Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Redistributing Lossless-Compression Test Corpora: An Audit of Nine Corpora and a File-Level Publication Procedure

Eugen Spivak

publications@eugenspivak.com

Independent researcher https://orcid.org/0009-0005-7509-8609

Research Article

Keywords: lossless data compression, benchmark corpus, dataset licensing, data provenance, reproducibility, research data publication, GPU texture formats

Posted Date: October 7th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11273530/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: The authors declare potential competing interests as follows: The author develops commercial data-compression software. No other competing interests.

Abstract

Decades of published lossless-compression results rest on a few dozen classic test files, assembled between 1987 and 2006, and almost nobody can say whether those files may be copied. Anyone who wants to mirror, extend or replace them has to establish that right file by file, and no published procedure says how. I audited the public pages and distribution archives of nine widely used compression test corpora, as available on 2026-10-03, and then applied a file-level publication procedure to a new data family. The audit found 76 files with no stated license across eight of the nine corpora. Only one corpus states a license for each file. Two publish checksums. None records both the command and tool version used to construct a file, and none has a versioned persistent identifier.

The procedure gives each file a lineage record (source URL, input checksums, the generating tool, its version and the exact command) and license evidence (the license page, the permitting sentence quoted verbatim and dated, and the attribution owed). A default-exclusion gate admits a file only when an independent second review, which does not see the first classification, reaches the same verdict. On the first data family the second review removed 30 of the 774 candidates (3.9%) that the first pass had accepted, every one of them a question of evidence or content that a review of license names alone would have missed. Most of the record fits Croissant, Croissant-RAI, SPDX 3.0 and W3C PROV. Three fields do not: the quoted permitting evidence, command determinism and second-review agreement.

Applied to game-asset data (GPU texture containers, glTF meshes, Ogg audio), the procedure produced two public records: 744 CC0 files (1,081,386,209 bytes), 700 of them generated by five public encoders, and 13 unmodified CC BY 4.0 files. A post-publication audit changed 118 of the 7,440 manifest cells (1.6%) and no data file. All seven archives rebuild byte for byte from their members, but only with every writer parameter fixed: with the host-system field a Unix host writes, none of the seven matches. The conclusion is bounded to what was measured: a corpus can be published with per-file evidence at the cost of a second review, and an archive checksum is reproducible only when its writer parameters are published with it.

1. Introduction

Every published ratio of a lossless compressor is a ratio on some set of files, and for most of the literature that set is one of a handful of corpora assembled between 1987 and 2006. The files are small, stable and easy to obtain, which is why results on them can be compared across decades. They also predate much of the data that storage systems now hold. More importantly for this paper, eight of the nine corpora audited here state no license for their files: 76 files in all (Section 3).

What happens if this stays as it is? A benchmark built on these files cannot be redistributed with confidence, a mirror cannot be defended, and a corpus meant to replace them inherits the same gap unless its builder proves, file by file, that each may be shared. Machine-learning datasets reached the same point earlier, and audits there found license information missing or wrong for a large share of popular datasets [Longpre2024; Rajbahadur2021; Kim2025; Jewitt2026]. The license status of compression corpora has not been examined at all; I found no audit of it.

The question is not which files to measure on. It is whether anyone may copy them, and how a corpus builder proves it.

This paper has two objectives. The first is to establish the current state: what the corpora in use actually publish about license, origin, integrity and construction. The second is to show a way from that state to one in which every file in a public corpus carries evidence a skeptical reader can check, and to report what the way costs and what it catches. The scope is lossless-compression test corpora and one new data family; the procedure is general, the measurements are not.

Most measurements of lossless compressors still use:

- the Calgary corpus [Bell1989];
- the Canterbury corpus [Arnold1997];
- the Silesia corpus [Silesia];
- enwik8 and enwik9, prefixes of a 2006 English Wikipedia dump used by the Large Text Compression Benchmark [LTCB].

Newer corpora have been proposed [Holub2011; Squash], and domain benchmarks exist for scientific floating-point data [Zhao2020; Chen2024], genomic sequences [Kryukov2020] and source code [Boffa2025]. A general, format-rich corpus with file-level redistribution evidence has not emerged in the literature reviewed here. Collecting newer data is comparatively easy. The harder part is publishing it so that each file can be redistributed with evidence that can survive legal or compliance review.

