in the computer algebra system maple. Two maple sheets corresponding to the extreme and non extreme cases can be found in supplementary materials and in the repository below. The discretization of differential operators (23)–(25) and (41)–(43) using the tchebyshev type spectral method is equally performed in maple computer algebra system. Finally, the numerical solution of the resulting quadratic eigenvalue problem (48) is performed in matlab software using the polyeig function. All these materials are freely accessible at the following repository <https://github.com/dutykh/ellisbronnikov/>]." and the second section in the first hard positive pair is classified as "open_access" because the only APSA mentioned "maple sheets" is described as being openly available. All other elements of code/software are ESTs. The statement "this manuscript has no associated code/software. [authors' comment: the code used in this project is publicly available at the following repository: <https://github.com/opengpd/gpddatabase> (see ref. [78])." showcases the notion that you can IGNORE the first part of a code availability statement saying there is no code associated with the study IF what follows clearly mentions an APSA. In this case, in the author's comment, there's mention of "all code discussed in this paper" that is a signal for an APSA being present. Since the only APSA mentioned is described as openly available, the section is classified as "open_access" The first section of the second hard positive pair showcases an important concept. In the section, "mixing matrices [...] contained within the mathematica package" are mentioned. These should be considered APSA because their availability in supplemental material AND the fact they were "computed" as part of the study indicates they are author-controlled/produced. Even if a third-party software is used as container/computational environment for an author-produced artifact, that artifact should still be considered as an APSA. The positive example of the fourth hard positive pair mentions "code provided at the link". Considering the code is "provided" with an associated link, this should be considered as an APSA. Since it is the only APSA mentioned and it's openly available, the section is classified as "open_access".

partial_access

Definition. This category includes code availability statements stating that only part of the Author-Produced Software Artifacts mentioned are openly available to anyone.

Exemplars. (1) code/software will be made available on reasonable request. [authors' comment: the code is also available on github at <https://github.com/croi900/genesys>.] (2) "this manuscript has associated code/software in a data repository. [authors' comment: the code/software generated during and/or analysed during the current study is available in the repository: <https://github.com/bfuku/dmsimpt>. software configuration cards can be obtained upon request.]" (3) "Software/code that is associated with this publication and that is publicly available is referenced within the publication content. Specific analysis software/code used to produce the results shown in the publication is preserved within the lhcb collaboration internally and can be provided on reasonable request, provided it doesn't contain information that can be associated with unpublished results."

Hard positive. The python scripts used for numerical computations are available on Zenodo. The mathematica packages used for numerical calculations are available on request / "code/software will be made available on reasonable request. [authors' comment: the code used for the event generation, the network architecture and its training can be found at https://github.com/amorandini/param_hunt. An example jupyter notebook which explains how to derive the plots and results in this paper is also provided.]"

Hard negative. code/software will be made available on reasonable request. [authors' comment: the code is also available on github at <https://github.com/croi900/genesys>.] / "code/software will be made available on reasonable request. [author's comment: to date, the mocanlo, stripper, mulbos, and bbmc codes are not public tools, but numerical results can be made available upon reasonable request. the sherpa (version 3), powheg box (version res, package v v pol), and mad graph (version 5) are publicly available event generators.]"

Inclusion. A code availability statement section is included in this category when AT LEAST ONE but NOT ALL Author-Produced Software Artifacts mentioned are described as openly available.

Exclusion. A code availability statement section is excluded from this category when a section mentions that APSA mentioned are either ALL openly available, ALL under restricted access conditions, ALL unavailable or when SOME are under restricted access conditions AND SOME are unavailable. At least one but not all ESTs mentioned being openly available SHOULD NOT qualify a section as being "partial_access" for the dimension "manuscript_code_availability". Access conditions of ESTs are irrelevant for classification in this dimension.

Ambiguous cases. The statement "This manuscript has associated code/software in a data repository. [authors' comment: code sharing is not applicable to the analysis software. The software used in the model dependent interpretation of the exclusion limits is available in the zenodo repository <https://doi.org/10.5281/zenodo.5844010>.]" mentions "the analysis software" and "software used in model dependent interpretation". These are arguably difficult to cleanly identify as either APSA or EST. As mentioned in the general instructions provided in the initial prompt, these elements of code/software should be heuristically interpreted as APSA due to the section explicitly stating there is code associated with the manuscript, and no identifiable, explicit third-party ownership attribution. Therefore, the section is classified as "partial_access" since at least one but not all elements are described as openly available. The statements "code/software will be made available on reasonable request. [author's comment: code available at <https://zenodo.org/records/12576982>.]" and "code/software will be made available on reasonable request. [authors' comment: the code is also available on github at <https://github.com/croi900/genesys>.]" first mention code that can be made available upon request, while later mentioning code being made available in two repositories. Indications on provision and the first part mentioning code available upon request qualifies the elements of code mentioned as APSA. Given there is mention of code being available upon request AND code being openly available in a repository, AT LEAST ONE but NOT ALL APSAs mentioned are indicated as openly available. Therefore, the section is classified as "partial_access". The statement "Software/code that is associated with this publication and that is publicly available is referenced within the publication content. Specific analysis software/code used to produce the results shown in the publication is preserved within the lhcb collaboration internally and can be provided on reasonable request, provided it doesn't contain information that can be associated with unpublished results." is an arguably ambiguous case. Two distinct elements of software are identifiable given the section makes two different, non-contradictory access claims for two objects (i.e. one "publicly available", the other available "on reasonable request"). The second is an APSA since it's described as a study-specific software, internal, whose access is request-based and that's being used to produce the results in the analysis. The first (i.e. "Software/code that is associated with this publication and that is publicly available") defaults to APSA classification due to lack of strong EST indicators and presence of weak APSA signals, specifically the linguistic mark "associated with the study". Therefore, the final classification is "partial_access", since at least one (even if weakly inferred as such) BUT not all mentioned APSA are openly available. The uncertainty around correctly identifying one element as an APSA or EST should be reflected in the level of confidence you provide alongside the classification. The negative example of the first hard negative pair should not be included in this category

because only ESTs are described as openly available. Specifically, "the software for detector simulation [...]" and the "root open source software" are the identifiable elements of software described as open. Considering there is no indication of authorship or of these elements being made available by the authors, they are classified as ESTs. Therefore, their access condition does NOT influence classification for the "manuscript_code_availability" dimension. Configuration files and macros should instead be considered APSAs and should therefore guide classification. Since none of these elements is described as openly available, the section is not classified as "partial_access". Similarly to the case discussed above, also the negative example of the second hard negative only mentions ESTs as openly available and should therefore NOT be categorized as "partial_access". There is only one, weakly inferred ASPA (i.e. "code/software [...] available on reasonable request") mentioned in the section which is not described as open. The third hard negative pair showcases a case where the initial boilerplate sentence can be used to qualify access conditions for an element described subsequently. The initial sentence does not have an explicit referent, but later qualifies access conditions for an APSA mentioned. Specifically, in the positive case, the author mentions there is code associated with the study in a repository, and later mentions an APSA for "numerical computations". In such a case, information given in the initial statement is related to the APSA described in the author's comment. The "collaboration code" mentioned is an EST. Given the APSA mentioned is qualified as open, at least one but not all (i.e. "config files" are not) APSAs are described as openly available, and the section is therefore classified as "partial_access". The negative example is structurally similar in that the initial sentence also qualifies the access condition of a later-mentioned APSA (i.e. "simulation code"), but it qualifies it as being available upon request. Therefore, no APSA mentioned is described as openly available, and the section is not classified as "partial_access". In the positive example of the fourth hard negative pair, the first distinct identifiable element of code/software is an ASPA, as evidence by the authorship linguistic signal "our code". Therefore, the section is classified as "partial_access" since at least one but not all elements are described as openly available. In the negative example of the pair, the first distinct identifiable element of code/software is an EST. This is because there is a generic reference to "codes" already available with no authorship or provision linguistic signal. Since no identifiable APSA is described as openly available, the section is NOT classified as part of this category.

restricted_access

Definition. This category includes code availability statements stating that the Author-Produced Software Artifacts mentioned are available upon requesting them.

Exemplars. (1) code/software will be made available on reasonable request. [authors' comment: no additional remarks.]. (2) code/software will be made available on reasonable request. [author's comment: the current study uses publicly available software relevant to high energy physics, and the analysis code is available from the corresponding author on reasonable request.]. (3) code/software will be made available on reasonable request. [authors' comment: the code for models developed in the context of this work will be made available with future releases of the herwig event generator.].

Hard positive. "This article has associated code in a code repository. all our analytical results for the observables (impulse, spin kick, scattering angle) up to and including the (physical) 7 pm order as well as the ssc preserving wilson coefficients are provided in an accompanying zenodo.org [104] submission. The code to generate these, that can be used to generate even higher orders or work with different metrics, can be made accessible by reasonable request to the authors." / code/software will be made available on reasonable request. [author's comment: the simulation code is based on the milc collaboration's code, publicly available at <https://github.com/milcqcd/>.]

