SIMONS OBSERVATORY

DIGITAL ASSETS AND AI GOVERNANCE POLICY

REVISION HISTORY

Version Letter	Revision Date	Author: Notes
A	2026-05-15	First version released to Co-Dir's and SP for comments
B	2026-06-22	Released to the SO Collaboration for comments
C	2026-08-06	Version approved for public release

REFERENCED & APPLICABLE DOCUMENTS

The requirements in the following documents apply, but this document supersedes if there is a conflict.

Reference used within this doc	Version	Title & Description, including SO Document Key if applicable	Notes, relevant part of document
[1]		SO Code of Conduct	
[2]		NSF PAPPG	

Table of Contents

1. Rationale and Scope

The Simons Observatory (SO) generates substantial intellectual property in digital form, including software, raw and processed data, derived data products, and associated documentation, collectively referred to as Digital Assets. Clear policies are necessary to ensure:

- the confidentiality of information and digital assets prior to public release;
- accountability, transparency, and traceability in collaborative work;
- the appropriate and secure use of digital assets without compromising observatory operations or cybersecurity; and
- compliance with funding agency requirements (also see [2] for NSF guidelines) and the SO Code of Conduct [1]. These policies are not intended to limit the creativity or scientific independence of SO Collaborators. Rather, they establish a governance framework that enables innovation and reliable operational practices to coexist.

This document defines policies governing three specific areas:

- The responsible use of artificial intelligence (AI) tools in research and collaboration activities conducted under the auspices of SO;
- The copyright, licensing, and stewardship of software developed using SO funds and/or deemed mission-critical for observatory operations and the delivery of core SO science, including software developed within Data Management (DM) and the Analysis Working Groups (AWGs); and
- The licensing, versioning, and unique identification of data products generated through the reduction, simulation, or analysis of SO data, including derived AWG products intended for public release.

These policies may inform, but do not directly regulate, activities such as software documentation practices (addressed in the SO Developer Guide), collaboration-related matters or procedures, unless specifically requested by the relevant panels (see, for e.g. use of AI for SO Publications), or the use of collaboration infrastructure and digital platforms such as Confluence and Slack.

This policy has been developed by the Digital Assets and AI Governance Committee, established by the OEO and composed of:

- Simone Aiola (DM L2, DM Coordination Committee Chair, OEO Representative)
- Adrian Bayer (Domain Expert, TAC representative)
- Josh Borrow (Domain Expert, DM L3, AI Policy first developer)
- Matthew Hasselfield (Domain Expert, DM Software Architect, SO Digital Resources Administrator, DM L3, Software Licensing Policy first developer)
- Brian Koopman (Domain Expert, Chief Information Security Officer)
- Andrea Zonca (Domain Expert, DM L4, Data Licensing Policy first developer)

The committee will present the initial policy to the collaboration following approval by the SO Co-directors and the Spokesperson. Future revisions will be developed in accordance with SO governance procedures.

2. Artificial Intelligence (AI) Usage Policy

2.1 Scope

This policy covers contributions made to Simons Observatory software tools, documents, procedures, and deployments. This covers things like software hosted under https://github.com/simonsobs and computing infrastructure. This document only covers the usage of 'external' AI tooling like ChatGPT, and not the training of AI and Machine Learning tools using SO data and resources (for example, using a machine learning model to predict detector bias, etc.); for those products, please see the SO policy on data releases (Section 4). Because of the rapid evolution of this technology, we (the Digital Assets & AI Governance Committee and SO Collaboration, under the direction of the SO Directors and Spokesperson) reserve the right to modify this policy at any time, with formal reviews every 6 months as this rapidly changing field evolves.

2.2 Broad Overview

As we enter the age of 'Agentic' Artificial Intelligence (AI), it has become clear that we require a policy regarding the usage of this new technology. These tools can significantly improve our productivity, make our lives easier, and allow a wider range of SO members to contribute to software. They also pose significant security risks, have the potential to yield unverifiable scientific results, and can increase maintainer burden. *Our policy, distilled into one sentence, is that the use of these tools is allowed in a supervised manner, but unsupervised use is explicitly prohibited. SO members are entirely responsible for the output and implications of any AI tool they choose to use and should act accordingly.*

2.3 Defining AI

A strict, stable definition of 'AI' is difficult to come up with. For this document, we'll focus on 'generative AI' as it pertains to text creation. This includes tools like ChatGPT, Apple Intelligence, Claude Code, and others, which are currently backed by 'Large Language Models' or LLMs. These tools take text snippets as input, and produce text as output, in either structured or unstructured formats. They are often accompanied by either a web interface (e.g. ChatGPT) or a so-called 'harness' that users run locally (e.g. Claude Code or Codex).