The paper does four things:

1. **An audit** of the license and provenance information that nine compression test corpora publish (Section 3).

2. **A file-level publication procedure**: a lineage and license-evidence record, a five-condition gate with default exclusion, and an independent second review that must agree file for file, mapped onto existing metadata standards (Section 4).

3. **Two public records** of game-asset data produced by that procedure, with their composition and license evidence (Section 5).

4. **Measured results** on the published records: verification, what the gate excluded and why, reproducibility of the archive bytes, and a post-publication audit (Section 6).

Section 7 states what the results mean, what they do not establish, and what a corpus builder should do differently. Section 8 is the checklist. The data, tools and checks are all public; Section 9 says how to obtain and recompute every number.

2. Related work

The literature answers three separate questions, and it helps to keep them apart: which corpora exist and what they document; how a dataset should be described; and what license audits of other fields have found.

Which corpora exist, and what they document. The Calgary corpus came with the survey that introduced it [Bell1989]. The Canterbury corpus replaced it with files chosen by a documented, repeatable selection method, and was declared frozen so that results stay comparable [Arnold1997]. Holub et al. argued that static corpora omit newer file types and proposed the Prague corpus with a test bed [Holub2011]. The Silesia corpus [Silesia] and enwik8 / enwik9 [LTCB] became the de facto standards for larger inputs. The Squash corpus proposal [Squash] is the closest predecessor of this work: it gives a "Legal Status" paragraph for each of its files, including copyleft (GPLv2) and share-alike (CC BY-SA 3.0) items. It has no manifest, checksums or construction commands. Domain benchmarks for scientific data [Zhao2020; Chen2024], genomic sequences [Kryukov2020] and source code from the Software Heritage archive [Boffa2025] were published as papers. None of them examines the redistribution status of each input file. The practical result is simpler: selection methods are sometimes documented, licenses rarely are, and construction commands are not.

How a dataset should be described. Datasheets [Gebru2021], data statements [Bender2018] and data cards [Pushkarna2022] are human-readable questionnaires at the level of a whole dataset, including its legal and personal-data aspects. The FAIR principles [Wilkinson2016] ask for rich metadata and clear licenses. Machine-readable descriptions exist as Croissant [Akhtar2024] with its responsible-AI extension [Jain2024], the SPDX 3.0 dataset profile [Bennet2025] and W3C PROV [Missier2013]; content-hash provenance at archive scale is shown by Software Heritage [Rousseau2020]. What these standards share is the dataset as the unit of description. What a corpus builder needs is the file as the unit, with the evidence attached to it; Section 4.6 shows where the standards stop.

What license audits elsewhere have found. The Data Provenance Initiative audited more than 1,800 text datasets. It found licenses omitted on hosting platforms for more than 70% of them and miscategorized for more than half [Longpre2024]. Rajbahadur et al. showed that the license of a public image dataset can permit commercial use while its sources do not [Rajbahadur2021]. Kim et al. argued that license text alone is not enough and that redistribution chains must be traced [Kim2025]. Jewitt et al. found that required license text and attribution rarely propagate downstream [Jewitt2026]. Bandy and Vincent showed with a retrospective datasheet how undocumented sources accumulate as "documentation debt" [Bandy2021]. Several corpora publish only openly licensed material:

- The Stack, with licenses detected per repository [Kocetkov2022];
- Common Pile, public-domain and openly licensed text with its construction code [Kandpal2025];
- Lit2Vec, conservative metadata-based license screening and reconstruction from pinned sources [Amiri2026];
- SetGo, which checks license and provenance metadata before publication [Wilkinson2026].

The common problem is that a dataset-level license can hide file-level exceptions. The practical response is to keep the evidence per file.

What this paper adds. I do not claim the first corpus with per-file license information (Squash has it), nor a new metadata standard. This paper closes three practical gaps. First, it measures what compression corpora actually publish. Second, it keeps the evidence behind each license and the command behind each constructed file. Third, it tests an independent second-review gate on a new data family and reports what was published, what was excluded, and what still went wrong.

3. An audit of license and provenance information in compression corpora

3.1 Method

For each corpus I read its main public page or repository README and listed the members of its distribution archives, as available on 2026-10-03. Calgary, Canterbury and Canterbury Large were checked through the Canterbury corpus site. For enwik8 / enwik9 I read the LTCB description of the test data.