Hard negative. code/software will be made available on reasonable request. [author's comment: the code is currently insufficiently documented for public release, but we will be happy to provide it along with usage instructions upon reasonable request.]. / code/software will be made available on reasonable request. [authors' comment: code/software sharing is not applicable to this paper.].

Inclusion. A code availability statement is included in this category when the content states that AT LEAST ONE Author-Produced Software Artifacts is available from a source upon requesting it, while also NOT stating that ANY OTHER Author-Produced Software Artifact is openly available.

Exclusion. A code availability statement is excluded from this category if it states that AT LEAST ONE Author-Produced Software Artifact is openly available, ALL Author-Produced Software Artifacts are available, or NO Author-Produced Software Artifact is available.

Ambiguous cases. The initial sentence in the statement "this manuscript has associated code/software in a data repository. [authors' comment: the code generated during and analysed during the current study

is available from the corresponding author on reasonable request.].” says the manuscript has “associated code in a data repository”. This sentence alone would be an indication of the existence of an APSA to be classified as “open_access” BUT this interpretation is later contradicted by the following sentence stating that the APSA in question is available upon request. This ambiguous case is also captured by the fourth hard negative pair and by the statement “this article has associated code in a code repository. The code to reproduce the figures of this article will be shared upon request.” In the statement “code/software will be made available on reasonable request. [author’s comment: the code we created for this study can be shared upon request. we are still developing the code and we would like to make it publicly available at a later stage.]”, authors mention that they would like to “make it publicly available at a later stage”. This by itself is NOT enough to classify the ASPA mentioned as “open_access”. The classification should only be guided by the current access conditions specified in the section. That is, the ASPA “can be shared upon request” and therefore the section is classified as “restricted_access” along the “manuscript_code_availability” dimension. The statement “code/software will be made available on reasonable request. [author’s comment: the software used for event generation and analysis is fully open source. Parameter settings can be made available upon reasonable request.]” mentions “Parameter settings”. Those should be considered as APSA. Therefore, the section is classified as “restricted_access” along the “manuscript_code_availability”. The statement “this manuscript has no associated code/software. [author’s comment: the code for this work is written in python and handles. It is available in source format from the corresponding author upon reasonable request for colleagues needing it for similar analyses.]” together with many others showcase the ambiguous case whereby the initial boilerplate sentence makes a negative availability claim (i.e. no code associated) BUT are later contradicted by other sentences mentioning an APSA and its related access condition. As per the rules explained in the general prompt, the initial boilerplate sentence’s assertion that no code is associated with the study is overridden by subsequently provided information. Therefore, these statements are classified following the access condition specified for the ASPA, which in cases mentioned for this category is “restricted_access”. In the statement “code/software will be made available on reasonable request. [author’s comment: all data included in this manuscript were generated using the iebe music code, which is publicly accessible on github. The code analysed during the current study is available from the authors on reasonable request.]” the only element of software that is described as openly available is an EST since no authorship or provision signal is present, and the code is described as “already publicly available”. The only APSA present is described as available upon request. Therefore, the section is classified as “restricted_access” along the “manuscript_code_availability” dimension. The second statement in the third hard positive pair shows an example of where the access conditions specified in the initial boilerplate statement can be attributed to a subsequently mentioned APSA. This is possible because all criteria necessary for this attribution are met. There is only one eligible APSA the access conditions can be attributed to and the APSA mention does not specify any other access condition.

no_access

Definition. This category includes code availability statements where Author-Produced Software Artifacts are not accessible under any access condition explicitly specified in the statement.

Exemplars. (1) “My manuscript has no associated code/software. [author’s comment: code/software sharing for this article is not applicable because no software was generated during the process and codes used by authors were not written for publication purposes. For colleagues who want to perform the same calculations, following the theoretical explanations which are provided throughout the text will be sufficient.]” (2) “code/software cannot be made available for reasons disclosed in the code availability statement. [authors’ comment: the code/software developed for this study is not publicly available in order to protect intellectual property and avoid unintended reuse or misinterpretation. However, the corresponding author is happy to provide methodological guidance or relevant implementation details upon reasonable request via email, especially for academic and noncommercial research purposes.]” (3) “Code/software cannot be made available for reasons disclosed in the code availability statement. [authors’ comment: the code utilized in the current study is not publicly available because co-owned by other authors, but details can be shared by the corresponding author on reasonable request.]”

Hard positive. The code associated with this manuscript cannot be made available due to the terms of use preventing public distribution / “This manuscript has no associated code/software. [author’s comment: the calculations were performed with short, standard mathematica scripts that simply evaluate the same results. These scripts contain no new algorithms or software packages and therefore offer no added value to the reader.]”

Hard negative. The software developed in this work is not publicly available / The numerical analysis

software used in this work is not publicly available

Inclusion. A code availability statement is included in this category when the content implies that the Author-Produced Software Artifacts described are NOT accessible under any condition to the reader. This criterion for inclusion is independent of a reason being specified for lack of accessibility and if specified, of the reason itself. A code availability statement is ALSO included in this category when an APSA is mentioned without any access condition BUT the initial boilerplate sentence says there is no code associated with the study. In such cases, the APSA is to be considered NOT available and the section is classified as part of this category

Exclusion. A code availability statement is excluded from this category if it states that an Author-Produced Software Artifact can be accessed, regardless of the access conditions. This includes cases where all identifiable APSAs can be accessed publicly, only part can be accessed publicly, all can be accessed but under specific circumstances (e.g. upon request) or some can be accessed under specific circumstances. As long as at least one identifiable Author-Produced Software Artifact can be accessed under some condition, code availability statements should be excluded from this category.

Ambiguous cases. In the statement "This manuscript has no associated code/software. [authors' comment: atlas collaboration software is open source, and all code necessary to recreate an analysis is publicly available. The athena (<http://gitlab.cern.ch/atlas/athena>) software repository provides all code needed for calibration and uncertainty application, with configuration files that are also publicly available via docker containers and cv mfs. The specific code and configurations written in support of this analysis are not public; however, these are internally preserved.]" only one identifiable APSA is present (i.e. "The specific code and configuration") and it is described as "not public" and "internally preserved". That qualifies it as not obtainable by the reader. Therefore, the section is classified as "no_access" under the "manuscript_code_availability" dimension. The second example of the first hard positive pair is a statement that satisfies the second rule for inclusion. Specifically, one APSA is mentioned (i.e. "short, standard mathematica scripts") with no accompanying specified access condition. Since the initial boilerplate sentence states that the manuscript "has no associated code/software", the section is therefore classified as "no_access" under the "manuscript_code_availability" dimension. The negative case of the first hard negative pair does NOT satisfy the second rule for inclusion in this category. This is because even though the initial boilerplate sentence states that the manuscript "has no associated code/software", only an EST is mentioned in the subsequent sentence. Since there is no identifiable APSA presented, the section CANNOT be classified as "no_access" In the third hard negative pair, the positive example gives information about access condition of an APSA, as evidenced by linguistic signals of authorship (i.e. "developed"). Since the software is described as not publicly available, the section is classified as "no_access". The negative example only mentions an EST not being publicly available (i.e. "The numerical analysis software used"). Since no APSA is mentioned, the section is not classified as "no_access". The statement "code/software cannot be made available for reasons disclosed in the code availability statement. [authors' comment: the code/software developed for this study is not publicly available in order to protect intellectual property and avoid unintended reuse or misinterpretation. However, the corresponding author is happy to provide methodological guidance or relevant implementation details upon reasonable request via email, especially for academic and noncommercial research purposes.]" explicitly states that code is not publicly available and does not specify any condition under which code can be accessed. "Implementation details" and "methodological guidance" are not APSAs. Therefore, access conditions relative to these objects does NOT influence classification as per code availability. The statement "Code/software cannot be made available for reasons disclosed in the code availability statement. [authors' comment: the code utilized in the current study is not publicly available because co-owned by other authors, but details can be shared by the corresponding author on reasonable request.]" is a similar case where "details" cannot be considered APSAs and therefore the section is also classified as "no_access"

incorrect

Definition. This category includes data availability statements that do not provide information about availability, use or existence of identifiable elements of software, whether Author-Produced Software Artifacts or External Software Tools.

Exemplars. (1) The graphs in Figs. 1 and 2 were plotted from equations appearing in this paper and in reference [5], as detailed in the figure captions (2) The author declares that there are no conflicts of interest (3) The numerical values of the measured differential decay rates and their correlation matrices in full precision, as well as the HistFactory likelihood, are openly available in Ref. [37]

Hard positive. Simulations were carried out in this work / Data, including MCMC chains from the fit without the RC applied, have been released in Ref. [68].