2.4 Why This Policy?

As a scientific experiment, our correctness expectations are significantly higher than most consumer software. In addition, much of our work represents novel problem-solving with code, necessitating the development of new algorithms. A hidden bug or assumption deep within our stack could lead to significant errors in our reported results.

Most 'AI' models excel at solving 'embarrassingly replicated' tasks in open source software: writing plotting scripts, reducing data, or writing HTML/CSS for webpages. Though they certainly have promise for our more novel problems, they must be supervised carefully.

AI agents and tools can produce tens of thousands of lines of code or text a day with ease. This has the potential to represent a significant review burden on maintainers (please see the SO developer's guide for information on how we store and use code as part of SO projects). At the same time, if we can leverage these tools to more effectively and efficiently accomplish our scientific goals, without overburdening core contributors, we may wish to embrace them.

This policy attempts to strike a balance between security and correctness considerations, maintainer burden, and empowering SO members to explore this new and helpful technology.

2.5 Agentic Tools

Agentic tools are broadly those that can function autonomously and have access to an 'environment'. This contrasts with AI tools like ChatGPT, which primarily operate in a call-and-response model via a chat box.

Examples of Agentic tools are Cursor, GitHub Copilot, and Claude Code. These tools take the information about a request, along with that environment (e.g., "implement feature X" along with an existing code repository), to 'loop' until completion of that task. They can also represent 'Bots' that have access to logs or a document to continually raise alerts or improve content (e.g., GitHub's Copilot Review feature).

A recent development is the creation of 'fully agentic' tools like 'OpenClaw'. These programs, as of writing, are characterized by a file providing general instructions to the model along with its 'motivations'. The agent then functions entirely autonomously, looping forever, with full access to its host machine, including full internet access.

2.6 General Policies

- AI tools should never be used to produce results for scientific publications without human oversight. They may write code or documentation, which is then reviewed by a human, to reduce data, but direct calculations within the model are prohibited (e.g., uploading a CSV file and asking the system for the mean value).
- Code and text generated by AI must be described as being so, with the tool used acknowledged. For example, in a pull request where AI assistance was used, you may write "AI Usage: The content of this pull request was generated by Claude Code". It is helpful to also include information on exactly how the AI was used in this process, e.g., "The AI developed the code in process_fits.py, and I integrated it into main_system.py".
- The use of AI systems on the SO Site Computing (SC) is entirely prohibited. This includes, but is not limited to, running AI agents and agentic tools on SC hardware; interacting with SC over network connections with such tools running on a separate machine; and running such tools on computers directly connected to the site network (e.g., laptops at the site).
- All content must be assigned to a responsible human. Commits and other artifacts must originate from a human, and they are ultimately entirely responsible for their content. If a reviewer or telecon participant asks a contributor how a specific block of code works or how a figure was generated, the contributor must be able to explain it clearly. "The AI wrote it, and it passes the tests" is not an acceptable response.
- Fully agentic systems, which are capable of fully controlling the environment (e.g., 'OpenClaw' Agents) are completely prohibited from contributing to Simons Observatory work (see also the "Unacceptable Use" section). We welcome the use of systems such as Claude Code and Codex.
- Any results and impacts from the use of agents are the responsibility of the member who initiated the agent and is its operator. This includes, but is not limited to, impacts from: logic errors in code; shell commands executed by agents or under instruction from AI tools; and 'prompt injection' attacks where malicious information is hidden in webpages or other content that over-ride AI instructions.
- When using agentic tools to produce code or text, be mindful that their outputs may incorporate copyrighted or otherwise protected material. Publicly releasing such outputs—even under an appropriate open-source license—could still infringe copyright or give rise to plagiarism concerns, in direct violation of the SO Code of Conduct [1].
- The SO collaboration expects authentic engagement from contributors in all settings, be this pull requests, discussions in Slack or email, or presentation of results. The expectation is that the primary contributor is responsible for leading the conversation and response, potentially using AI tools if warranted, but not delegating responses to questions and comments entirely to such tools.
- Members must not paste unpublished SO scientific data, sensitive infrastructure configurations, passwords, API keys, or any other proprietary or unpublished information into external AI models (e.g., ChatGPT web interfaces), as this data may be retained or used for model training by the provider. Certain subscriptions may prohibit retention for privacy (e.g., university subscriptions), but care should still be used in these scenarios, especially with security-critical data and applications.