Each corpus was scored on five criteria:

- **L**: a license stated for each file;
- **O**: the origin of each file;
- **H**: published checksums;
- **C**: for constructed or modified files, a description of how they were made;
- **P**: a versioned persistent identifier.

Scores are Yes, Partial (some files, or prose without a source link) or No ("not found on the pages read"). The audit records what a reader can find, not what the authors may know. That is the only state that matters to someone who wants to reuse the files.

3.2 Results

Table 1. Published license and provenance information (checked 2026-10-03).

corpus	files	L license per file	O origin per file	H checksums	C construction	P persistent id
Calgary [Bell1989]	18	No	Partial (category descriptions)	No	No	No
Canterbury [Arnold1997]	11	No	Partial (category descriptions)	No	Partial (selection method in the paper)	No (declared frozen)
Canterbury Large	3	No	Partial	No	No	No
Silesia [Silesia]	12	No	Yes (prose, per file)	Yes (MD5 per file)	Partial (one truncation described)	No
enwik8 / enwik9 [LTCB]	2	No (license of the Wikipedia text not stated)	Yes (named dump; the page notes the link no longer works)	Yes (MD5, SHA-1)	Yes (prefixes of the dump)	No
Pizza&Chili texts [PizzaChili]	6	No	Yes (prose, "obtained from")	No	Partial (concatenation described)	No
Squash corpus [Squash]	9	Yes (prose; includes GPLv2 and CC BY-SA 3.0)	Partial	No	Partial (the random file's source described)	No
zlib-ng corpora [ZlibngCorpora]	aggregate	No ("see the sources above for any licensing information")	Partial (by aggregated corpus)	No	No	No
Kodak images [Kodak]	24	No (the host's "understanding" of release "for unrestricted usage")	Partial	No	No	No
records of this paper	757	Yes (SPDX id + quoted evidence)	Yes (URL + input SHA-256)	Yes (file and archive)	Yes (tool, version, command)	Yes (version and concept DOIs)

The distribution archives of Calgary, Canterbury, Canterbury Large and enwik8 contain the data files only, with no license or README file. The Prague corpus could not be assessed: its page is a frame set, and the corpus files were not reached.

The current state in one number. The corpora that state no license hold 18 (Calgary) + 11 (Canterbury) + 3 (Canterbury Large) + 12 (Silesia) + 2 (enwik8 / enwik9) + 6 (Pizza&Chili) + 24 (Kodak) = **76 files**. The zlib-ng collection only aggregates other corpora and is not counted twice. These 76 files are the classic test inputs of Section 2, and for none of them can a reader find a statement that permits redistribution. The next time a benchmark cites them, that is what the citation rests on.

Four observations follow. Each is a gap a corpus builder can close deliberately, and each turns up again in Section 6 as something the procedure had to handle:

- **License.** Only the Squash proposal states a license per file, and some of its licenses are copyleft or share-alike. Whether redistribution is permitted is unstated for every other corpus. In practice the corpora circulate through mirrors and aggregations (zlib-ng's collection explicitly refers readers back to the sources), so the gap propagates with every copy.
- **Integrity.** Silesia and LTCB publish checksums, so a reader can confirm that their copy is the reference copy. For the other corpora, nothing but the size identifies a file, and a silently altered mirror would pass as the original.
- **Construction.** The corpora that changed a file in construction (a truncated database and concatenated applications in Silesia, concatenated sources in Pizza&Chili, a random file in Squash) describe the change in prose. None gives a command that regenerates the bytes, so none of those files can be rebuilt from its source.
- **Personal data.** No corpus carries a statement about personal data. Silesia includes two medical images, an X-ray of a child's hand and a magnetic-resonance image of a head; its page says nothing about consent or de-identification. I do not suggest that anything was wrong at the time. A publication gate written today would ask the question (condition (c) in Section 4.3).

4. A file-level publication procedure

The target state is simple to state: every published file carries evidence that it may be shared, and a reader can check that evidence without trusting the publisher. The procedure has six parts: file classes, the record, the publication gate, the independent second review, packaging, and its mapping to existing standards.