Hard negative. In this work, we used standard numerical methods / In this work, we used standard numerical methods using Mathematica

Inclusion. A code availability statement is included in this category if it DOES NOT provide any information on the availability, use or existence of any identifiable element of code/software, whether APSA or EST. For instance, statements EXCLUSIVELY providing information about data availability are included in this category. Statements mentioning information about an article's general accessibility conditions (e.g. license associated to the article) are also included in this category.

Exclusion. A code availability statement is excluded from this category if it provides information on availability, use or existence of AT LEAST ONE identifiable element of code/software, whether APSA or EST.

Ambiguous cases. Information about data, calculation methods, equations, plots or other research artifacts should NOT be considered as information about code unless explicitly defined as being associated with an element of code/software. The statement "Note that the sources for these examples can be found in the public GitHub repository [57]" mentions "sources for these examples" being provided in a public Github repository. Repository location alone is not sufficient evidence of software unless the contents are described as code/software. Therefore, no element of code/software can be identified and the section is classified as "incorrect". The negative examples of the first three hard negative pairs all explicitly specify an identifiable element of code/software (e.g. Mathematica, Python scripts) associated with the research artifact described (e.g. "Standard numerical methods", "Numerical analysis"). Since an element of code/software can be identified, these statements are not to be classified as part of the category "incorrect".

unclear

Definition. this category includes code availability statements stating ambiguous or unclear information about the availability of Author-Produced Software Artifacts

Exemplars. (1) code/software cannot be made available for reasons disclosed in the code availability statement. (2) inquiries about the code in this work will be responded to by the corresponding authors. (3) code/software will be made available on reasonable request. [author's comment: the software is not available since presently not all parts have been updated to modern computing.].

Hard positive. code/software will be made available on reasonable request. [authors' comment: code/-software sharing is not applicable to this paper.]. / inquiries about the code in this work will be responded to by the corresponding authors.

Hard negative. this manuscript has no associated code/software. [author's comment: all the information about use of code is given in manuscript.]. / this manuscript has no associated code/software. [author's comment: in this theoretical study, all analyses and derivations were performed using established theoretical frameworks and analytical methods.].

Inclusion. A code availability statement is included in this category when information on the access conditions of Author-Produced Software Artifacts is ambiguous or circular. Also, sections are included in this category when they make positive availability claims and later statements logically contradict them. For instance, a statement mentioning that code will be made available upon request and later stating that code sharing is not applicable should be classified as "unclear".

Exclusion. A code availability statement is excluded from this category when the content states information about code availability that clearly indicates the access conditions of identifiable Author-Produced Software Artifacts. Also, sections are NOT automatically included in this category if they make negative availability claims and later statements logically contradict them. In those cases, the negative availability claim can be overridden by later information on the access conditions of identifiable Author-Produced Software Artifact.

Ambiguous cases. Statements like "code/software will be made available on reasonable request. [author's comment: code/software sharing not applicable to this article as no code/software was generated or analysed during the current study.]" or "this manuscript has associated code/ software in a data repository. [authors' comment: code/software sharing not applicable to this article as no code/software was generated or analysed during the current study.]" are examples of statements that first make a positive availability claim and later CONTRADICT it by stating that code sharing is not applicable to the study at hand or that code is not available. The statement "code/software cannot be made available for reasons disclosed in the code availability statement." is an example of statement that ought to be considered circular. The statement states that information on the reasons for a certain access condition is given in the statement itself, without specifying anything more. The statement "this manuscript has no associated

code/software. [author's comment: all the information about use of code is given in manuscript.]. makes a negative availability claim that could potentially be overridden by further information on access conditions BUT "information about the use of code" cannot be considered clear information about availability of an Author-Produced Software Artifact. Therefore, the section is classified as "unclear". The statement "this article has no associated code available at the time of publication. The code used in this article will be made available online soon." also makes a negative availability claim that could potentially be overridden by further information on access conditions BUT information stating that APSA "WILL be made available soon" is not considered clear information about availability of APSAs. There is no clear timeline on when the APSA in question will be made available. Therefore, the section is classified as "unclear". The statement "my manuscript has associated code/ software in a data repository. [author's comment: the code used to generate results in this article will be made available in the next release of the pythia 8.3 code [https://pythia.org/]." is a similar case where a positive availability claim is made about an APSA, but then the use of the future tense makes the access conditions ambiguous. The statement "inquiries about the code in this work will be responded to by the corresponding authors." exclusively states information about the opportunity of giving further information about Author-Produced Software Artifacts. Even if it is information about Author-Produced Software Artifacts, it cannot be considered clear information about availability of APSAs. Therefore, the section is classified as "unclear".

A.3 CAS External Tool Availability Schema

no_tool

Definition. This category includes code availability statements that do not mention any External Software Tool.

Exemplars. (1) this manuscript has no associated code/software. [authors' comment: the code used in this project is publicly available at the following repository: <https://github.com/opengpd/gpddatabase> (see ref. [78]). (2) this manuscript has no associated code/software. [authors' comment: this study does not involve computational analysis, modeling, or simulation, so no custom code or software was generated or utilized.]. (3) this manuscript has no associated code/software.[authors'comment: all code analysed in this manuscript are available from the corresponding author on reasonable request.].

Hard positive. "this manuscript has no associated code/software. [authors' comment: code/software sharing for this article is not applicable because no software and codes were generated during the process. for colleagues who want to perform the same calculations, following the theoretical explanations which are provided throughout the text will be sufficient.]." / this manuscript has no associated code/software. [authors' comment: the code used in this project is publicly available at the following repository: <https://github.com/opengpd/gpddatabase> (see ref. [78]).

Hard negative. this manuscript has associated code/ software in a data repository. [authors' comment: the code and software presented in the current study are available in the pierre auger observatory's zenodo repository. <https://doi.org/10.5281/zenodo.4487612>]. / this manuscript has no associated code/ software. [author's comment: the code for this work is written in python and handles. It is available in source format from the corresponding author upon reasonable request for colleagues needing it for similar analyses.].

Inclusion. A code availability statement is included in this category if no External Software Tool is mentioned in the section. Sections that do not mention any Author-Produced Software Artifact are NOT included in the section UNLESS they also do not mention any EST being used in the study.

Exclusion. A code availability statement is excluded from this category if AT LEAST ONE External Software Tool is mentioned, regardless of access conditions. Sections are NOT excluded from this category if they mention at least one Author-Produced Software Artifact UNLESS they also do not mention any EST being used in the study.

Ambiguous cases. In a vast majority of exemplars provided, the only element of code/software mentioned is an Author-Produced Software Artifact (e.g. "code included [...]", "code used in this project", "code/software generated [...] during the current study"). Given that in all of these, no ESTs are mentioned, the sections are classified as "no_tool" All negative cases of the first, second and third hard negative pairs mention ESTs (i.e. "event generators", "python") or imply EST usage (e.g. "the *maple* code used"). Therefore, none of these sections is classified as "no_tool" along the "external_tool_availability" dimension.

open_tool

Definition. This category includes code availability statements stating that only open, free-to-use External Software Tools were used.

Exemplars. (1) "This manuscript has associated code/software in a data repository. [author's comment: the software for detector simulation and event reconstruction is publicly available on <https://github.com/muoncollidersoft>. Event generation is performed by using open source programs as described in the text, configuration files can be provided upon request. Final data analysis is performed with the root open source software, macros can be provided upon reasonable request.]." (2) code/software will be made available on reasonable request. [author's comment: the current study uses publicly available software relevant to high energy physics, and the analysis code is available from the corresponding author on reasonable request.]. (3) "code/software will be made available on reasonable request. authors' comment: a large fraction of the software used in the experiment is publicly available at the repository <https://doi.org/10.5281/zenodo.5574115> Specific software that was utilized to obtain the results shown in this paper can be made available upon request.]"

Hard positive. "This manuscript has no associated code/software. [authors' comment: we utilized publicly available codes in this study. While we developed additional code to efficiently interface with these public codes, we believe that this auxiliary code does not provide significant standalone value to warrant publication or public sharing.]" / my manuscript has associated code/ software in a data repository. [author's comment: the code used to generate results in this article will be made available in the next release of the pythia 8.3 code [<https://pythia.org/>].

Hard negative. "This manuscript has associated code/software in a data repository. [author's comment: the software for detector simulation and event reconstruction is publicly available on <https://github.com/muoncollidersoft>. Event generation is performed by using open source programs as described in the text, configuration files can be provided upon request. Final data analysis is performed with the root open source software, macros can be provided upon reasonable request.]" / this article has associated code in a code repository. The data on which this paper is based may be reproduced using the code and scripts found on github at <https://github.com/zgmou/sewedthimble>.