2.7 Acceptable Use

Our general policy for acceptable use is that these tools are generally allowed if and only if all outputs are reviewed by a human before submission. This allows:

- The use of tools like ChatGPT or Grammarly for copyediting and translation, for example, to translate information from your native language to English.
- Specifically, when preparing SO manuscripts, please note that to ensure human oversight, if AI tools are used, they can only be involved in the initial drafting before the manuscript begins the formal review process.
- The use of tools like Claude Code for programming, if and only if the outputs are fully reviewed (i.e., the code is read and understood by the human operator).
- Code submitted as pull requests should indicate it was created using AI assistance, and must have been tested by the human operator (e.g., using automated tests or manual verification) before submission.

2.8 Unacceptable Use

Our general policy for unacceptable use is to bar the use of fully autonomous tools (e.g. OpenClaw) to contribute to work for the Simons Observatory. Any and all outputs from AI models should be viewed and understood by the human supervisor before code or text is submitted. Generative AI tools should never be used to:

- Function entirely autonomously, when given wide-ranging access to a full personal machine (e.g., laptop, desktop computer with associated keys and passwords) without sandboxing and without a 'human in the loop' (e.g., the use of 'Ralph Wiggum' strategies or tools like 'OpenClaw' are prohibited).
- Solely develop responses to comments and questions from other SO contributors. For example, copying a comment into ChatGPT and copy-pasting its response directly back to the commenter.
- Autonomously submit artifacts that require human review. This includes pull requests, documents, and e-mails.
- Autonomously deploy and use software that involves the use of secret keys (e.g., API keys) that the agent has direct access to. It is never acceptable to share such secrets with these models. When required, users should thoroughly familiarize themselves with the latest security practices in this domain and discuss with the service maintainer before doing so.
- Develop untested and unverified code that is used to produce scientific results. 'Vibe-coding' analysis scripts and running them without understanding the code or using an automated testing strategy would be an example of a prohibited activity.

2.9 Policy Violations and Exceptions

Members who believe that AI agents are being used in a manner inconsistent with this policy are encouraged to contact the Spokesperson, the SO Data Manager, or a member of the Digital Assets and AI Governance Committee.

Exceptions to this policy may be granted. Requests for an exception should be submitted to a member of the Digital Assets and AI Governance Committee for review. Final approval must then be obtained from the Project Office, the Spokesperson, or both, depending on the scope and subject matter of the request. Documentation about such exceptions will be made available to the collaboration.

3. Software Copyright and Licensing Policy

Because of the complex structure of private entities, academic institutions, and individuals working on SO code, the collaboration will benefit from copyright and licensing practices that place minimal restrictions on the use and re-use of code. Such an Open Source approach has been adopted by a huge range of scientific and non-scientific projects.
There is a complex interplay among copyright, licensing, and exposure (whether code is public or private), and this policy recommends approaches to help protect the rights of code contributors and meet the observatory's needs.
The discussions and recommendations in the sections below are intended to apply broadly to code that is important for the timely success of SO, also referred to as mission-critical. The Data Manager (DM), the TAC (Theory and Analysis Committee) Chair, and the Spokesperson are the ultimate authorities for enforcing these standards as they deem appropriate and for resolving disputes involving code hosted on shared collaboration resources (such as the SO GitHub organization).
At the very end of this section, you will find instructions for how to record copyright and license information in your codebase.