4.1 Classes

The procedure starts from a larger private collection of test files that cannot be published in full. Every candidate file falls into one of three classes:

- **public originals**: the author permits redistribution of the unmodified file;
- **generated files**: produced from eligible inputs by a recorded command;
- **restricted files**: unclear, missing, non-commercial or no-redistribution terms, personal data, or doubtful origin. These are never published.

Classes apply per file, never per data set. A site that states a license can still host individual files under other terms (Section 6.2). In my view, this is the most useful habit in the procedure, and one that is easy to skip.

4.2 The lineage and license-evidence record

At collection time each file gets a lineage row. It records the origin, the retrieval time, the checksum the upstream publishes (if any), the file's SHA-256, the inputs and their checksums, and the tool environment (the library versions that determine the bytes). It also records the command, whether re-running the command reproduces the bytes, a personal-data flag, the license evidence and what was checked after download. The published manifest carries the subset in Table 2. A separate per-source table (licence_table.tsv) holds:

- the license page;
- the SPDX identifier [SPDX];
- the sentence that permits redistribution, quoted verbatim and dated;
- the number of files the source covers.

Copies of the license pages are saved when they are read. Pages change. Two years from now, when someone asks why a file is in the record, the saved copy and its date are the answer.

Table 2. Columns of the published MANIFEST.tsv (one row per file).

column	content
path, bytes, sha256	path inside the record, exact size, SHA-256
archive	the record archive holding the file
licence	SPDX identifier
attribution	the attribution text the license requires, or "none required"
changes	(attribution set only) modification statement
source_url	direct URL of the upstream file; empty for files generated without third-party input
inputs	generated files: each input file and its SHA-256
generation	generated files: tool, version, exact command
format	detected format and main parameters

4.3 The publication gate

A file may be published only if all five conditions hold:

1. **(a)** Its license is recorded with its URL and the quoted sentence that permits redistribution of the unmodified file.

2. **(b)** For a generated file: every input is itself eligible, the generating tool's license places no condition on its output, and the command is recorded.

3. **(c)** It contains no personal data. For example, photographs showing identifiable people are excluded.

4. **(d)** Its status is not unverified, unclear, "per file" without the per-file term, or mixed.

5. **(e)** The attribution text the license requires is included in the record.

Two further rules apply:

- **Default exclusion.** A file that fails a condition, or about which there is doubt, is left out. Doubt is a verdict, not a backlog item.
- **Single-license first records.** The first record holds CC0 files only [CC0]. Attribution-bearing files (CC BY 4.0 [CCBY4]) follow in a separate record with its own review and an attribution file.

4.4 Independent second review

A first pass (staging) produces a candidate list with per-file evidence. A second reviewer then re-derives eligibility for every file from the manifest, the lineage rows and the saved license texts, without seeing the first pass's classification. The reviewer re-reads the live license pages and re-downloads sources to compare checksums. A file is published only if both lists call it eligible. Every disagreement is resolved with evidence or excluded.

The review also scans every public cell and text for private information: local paths, machine and user names, internal document names and secrets. The term list grows with every miss. Public texts describe what a file is and how it was produced with public tools. They add no analytical annotations about content, so the corpus does not steer its users toward any one kind of compressor.

The record is created as a draft. I read the review and publish. Errors found after publication are corrected by a new version, never by silent replacement.

AI assistants supported staging, the independent second review and preparation of the manuscript. I made every publication decision and am responsible for the content.

4.5 Packaging and verification

Each record holds:

- the files in uncompressed (STORED) ZIP archives, members sorted by name, with fixed timestamps;
- the manifest, and SHA256SUMS.txt with the SHA-256 of every file and every archive;
- the license texts;
- a README with the credits.

A public repository holds the manifests, the license tables, the generation recipes and a standard-library Python script. The script downloads a record through the public Zenodo API and checks:

- every file against Zenodo's MD5;
- every archive against SHA256SUMS.txt;
- after unpacking, every member's SHA-256 and size against the manifest.

A missing file and an unlisted member both count as failures. Each record has a version DOI and a concept DOI that resolves to the newest version. Nothing in the check requires the reader to trust the publisher, including me.

4.6 Mapping onto existing metadata standards

A data office that already emits one of the machine-readable standards will ask whether this record is a new format. It is not. Table 3 maps the record onto four standards, as published on 2026-10-03.