Inclusion. A code availability statement is included in this category if ALL identifiable External Software Tools are publicly available software tools. Sections that do not mention any identifiable ETS are NOT categorized as part of this category and are instead classified as "no_tool". Remember that access conditions of Author-Produced Software Artifacts are IRRELEVANT for classification along the dimension "external_tool_availability".

Exclusion. A code availability statement is excluded from this category when AT LEAST ONE identifiable External Software Tool is NOT a publicly available software tool OR when there are NO identifiable ETSs mentioned in the statement.

Ambiguous cases. The statement "code/software will be made available on reasonable request. authors' comment: a large fraction of the software used in the experiment is publicly available at the repository <https://doi.org/10.5281/zenodo.5574115> Specific software that was utilized to obtain the results shown in this paper can be made available upon request.]" mentions the External Software Tool "large fraction of the software used in the experiment". This element of software is characterized as an EST due to the generic phrasing and the absence of authorship/provision signals. Since this is the only identifiable EST and it is described as publicly available, the section is classified as "open_tool". In the statement "This manuscript has no associated code/software. [authors' comment: atlas collaboration software is open source, and all code necessary to recreate an analysis is publicly available. The athena (<http://gitlab.cern.ch/atlas/athena>) software repository provides all code needed for calibration and uncertainty application, with configuration files that are also publicly available via docker containers and cv mfs. The specific code and configurations written in support of this analysis are not public; however, these are internally preserved.]" the only element of software that's described as "internally preserved" is an Author-Produced Software Artifact. All other identifiable elements of software are External Software Tools. These are all described as publicly available. Therefore, the section is classified as "open_tool". The negative example of the first hard negative case mentions two ESTs. One of the ESTs mentioned is NOT publicly available (i.e. "mathematica"). Therefore, the section is NOT classified as "open_tool". The positive example mentions a variety of ESTs openly available (i.e. "software for detector simulation", "open source programs", "root open source program"), while only describing Author-Produced Software Artifacts (i.e. "configuration files", "macros") as available upon request. Since all identifiable ESTs are described as publicly available, the section is classified as "open_tool"

mixed_tool

Definition. This category includes code availability statements that mention a combination of publicly available tools, proprietary tools or tools whose access conditions are unknown.

Exemplars. (1) "this manuscript has no associated code/ software. [authors' comment: we employed mathematica and python for the numerical computations and graphical analyses presented in this study. no additional code or software was developed or utilised during the course of this research]." (2) this manuscript has no associated code/ software. [author's comment: we employed python and maple for numerical analysis] (3) this manuscript has associated software in a code repository. [author's comment: our code can be found on Github [10]. We employed mathematica and open source software flavio for numerical analysis]

Hard positive. this manuscript has no associated code/software [author's comment: we employed maple and python for numerical computations] / code/software will be made available upon request [author's comment: we used mathematica for numerical analysis. We also utilized publicly available software package GetDist]

Hard negative. this manuscript has associated software in a code repository. [author's comment: our code can be found on Github [10]. We employed mathematica and open source software FeynCalc for numerical analysis] / this manuscript has associated software in a code repository. [author's comment: Wolfram Mathematica and Maple code scripts were used for the analysis in this manuscript]

Inclusion. A code availability statement is included in this category when at least two External Software Tools mentioned in the section have DIFFERENT access conditions. Therefore, a statement is classified as "mixed_tool" along the "external_tool_availability" dimension if: - it mentions one or more PUBLICLY AVAILABLE tool and one or more PROPRIETARY tool; - it mentions one or more PUBLICLY AVAILABLE tool and one or more UNKNOWN tool; - it mentions one or more PROPRIETARY tool and one or more UNKNOWN tool; - it mentions one or more PUBLICLY AVAILABLE, one or more PROPRIETARY, and one or more UNKNOWN tools;

Exclusion. A code availability statement is excluded from this category when ALL External Software Tools mentioned have EQUAL access conditions. That is, a statement is excluded when all ETSs are PUBLICLY AVAILABLE, all are PROPRIETARY, or all are UNKNOWN.

Ambiguous cases. The positive examples of the first and second hard negative pairs both mention a mix of publicly available and proprietary tools. On the other hand, the negative example of the first pair mentions three publicly available tools, while the negative example of the second pair mentions two proprietary tools. Since in the negative example, all mentioned ETSs have the same access conditions, they are NOT classified as "mixed_tool" under the "external_tool_availability" dimension

proprietary_tool

Definition. This category includes code availability statements stating that only third-party, proprietary External Software Tools were used.

Exemplars. (1) this manuscript has no associated code/software. [author's comment: we used mathematica software for most computations. however, this software is not free.]. (2) the manuscript has no associated code/software. [author's comment: all the figures are generated by the mathematica program, which can be requested from the authors.]. (3) "My manuscript has associated code/software in a data repository. [author's comment: all analytical computations reported in this manuscript have been rechecked in the computer algebra system maple. Two maple sheets corresponding to the extreme and non extreme cases can be found in supplementary materials and in the repository below. The discretization of differential operators (23)–(25) and (41)–(43) using the tchebyshev type spectral method is equally performed in maple computer algebra system. Finally, the numerical solution of the resulting quadratic eigenvalue problem (48) is performed in matlab software using the polyeig function. All these materials are freely accessible at the following repository <https://github.com/dutykh/ellisbronnikov/>]." /

Hard positive. code/software will be made available on reasonable request. [author's comment: maple is used for calculation. No specific code is generated.]. / "This manuscript has associated code/software. [authors' comment: the code used for the event generation, the network architecture and its training can be found at https://github.com/amorandini/param_hunt. An example jupyter notebook which explains how to derive the plots and results in this paper is also provided. Wolfram Mathematica was used for some calculations in the paper]."

Hard negative. this manuscript has no associated code/software. [author's comment: all calculations were performed analytically. some results were verified using maple symbolic computation software.]. /

this manuscript has no associated code/software. [we used a computation software for numerical analysis in this manuscript]

Inclusion. A code availability statement is included in this category if ALL identifiable External Software Tools are third-party, proprietary software tools. Remember that access conditions of Author-Produced Software Artifacts are IRRELEVANT for classification along the dimension "external_tool_availability".

Exclusion. A code availability statement is excluded from this category when AT LEAST ONE identifiable External Software Tool is NOT a third-party, proprietary software tool. That is, when at least one ETS is an open, publicly available tool or its access conditions are UNKNOWN.

Ambiguous cases. The second statement of the first hard positive pair is included in the "proprietary_tool" category because the only External Software Tool mentioned (i.e. "Wolfram Mathematica") is a proprietary tool. All other elements of code/software mentioned are Author-Produced Software Artifacts. Therefore, their associated access conditions do not count for classification in the "external_tool_availability" dimension. This is also the case for the second statement of the second hard positive pair. All negative examples in the first, second and third hard negatives do NOT exclusively mention External Software Tools that are proprietary. Therefore, they are not classified as "proprietary_tool".

unknown_status_tool

Definition. this category includes code availability statements mentioning that only External Software Tools whose access conditions (i.e. open, proprietary) are unknown were used.

Exemplars. (1) this manuscript has no associated code/software. [author's comment: we used a computation software for numerical analysis in this manuscript] (2) this manuscript has associated code/software [author's comment: our code can be found at the following repository [14]. Software tools were used for some simulations carried out in the manuscript]

Hard positive. code/software will be made available upon request [author's comment: we used a software for event generations] / this manuscript has no associated code/software [author's comment: a numerical analysis software was used in this manuscript]

Inclusion. A code availability statement is included in this category if the access conditions of ALL External Software Tools used in the manuscript are UNKNOWN.

Exclusion. A code availability statement is excluded from this category if the access conditions of AT LEAST ONE External Software Tool is either PUBLICLY AVAILABLE or PROPRIETARY.

Ambiguous cases. Whenever generic terms (e.g. software tools) are used to describe an EST without any specification of their access conditions, the access conditions of those ESTs are to be considered unknown. If the access conditions of an EST cannot be reliably inferred from the content of the section, the list provided in the classification prompt, or previous knowledge you have of the access conditions of known ESTs, the access condition of that EST is to be considered unknown. The negative example of the third hard negative pair mentions two ESTs (i.e. "Python", "Mathematica") whose access conditions are KNOWN (i.e. open and proprietary, respectively). Therefore, even though the section mentions ESTs whose access conditions are unknown (i.e. "other softwares"), it is NOT classified as "unknown_status_tool" under the "external_tool_availability" dimension.

Appendix B

Registry Schema and Pipeline Metadata

This appendix summarizes the registry system used by the pipeline. The registry files provide structured metadata for source publications, pipeline outputs, stage-specific artifacts, embeddings, and classification outputs. The complete field-level schema is included in the released data documentation.