3.1 Policy on Openly Contributed Code

We will use the designation "Openly Contributed Code" to describe repositories that meet a certain desired standard for being unencumbered by restrictive licenses. Such a designation is desirable to SO because open licensing is an assurance that access to the code will not be revoked at any point in the future. The designation should also provide reassurance to individual contributors, as the open license gives them access to their own contributions and to those made by others in the repository.
The following points describe how we will work with Openly Contributed Code in SO:

- We define "Openly Contributed Code" to be source code that is created for or made available to the Simons Observatory through a simple Open Source license, such as the BSD or MIT License, and that is stored (even if only as a fork) on the SO GitHub organization.
- When first creating a new repository hosting Openly Contributed Code, the developer must make sure to include a license file that states the terms of the license. When such a license file does not exist, care should be taken to confirm that the copyright holders (i.e., the contributors to the repo) are all on board with releasing their code under this license.
- When someone contributes code to an Openly Contributed Code repository, they are agreeing that their contributions may be used by others under the terms of the license in the repository. Note that they (or their employers) may retain the copyright to their contributions.
- In some cases, an Openly Contributed Code repository may be in the "private" state on GitHub. The code in private repositories should be considered as proprietary to SO, notwithstanding the Open License stated therein, and may not be treated as public code nor freely shared outside of the collaboration. Decisions on when and how to take codes from the private state to the public state will be made by the DM and TAC. In other words, each contributor to a private repository does not have the authority to make any of their (or others') contributions public; rather, each contributor acknowledges that the decision to make the code public lies with the DM/TAC. The responsible authorities will endeavor to respect the wishes of the authors while protecting the mission of the Observatory.

3.2 Importance of Licensing and Copyright

Mission-critical SO source code must be clearly licensed to avoid loss of access for the duration of the project. SO must have sufficient rights to use, modify, and release such code without significant burden (such as licensing fees or inappropriate attribution requirements). This condition is likely to be satisfied, provided that any of the following is true:

- SO controls the copyright of the code.
- The code is openly licensed.

- The code is not openly licensed, but a specific license has been obtained that will grant the necessary rights over the necessary time period.

In contrast to "Openly Contributed Code", we defined "Specially Contributed Code" to be source code that is licensed to SO through some mechanism other than a full Open Source license. We do not expect this to happen very often, but if it does, it will be good to have a term for it.

3.3 Copyright

The person or entity that holds the "copyright" of a work may determine how the work may be used by other entities. It is not necessary for some SO legal entity (such as the OEO or the Foundation) to hold the copyright of all critical code. Rather, it is only necessary that collaboration members have sufficient license to use, modify, and release the code. For simplicity, we recommend that each contributor (or, in some cases, their employer) retain the copyright to their contributions as collaborators of the Simons Observatory, and instead grant the Simons Observatory Collaboration continuing use of those contributions through an open license.
In the context of source code, the copyright holder has the sole right to license the code for use by others; the importance of the license is discussed in the next section.

3.4 Licensing

It is SO expectations that the copyright holder of a work will grant licenses to others so they can use, modify, or redistribute the work under certain conditions. Exceptions to such expectations should be discussed with DM/TAC.
The Open Source approach is a simple way to ensure that individual contributors get credit for and maintain access to their contributed source code, while at the same time preventing the SO codebase from evolving into a complex, risk-laden system of closed, licensed software. Open Source ensures that we will all have access to the contributed code, forever.
Note that the absence of a copyright notice and license does not imply that a work is in the public domain. The best way to make code freely usable is by explicitly licensing the work, for certain uses, in the distributed source code text.
For general information about the principles of Open Source software, see https://opensource.org/osd and the other pages at that site.
We recommend that Openly Contributed Code be licensed under the BSD 2-Clause License. Similar licenses, such as the MIT License, are also acceptable. These licenses, in essence, allow for the free modification and use of the source code, but require any public release of the modified source code to include the original copyright, license, and disclaimer notices. We also emphasize that open-source licenses may appropriately be applied to content developed with the assistance of AI. However, developers should ensure that their use of AI complies with the guidelines outlined in Section 2.

3.5 Public vs. Private Repositories

When code resides in a "private" repository on the Simons Observatory GitHub organization, it is considered to be in a protected, proprietary state, even though it may contain licensing text that suggests it is Open Source software. SO members must not make such code public, nor distribute it to members outside the collaboration, except as directed by the appropriate authorities (DM/TAC).
Some SO repositories will be private, initially, and become public later. The transition to public should occur only once all the following conditions have been met:

- The copyright and licensing information is clear and accurate.
- The code has been reviewed to ensure that no software licenses would be violated by such publication (i.e., ensure that all license requirements for imported codes have been met).
- The primary developers of the code have been consulted.
- The Data Manager and/or the TAC Chair (depending on the code purview) agree to the change.