Table 3. Crosswalk of the record's fields ("–": no corresponding element found).

field of the record	Croissant 1.0 [Akhtar2024]	Croissant-RAI [Jain2024]	SPDX 3.0 [Bennet2025]	W3C PROV-O [Missier2013]
file, size, SHA-256	cr:FileObject (contentSize, sha256)	--	File, verifiedUsing (hash)	prov:Entity
license (SPDX id)	license (dataset level in the examples)	--	licenseExpression	--
attribution text	--	--	attributionText	--
quoted permitting sentence, page, date	--	--	only as a free-text Annotation	--
upstream URL	contentUrl	rai:dataCollection (prose)	downloadLocation	prov:wasDerivedFrom
inputs and their SHA-256	--	rai:dataManipulationProtocol (prose)	relationships between elements	prov:used
tool, version, command	--	rai:dataManipulationProtocol (prose)	-- (tools only at creation-info level)	prov:wasGeneratedBy, prov:Activ
determinism of the command	--	--	--	--
personal-data flag	--	rai:personalSensitiveInformation	dataset-level personal-information flag	--
second-review agreement	--	--	--	--
versioning	version	rai:dataReleaseMaintenancePlan	packageVersion	--

Three fields have no structured home: the quoted evidence, the determinism of a generation command, and the agreement of an independent review. If a data office already uses one of these standards, the practical change is small. Add three fields: quoted license evidence, command determinism and second-review agreement. The rest already maps onto the existing standards.

5. Application: game-asset records

5.1 Why game assets first

GPU texture formats are poorly represented in the classic corpora. These are block-compressed payloads such as BC1-BC7, ETC2 and ASTC, stored in DDS, KTX2 and .astc containers. The same holds for 3-D scene files (glTF 2.0 and its binary form GLB) and game audio. The family also suits a first test of the procedure, which is why it went first: good CC0 sources exist, and the public encoders that write the payloads carry permissive licenses that place no condition on their output. The literature search behind Section 2 found no public benchmark set of such files for lossless compression.

5.2 What was published

Table 4 lists the two records and their three published versions.

Table 4. Published records, as read back from the public Zenodo API on 2026-10-03.

record	version DOI (concept DOI)	published	license	files	file bytes	archives
game assets, CC0 set, version 1	10.5281/zenodo.23110892 (10.5281/zenodo.23110891)	2026-10-02	CC0 1.0	744	1,081,386,209	6
game assets, CC0 set, version 1.1	10.5281/zenodo.23112969 (same concept)	2026-10-03	CC0 1.0	744 (same files)	1,081,386,209	6
game assets, attribution set, version 1	10.5281/zenodo.23112972 (10.5281/zenodo.23112971)	2026-10-03	CC BY 4.0	13	675,680	1

The CC0 set is split by content into six archives (Table 5).

Table 5. Composition of the CC0 set (version 1.1 manifest).

archive	files	bytes	content
textures-dds	394	698,269,648	DDS with BC1-BC7 and BC6H payloads; cube, array and volume textures
wrappers	75	138,483,031	KTX2 with raw RGBA8, BC7 and ASTC 6x6 payloads, without and with zstd supercompression (levels 5 and 19); UASTC and BasisLZ
textures-astc	166	112,372,048	ASTC 4x4, 6x6, 8x8 (LDR and HDR) in .astc
textures-ktx2	62	74,172,672	KTX2 with BC7, ETC2 and ASTC payloads
meshes	29	54,327,815	glTF, GLB, Draco and quantized variants of six CC0 sample models
audio	18	3,760,995	15 Ogg Vorbis files and 3 uncompressed tar archives of whole packs
total	**744**	**1,081,386,209**	

By file type: 394 .dds, 166 .astc, 137 .ktx2, 12 .gltf, 12 .bin, 5 .glb, 15 .ogg and 3 .tar. 44 files are unmodified upstream files (29 mesh, 15 sound); 700 were generated. Sizes range from 2,413 to 22,370,164 bytes, with a median of 1,048,592.

The generated files come from CC0 sources:

- Poly Haven textures and one HDRI (415 files) [PolyHaven];
- ambientCG materials (218) [AmbientCG];
- the Kenney UI pack, packed into one atlas (22 of the 40 Kenney files) [Kenney];
- cube, array and volume textures from Poly Haven and ambientCG images (6);
- one procedural noise image generated for the record (36), with no third-party input.