B.1 Registry Overview

Registry	Main identifier	Purpose
Document registry	<code>doc_doi</code>	Stores publication-level metadata such as DOI, title, journal, publication year, source URLs, document type, and DAS/CAS presence.
Base output registry	<code>output_sha</code>	Stores metadata common to all pipeline outputs, including output type, software version, pipeline version, creation date, DOI, and dependencies.
Raw file registry	<code>output_sha</code>	Stores metadata about raw and converted files, including PDFs, XMLs, and full-text files.
Extraction registry	<code>output_sha</code>	Stores metadata about extracted and normalized DAS/-CAS sections, including section type, processing stage, file path, and text.
Embedding registry	<code>output_sha</code>	Stores metadata about embedding outputs, including file path and embedding model.
DAS classification registry	<code>output_sha</code>	Stores classification labels, confidence values, method, model, and classified text for DAS sections.
CAS classification registry	<code>output_sha</code>	Stores MCA and ETA labels, confidence values, method, model, and classified text for CAS sections.

B.2 Provenance Links

The registry system links artifacts through two main identifiers. Publication records are identified by `doc_doi`, which connects pipeline artifacts back to their source publication. Pipeline outputs are identified by `output_sha`, a hexadecimal hash used as the primary artifact identifier. The `dependencies` field in the base output registry stores the upstream artifact hash used to

produce a given output, making it possible to reconstruct the processing chain from source files to derived outputs.

B.3 Archive Paths

File paths stored in the registries are project-relative paths. In the released data bundle, archives preserve this layout under directories such as:

```
data/metadata/registries/  
data/metadata/manifests/  
data/records/software_version/  
data/descriptive_statistics/  
data/images/
```

This allows users to restore the local data workspace by extracting the release archives from the root of the matching code checkout or release workspace.

Appendix C

Additional Result Figures

This appendix reports additional final-run plots corresponding to descriptive results discussed in the main body but not displayed there as figures.

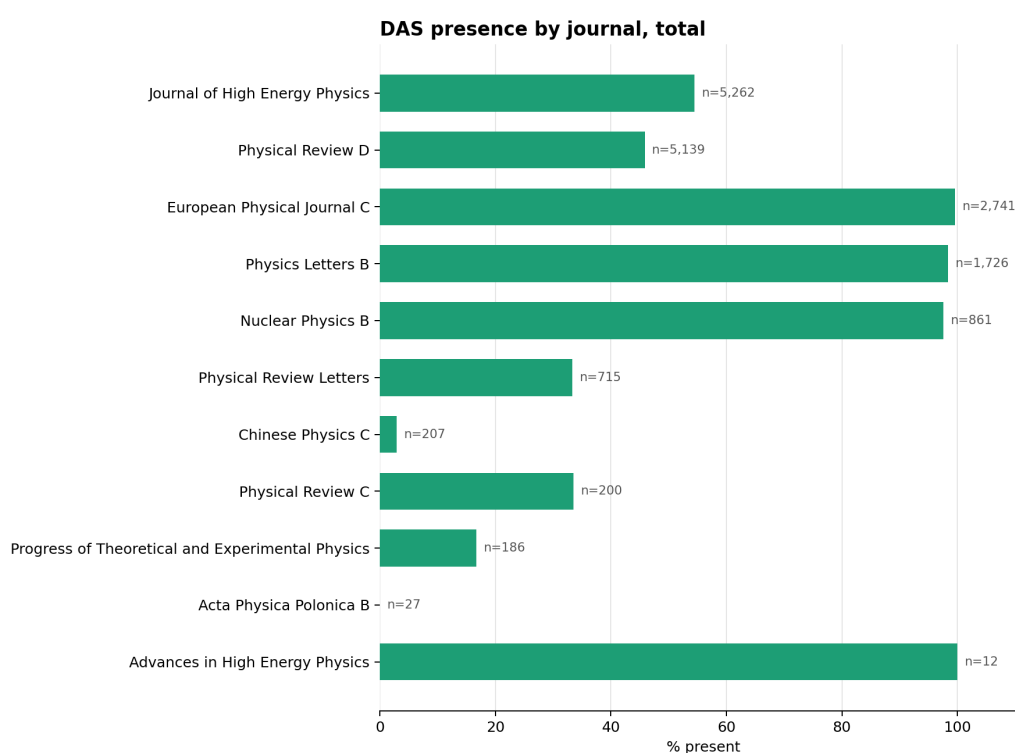


Figure 8: DAS presence by journal in the total sample of scientific studies.

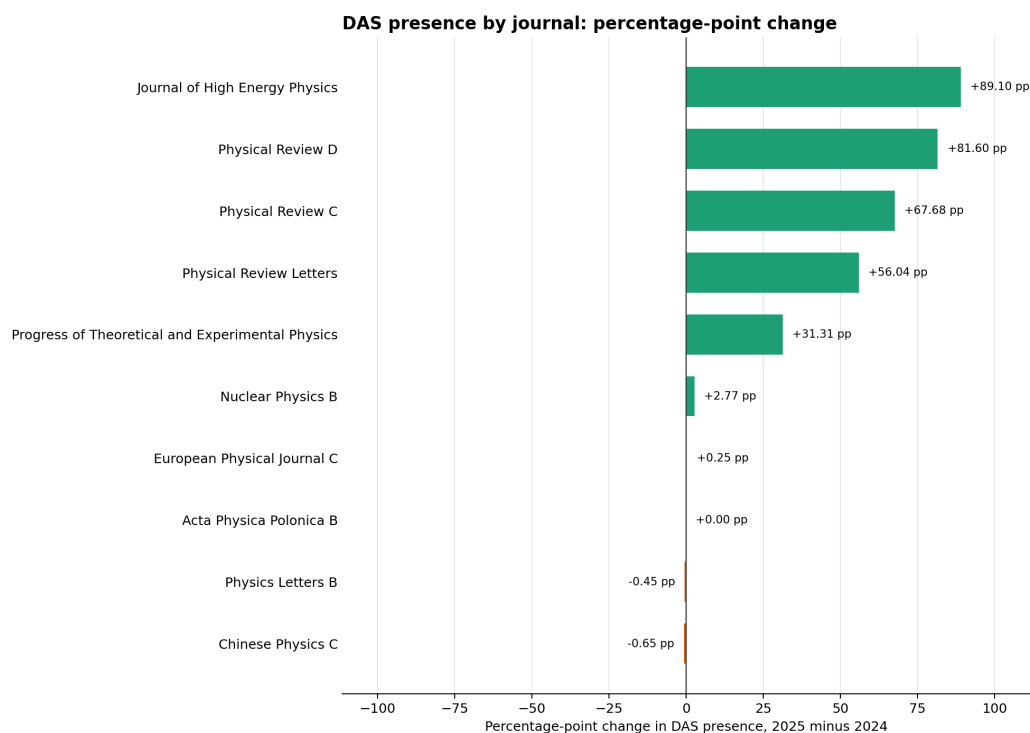


Figure 9: Percentage point change in DAS presence by journal between 2024 and 2025.

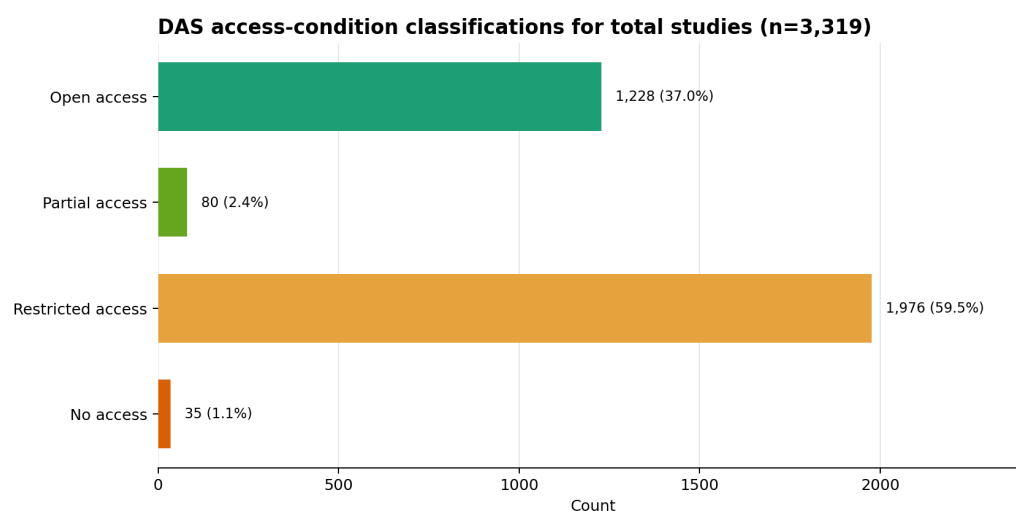


Figure 10: DAS access-condition distribution among studies with associated data and clearly reported access conditions.