3.6 Acknowledgment Requirements

In some cases, developers within SO may want to impose additional restrictions or requirements on the use of code they have produced. For example, it is not uncommon to see a statement of the form "If results from this code are used in a scientific publication, then some text must be included in the acknowledgments, and some paper must be cited."
Some contributors may object to having their work encumbered by a stringent citation requirement. So such requirements should not be applied without consensus among the contributing developers. The DM/TAC should be consulted if consensus cannot be reached; otherwise should still be informed.

3.7 External Collaboration Opportunities

Near the top of every source code file, state the copyright and mention the license. This should look something like:

> Copyright (C) 2026 Members of the Simons Observatory collaboration.
> Please refer to the LICENSE file in the root of this repository.

If you later make edits to this file, update the copyright year to include the date of all changes. For example, the date may ultimately be something like "2018-2020, 2022." It is not necessary or helpful to update the date string unless actual changes are made.
Note that we point the reader elsewhere to learn about the license conditions. Although one often encounters codes where a complete copy of the license is stated at the top of every source file, this is awkward and not legally necessary, so let's not do it unless we have code where individual files are likely to be extracted and somehow misused.
The vagueness of "Members of the ... collaboration" would probably be a concern if the code were not openly licensed. The details of exactly who owns the copyright to each line of code could probably be figured out, but it's largely irrelevant since we're making all those lines freely available.

At the top level of your repository, clearly state the license conditions. The best way to do this is to have a text file called LICENSE, that either contains the license (e.g., the text from the BSD 2-Clause License) or contains instructions on how to find the licensing information (as is done, for example, here).

If you have introduced source code from another project into an SO repository, please endeavor to respect the licensing conditions of that code. This means:

- Make sure that you understand and accept the conditions of the license. Some licenses may constrain the ways in which you can distribute modified code (e.g. SOFA forbids the distribution of modified routines unless you also change the names of the routines). Even some "Open Source" licenses, such as GPL, have strict requirements on how derivative code can be distributed.
- If the imported code requires you to preserve the license/copyright/disclaimer, take care to do so. It may be necessary to copy some license information into your project's LICENSE file, perhaps under a heading such as "This repository contains code by developer, which is distributed under the following license and conditions: ...".
- If you modify imported code, make a clear note that you have done so, near the top of the source code file.

4. Data Licensing and Data Attribution Policy

Public SO data products should be released under an explicit license. An explicit license is important for both legal clarity and for adherence to reuse standards such as the Findable, Accessible, Interoperable, and Reusable (FAIR) principles. It should make clear what downstream users are allowed to do and how they should credit the collaboration.
In addition, Public data releases need a citation model that is simple, stable, and useful both to people and to infrastructure. The question is not only how to give credit, but how to make the released data legible as a distinct research object that can be cited, tracked, and linked to its metadata over time.

4.1 Data Licensing

SO Data products will be released under the CC BY 4.0 license. This is the strongest general default because it preserves attribution while minimizing unnecessary barriers to reuse. It allows redistribution, adaptation, and broad downstream use, while still requiring that the SO Collaboration receive credit.
This proposal is specifically about public SO data products, not collaboration papers. Paper licensing, including what license to use for arXiv-posted manuscripts when the collaboration has discretion, should be handled in a separate publication policy.

4.2 Why CC BY 4.0

An unlicensed dataset is effectively proprietary, with reuse rights left unclear. That is the opposite of what we want for public-facing SO data products.
CC BY 4.0 provides a simple and widely understood framework:

- users may share and adapt the data (see list below for more details);
- attribution to the SO Collaboration is required;
- commercial use is not blocked;
- derived works do not inherit additional compatibility constraints

This makes CC BY 4.0 a practical balance between openness and credit. In practice, for users of SO data products, this reinforces typical academic conduct, and it means that:

- People can use SO data in publications as-is, for instance, reproducing a line (e.g., power spectrum) for comparison to their own data, but SO must be acknowledged when doing so.
- People can re-analyze SO data and write publications, for example, using a map and their custom power spectrum code to produce their own power spectra, but SO must be acknowledged when doing so and indicate what analysis took place.
- People can share their re-analyzed data as they wish, as long as they acknowledge that it is based upon a SO dataset. For example, someone can generate a cluster catalog list and share it under whatever license they choose, but when doing so, they must link back to our original SO dataset (e.g., the map they extracted the clusters from).