The method is simple: one source image, several encoders and settings. Table 6 lists the five encoders and how many manifest rows name each.

Table 6. Encoders named in the generation column.

tool	version	license	rows
bc7enc_rdo	commit b943862	MIT or Unlicense	264
astcenc	5.7.0	Apache-2.0	175
KTX-Software ktx create	4.4.2	Apache-2.0	137
DirectXTex texconv	May 2026 release	MIT	104
etcpak	0.9.15 (Python wheel)	MIT wrapper over BSD-3-Clause	94

texassemble appears in 6 rows and Python tarfile in 3. A row may name two tools (a payload encoder and the KTX2 writer). The recipe in the public repository describes the steps around the commands, such as the mip-chain filter and the sRGB handling.

The attribution set holds three unmodified Khronos glTF sample models (RiggedFigure, BoxAnimated, Fox) as 6 .gltf, 4 .bin and 3 .glb files. Its attribution file names, for each model, the creator, the copyright notice, the license and a link to the material, as CC BY 4.0 section 3(a) requires [CCBY4].

5.3 License evidence

For each source, Table 7 gives the sentence that permits redistribution and the number of files it covers.

Table 7. License evidence of the CC0 set (licence_table.tsv; sentences checked on the live pages on 2026-10-02).

source	quoted sentence or statement	files
Poly Haven, polyhaven.com/license	"You can redistribute them, share them around, include them when sharing your own work, or even in a product you sell."	415
ambientCG, docs.ambientcg.com/license	"You can copy, modify, distribute and perform the assets, even for commercial purposes, all without asking permission."	218
Kenney, kenney.nl/support	"Yes, all game assets on the asset pages are public domain licensed (CC0)."	40
Khronos glTF-Sample-Assets, six model READMEs	section *Legal*: "Creative Commons Zero v1.0 Universal", linked to the CC0 legal code	29
Poly Haven and ambientCG together	both sentences above	6
own generation	no third-party input; released under CC0 1.0 with the record	36

During the review every source of a generated file was downloaded again and its SHA-256 compared with the inputs column. All matched: 29 Poly Haven source files, 14 named ambientCG archive members and 4 Kenney pack archives. The 29 mesh and 15 sound files were verified against the files at their source URLs. For each of the five encoders, the license named in the manifest matched the live license file in the tool's repository or package page.

6. Results

This section reports what the procedure produced and what it caught: the verification of the published bytes, the files the gate kept out, how fragile a "reproducible" archive turned out to be, and what a post-publication audit still found. Each subsection ends with what the result means for someone building a corpus.

6.1 Verification

The public checks were repeated on 2026-10-03:

- **Published bytes = staged bytes.** The MD5 that the public Zenodo API reports for each file of the three record versions equals the MD5 of the staged file: 10 of 10 (CC0 v1), 10 of 10 (CC0 v1.1), 7 of 7 (attribution set).
- **Manifest verification.** The public script in --local mode reported "744 verified, 0 wrong or unlisted, 0 missing" for the CC0 set and "13 verified" for the attribution set.

The practical result is that the published record is the staged record, and anyone can establish that in minutes with the script in Section 9 without taking the publisher's word for it.

6.2 What the gate excluded, and why

The independent review, measured. The review of the CC0 record received the 774 candidate files that the first pass had accepted. It worked without the first pass's classification, so it could remove files but not add them. Table 8 gives the outcome.

Table 8. Outcome of the independent review of the CC0 record.

	files
candidates accepted by the first pass	774
removed by the independent review	**30 (3.9%)**
-- inconsistent or incomplete author credit (two sample models)	24
-- rights-holder statement not confirmable (two recordings)	2
-- people in a photograph (one HDR environment image)	4
published	744

None of the 30 had a wrong license identifier. Each was a question of evidence or content, the kind of error a review working from license names alone does not catch. A single-review process would have published all 30. That is the cost and the return of the second review in one line: a few hours of work against 30 files that would later have to be withdrawn.