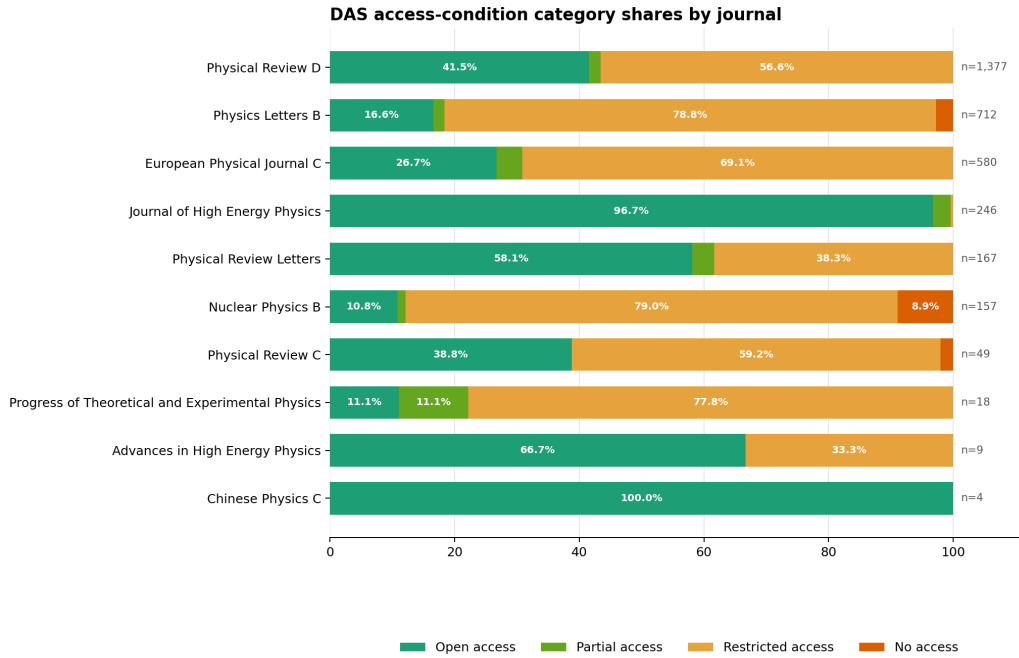


Figure 11: DAS access-condition category shares by journal among studies with associated data and clearly reported access conditions.

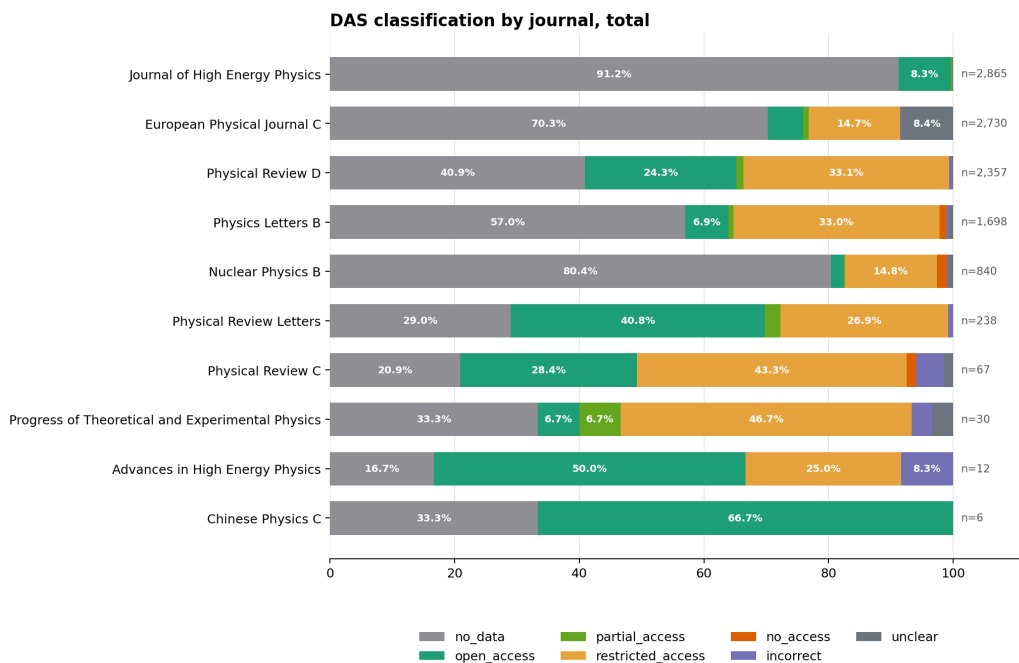


Figure 12: DAS classification counts by journal in the total classified sample.

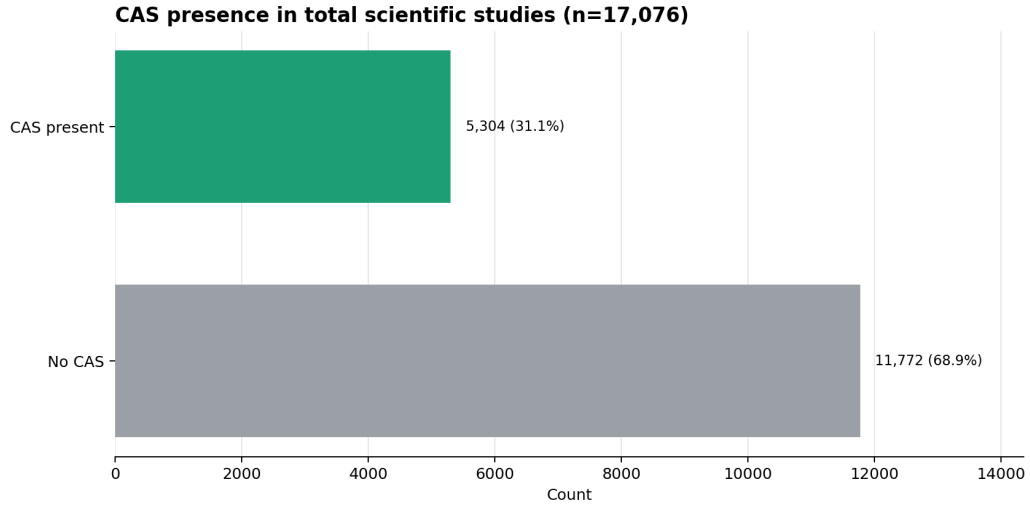


Figure 13: CAS presence in the total sample of scientific studies.

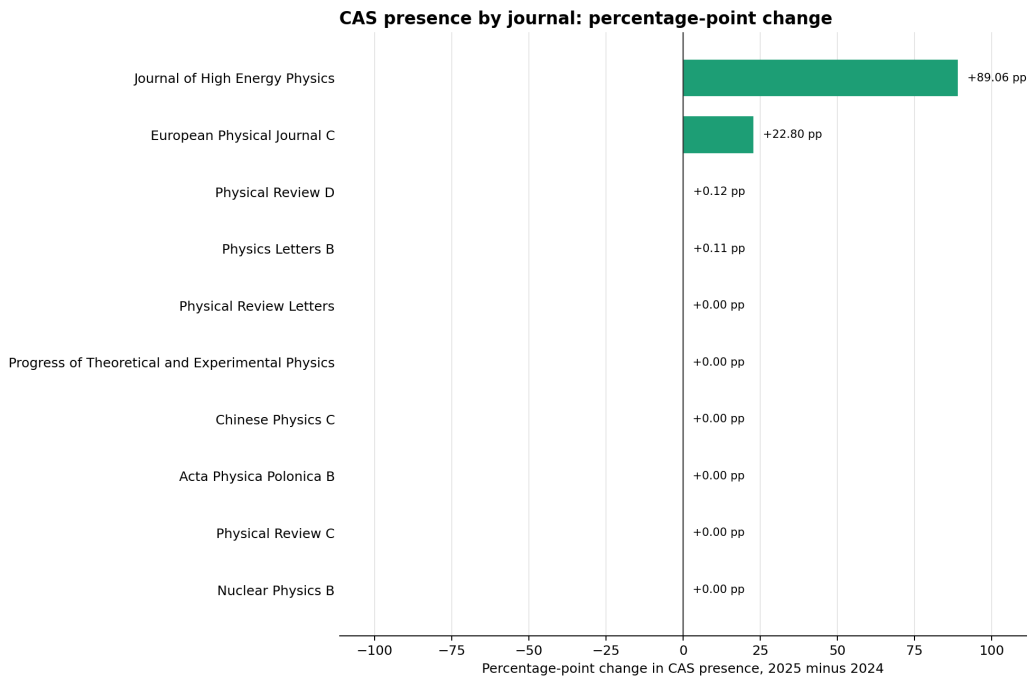


Figure 14: Percentage point change in CAS presence by journal between 2024 and 2025.