We intend to share these examples alongside any SO data release.

4.3 Other License Options

Creative Commons licenses are often written as CC {BY}-{NC}-{SA} 4.0:

- BY requires attribution
- NC restricts use to non-commercial purposes
- SA requires derivative works to be distributed under the same license

The NC and SA clauses may be useful in some cases, but they should not be the default.

NC can create ambiguity around what qualifies as commercial use, which can discourage downstream reuse in academic, educational, industry, and software-adjacent settings. SA can create compatibility obligations for derivative works that make redistribution and integration harder. Both therefore, have implications that may hinder reuse and require careful consideration before adoption, and thus require approval from SO Spokesperson and OEO.

4.4 Data Attribution

In this policy, we discuss the use of DOI for data attribution. Each official release, such as DR1, DR2, and so on, should receive a single DOI. When a new release is published, a new DOI should be minted, and its authorship should reflect the full collaboration membership at the time of that release, including those with builder status.
This is the best balance between simplicity and usefulness. It is only slightly more complex than citing the release paper alone, but it gives the data release its own persistent identifier and avoids the operational burden and citation fragmentation of more granular DOI schemes.

4.5 Why a DOI is needed

Traditionally, data are often cited through the corresponding release paper. That works reasonably well for academic credit, but it has important limitations.
A release paper is primarily a scholarly narrative and methods artifact. A DOI for the data release serves a different purpose: it identifies the released data as a citable object in its own right. That distinction matters because a DOI can point directly to the release landing page, carry structured metadata, and separate citation of the data product from citation of the methods or release paper.
Relying only on the paper makes several things harder:

- treating the data release itself as a distinct citable object;
- attaching structured metadata directly to the release;
- keeping data citation separate from paper citation;
- maintaining a clean historical record as releases evolve over time.

4.6 Policy for SO

DOIs are minted at the level of the official data release:

- each official release gets one DOI;
- each new release gets a new DOI;
- the author list is updated at each release to reflect the collaboration membership at that time;
- the DOI resolves to a stable release landing page on data.simonsobservatory.org (and connected reference to the NASA LAMBDA website), which can, in turn, link to the relevant HIPPO collection, documentation, and related materials.

Product-level detail can still be represented within HIPPO through release pages, metadata, and documentation. That information does not need a separate public DOI by default.

4.7 Implementation of Policy

The DOI should resolve to an official release landing page on data.simonsobservatory.org, using a simple and predictable URL scheme such as data.simonsobservatory.org/DR1, data.simonsobservatory.org/DR2, and so on. That landing page can then link out to the relevant HIPPO collection, documentation, software, and related references. This keeps the DOI target stable even if the underlying storage or presentation layers change over time.
This also gives SO a clean operational model for coordinating data-release DOIs with paper publication:

- create a placeholder release page at the final data.simonsobservatory.org/DRX URL before the public release;
- mint the DOI to that landing page before publication, so the DOI can already be cited in the release paper;
- include the DOI in the paper and related release materials;
- update the landing page with the full release content, HIPPO links, and documentation when the release becomes public;

Even in a one-DOI-per-release model, relationship metadata remain useful. For example:

- IsNewVersionOf and IsPreviousVersionOf can connect one official release to the next;
- citation metadata can connect the release DOI to methods or release papers.

4.8 Conclusions

In conclusion, we summarize the crucial aspects of the Data Licensing and attribution policy as:

- public SO data products should carry an explicit license;
- the default license should be CC BY 4.0;
- departures from CC BY 4.0, especially the use of NC or SA, should require specific justification and approval because they may hinder reuse;
- do not rely only on the release paper for data citation;
- mint one DOI per official data release;
- update authorship at each release to match collaboration membership at that time;
- use the release DOI for data citation and papers for methods and scientific context.

This gives SO a clear, broadly legible policy that protects attribution without undermining the value of making data public, treating public data releases as first-class, citable research products.

5. Appendix: Data Licensing Comparison With Other Archives and Experiments

The broader astrophysics and cosmology landscape does not converge on a single model, but several relevant patterns are clear.

- ESO Science Archive: https://www.eso.org/public/science/archive/. ESO data are typically proprietary for about one year and then become openly accessible worldwide through the archive. ESO states that these data retain ESO copyright and are distributed under a CC BY license.
- MAST: https://archive.stsci.edu/publishing/data-use. Most public data hosted at MAST are in the public domain and can be used without restriction after any exclusive-access period expires. MAST also notes that some community-contributed high-level science products are distributed under CC BY 4.0.
- IRSA: https://irsa.ipac.caltech.edu/data_use_terms.html. IRSA states that most public data it serves have no usage restrictions, while some collections remain proprietary for a period or carry separate copyright terms. IRSA emphasizes acknowledgments and citations rather than a single archive-wide attribution license.
- SDSS: https://www.sdss.org/collaboration/image-use-policy/. SDSS states that all data released in its public data releases are considered to be in the public domain. Its website images are separately provided under CC BY.
- NASA / LAMBDA: https://science.data.nasa.gov/about/license and https://lambda.gsfc.nasa.gov/product/aws/. NASA states that, unless otherwise marked, scientific data from NASA-led missions are licensed as CC0. LAMBDA also states that its hosted data are freely available for use, but in practice the applicable terms may still derive from the underlying mission or project for a given collection, so this should not be read as a single project-independent rule for every dataset hosted there.
- DESI: https://data.desi.lbl.gov/doc/acknowledgments/. DESI states that DESI data are licensed under CC BY 4.0, and it provides explicit acknowledgment and citation guidance for users of the public data releases.
- ACT: https://act.princeton.edu/front, https://act.princeton.edu/document/290, and https://lambda.gsfc.nasa.gov/product/act/act_dr6.02/. ACT states that its data products are released publicly through NASA LAMBDA and that ACT data and analysis will be made public in a timely manner. This is a strong public-release model, though I have not found an experiment-specific public data license statement comparable to CC BY.
- SPT: https://pole.uchicago.edu/, https://pole.uchicago.edu/public/data/camphuis25, and https://pole.uchicago.edu/spt/documents/dmp/Data_Management_Plan.pdf. SPT provides public data products and states in its data management plan that final data products will be released through LAMBDA and publication supplements, with reuse, acknowledgment, and citation guidance provided alongside the data. I have not found a single archive-wide license statement for SPT data products.
- POLARBEAR: https://lambda.gsfc.nasa.gov/product/polarbear/ and https://www.physics.lbl.gov/wp-content/uploads/sites/17/2019/08/Polarbear-DMP.pdf. POLARBEAR data products are distributed through LAMBDA, and its data management plan states that publicly available data will not have usage restrictions once released. I have not found a formal public-facing license page for the experiment.

Taken together, these examples show that public astrophysics data are often made broadly reusable, but the mechanism varies. In that landscape, CC BY 4.0 is a well-supported and defensible choice for SO.

6. Appendix: DOI Options Considered

- Option 1 - Cite only the data release paper: this is the simplest operational model. It gives users a familiar academic citation target and avoids introducing an additional identifier. Its weakness is that it collapses paper citation and data citation into the same object. That is convenient, but it means the release is not represented as a standalone data product with its own persistent identifier.
- Option 2 - One DOI per official data release: this is the recommended approach. Each official release receives one DOI, and each new release receives a new DOI with an updated collaboration author list. This preserves most of the simplicity of the paper-only model while giving the release a proper data identifier. It also keeps citation practice simple and consistent: users cite the release DOI for the data and cite the relevant papers for methods or analysis details.
- Option 3 - Hierarchical DOIs for releases, collections, and products: this approach would support relationships between releases, collections, and individual products. It is attractive if the main goal is to represent internal data structure directly in the DOI layer. The downside is complexity. It introduces more identifiers, more metadata maintenance, and more choices about which DOI users should cite. That additional flexibility is real, but it weakens consistency and increases policy overhead.
- Option 4 - One DOI per data product or sub-collection: this is the most granular approach. It provides the finest possible citation targets, but it also has the highest operational cost. It increases DOI minting volume, complicates authorship and maintenance, and risks fragmenting citations across many identifiers. For SO, that level of granularity is not worth the cost as a default policy.