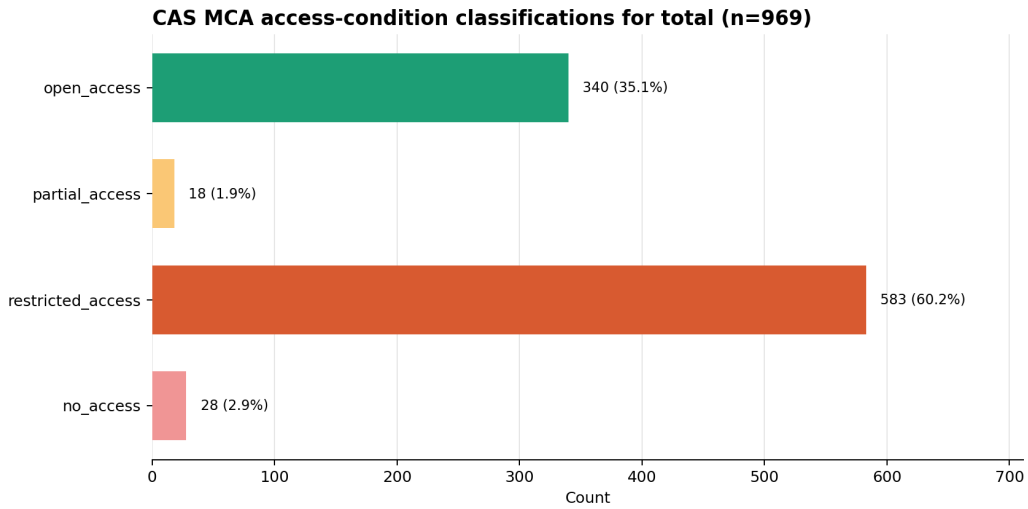


Figure 15: MCA access-condition distribution among studies with associated author-produced code and clearly reported access conditions.

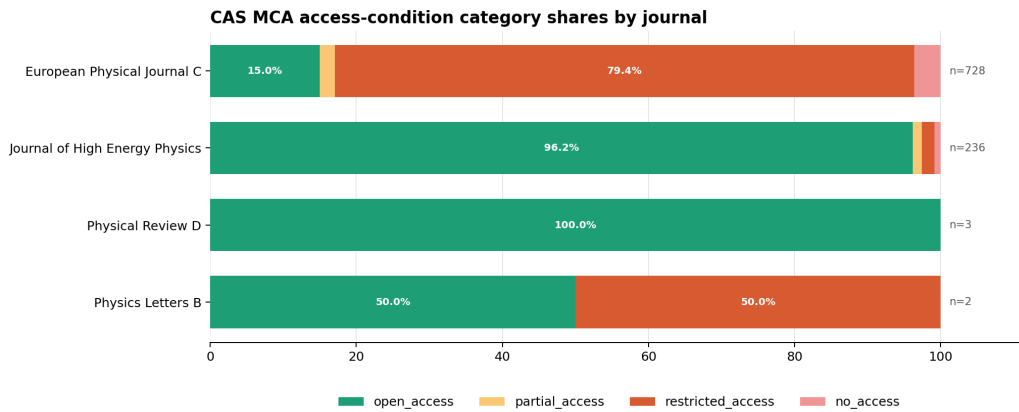


Figure 16: MCA access-condition category shares by journal among studies with associated author-produced code and clearly reported access conditions.

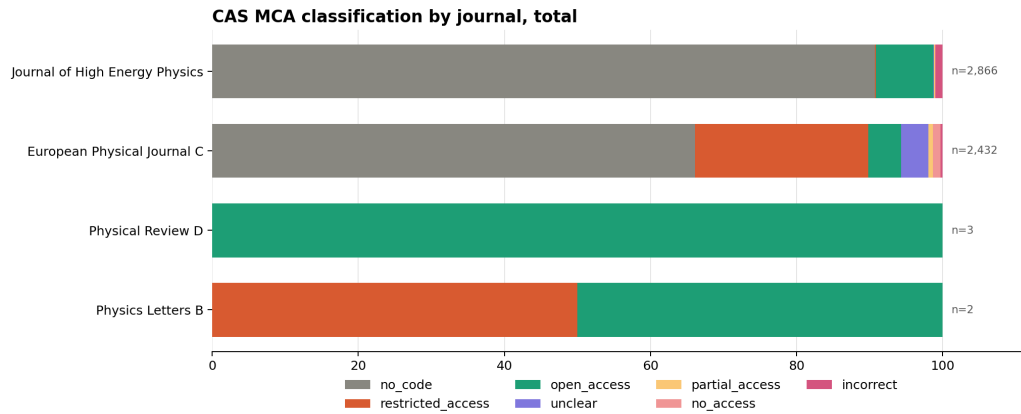


Figure 17: MCA classification counts by journal in the total classified CAS sample.

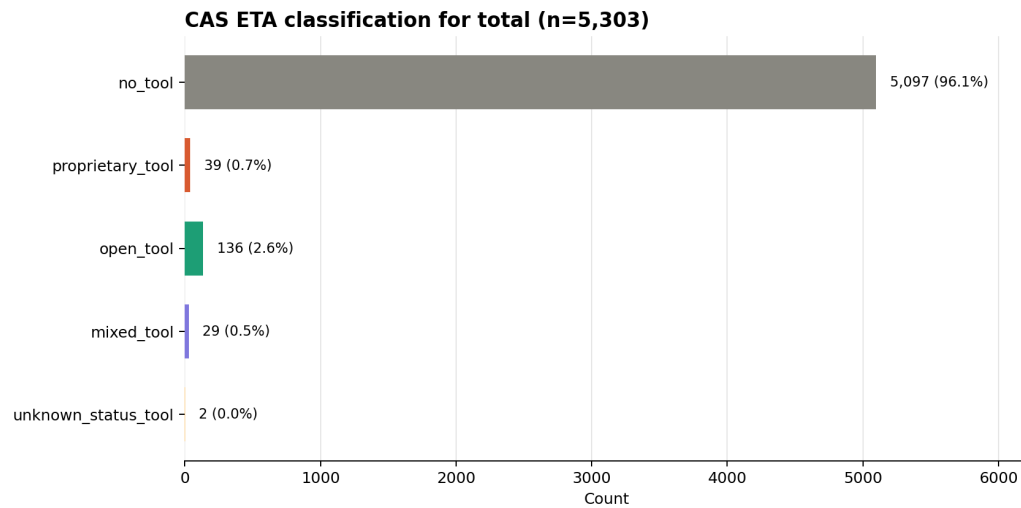


Figure 18: ETA classification distribution in the total classified CAS sample.

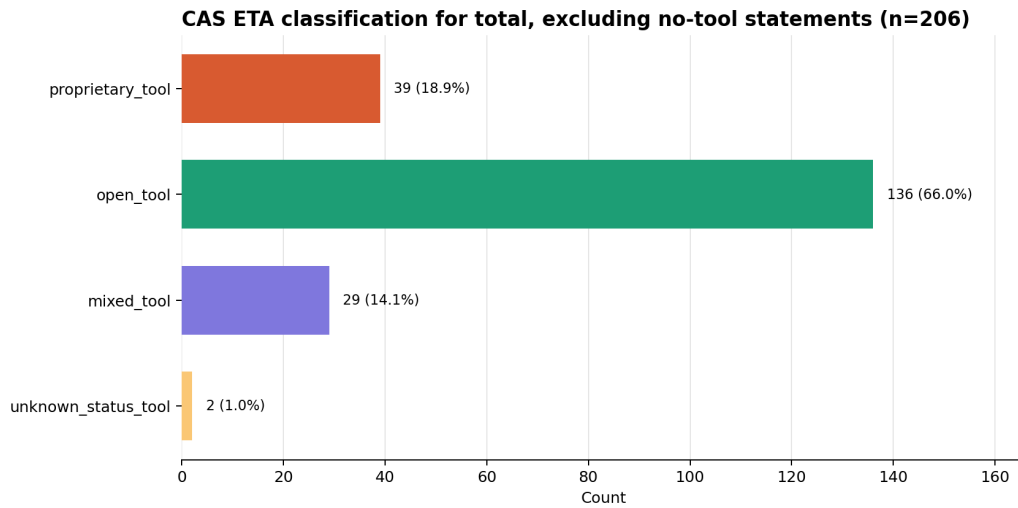


Figure 19: ETA classification distribution among CAS sections mentioning external software tools.

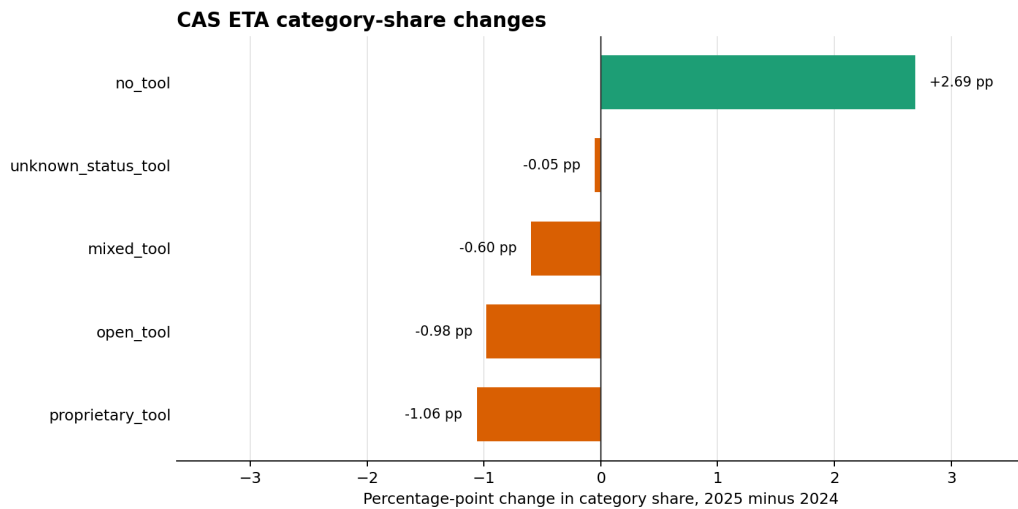


Figure 20: Percentage point change in ETA classifications between 2024 and 2025.

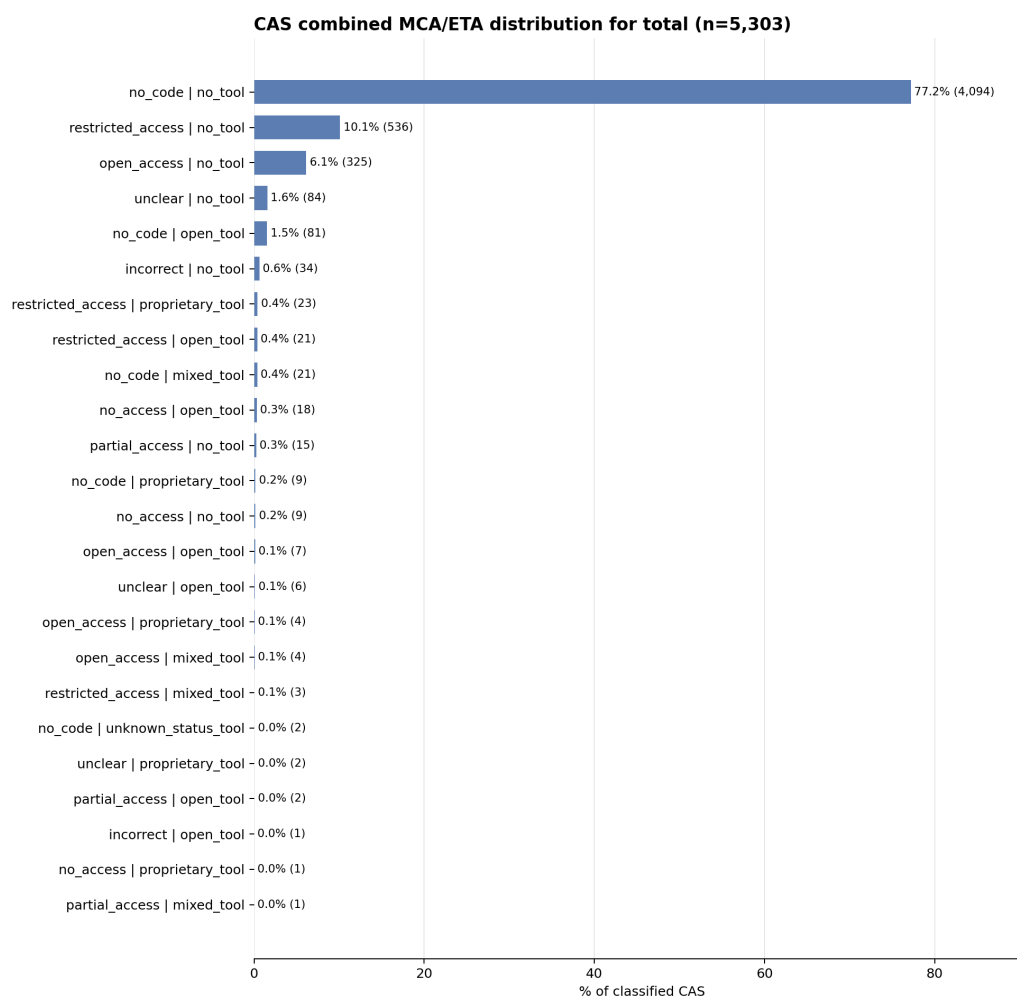


Figure 21: Combined MCA-ETA category distribution in the total classified CAS sample.

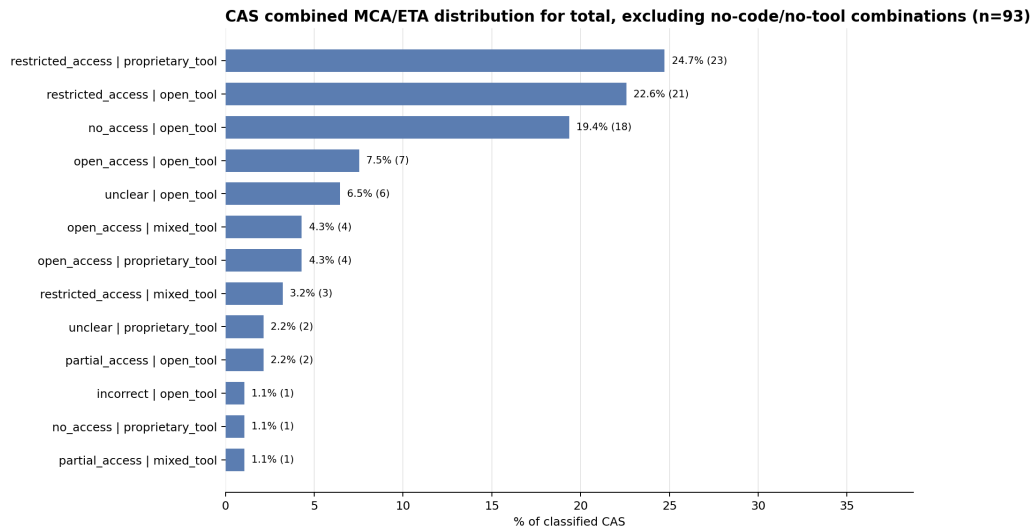


Figure 22: Combined MCA-ETA category distribution excluding sections classified as no_code/no_tool.

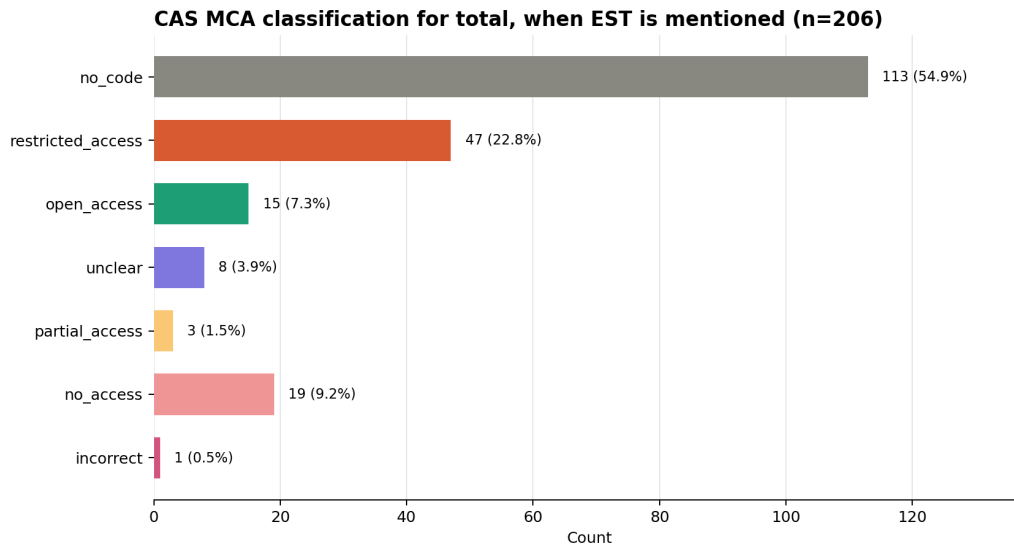


Figure 23: MCA classification distribution among CAS sections mentioning external software tools.