The first pass had already left out other game-asset files before the review. Each category, of both passes, is a lesson for anyone assembling a public corpus:

- **A second license beside a CC0 credit.** Test images inside a tool's source repository were credited as CC0, while the repository's machine-readable license metadata [REUSE] declared an additional license for the same paths: mixed terms, condition (d).
- **Permissive wording without a license.** The Kodak photographic test images are distributed with a host's statement that they were released "for unrestricted usage" (Table 1), but under no named license. Under default exclusion, wording that is not a license is not enough.
- **Attribution licenses.** CC BY files were moved to a later record. Only those whose attribution could be stated completely were published (the attribution set).
- **Inconsistent or incomplete credit.** Author credits that differed between two pages of the same publisher, or between the current and an earlier README of the same model, or that named nobody for the model itself.
- **People in a photograph.** An HDR environment photograph showing people was excluded under condition (c), although its license was CC0.
- **Rights holder not confirmable.** Recordings on a media repository carried a pending license-review flag or a dedication made by the uploader only, and the rights holder's own page could not be reached.
- **Locally modified tools.** Files written by a tool with local changes cannot be regenerated from the public tool. They were deferred.

In the first, fourth and fifth categories the source's top-level license statement was permissive; the problem sat with individual files. The lesson is straightforward: the gate has to work per file, not per source. Checking only the site's license page can admit files that later have to be withdrawn.

6.3 Archive checksums reproduce only with every writer parameter

The record README states that the archives are STORED, sorted and timestamp-fixed, "so the archives are reproducible from the files". I tested that claim rather than trust it. I unpacked each of the seven published archives and rewrote it with CPython 3.14.7's zipfile on Windows [PythonZipfile], using:

- members in name order;
- method STORED;
- date 1980-01-01 00:00:00;
- Unix permission bits 0644 in the external attributes;
- Zip64 extra fields forced on every member.

All seven rebuilt archives have exactly the published SHA-256. Two changes, each a writer default the README does not mention, break this:

- **Zip64 extra fields not forced.** Every rebuilt archive is exactly 20 bytes shorter per member (260 bytes for the 13-member archive, 7,880 for the 394-member one). None of the seven matches.
- **The host-system field set to Unix** (create_system = 3, which CPython writes on Linux and macOS, instead of 0 on Windows). None of the seven archives matches.

In both cases the member data still verify against the manifest; only the archive checksums change. This is the result I would ask a reader to remember: an archive checksum is reproducible only if the publication states every writer parameter that reaches the bytes, including parameters inherited from the host operating system [ZipAppnote]. The README states three of the five. This paper states all five. The per-file manifest makes the member data verifiable independently of the archive bytes, which is why it should accompany any archive checksum. Section 9 says where all five are published.

6.4 Post-publication audit

After version 1 was published, the independent review found defects in its manifest. A cell-by-cell comparison of the version 1 and version 1.1 manifests (both public) shows that **118 of 7,440 cells (1.6%) in 118 of 744 rows** changed: 112 in generation and 6 in inputs. No other column, data file or checksum changed. The public errata give five causes:

- 94 rows named the wrong license for the etcpak Python wheel;
- in 53 of those rows, an inserted tool note split the ktx create format names, so the commands did not run as written;
- 6 rows referred to an internal note instead of giving their source checksums;
- 3 rows named an internal script instead of the public tool and its parameters;
- 15 rows were reworded to describe how each file was written.

None affected eligibility; all were documentation defects. Version 1.1 replaced only the manifest and the README. Version 1 stays citable, and the concept DOI now resolves to version 1.1. Two things changed in the procedure as a result: the private-information scan now carries the names of internal folders and documents, and every public cell is scanned again before a draft is created. The practical lesson is to plan the correction path before publication. A first release may need one, and the versioning model should already know how to carry it.

7. Discussion and limitations

Three findings matter. The classic corpora publish no license for 76 of their files. A second review that works blind to the first catches errors of evidence and content that a review of license names cannot, at a rate of 3.9% of the candidates on this family. And an archive checksum reproduces only when every writer parameter is published, including the ones a library takes from the host.

What a corpus builder should do differently. The choice is not between publishing with evidence and publishing without. It is between three ways of handling the evidence, and they differ in cost and in what they catch:

- *Dataset-level license only*, the current practice of eight of the nine corpora audited. Cost: nothing. Catches: nothing; every file-level exception of Section 6.2 passes.
- *Per-file license names*, the Squash form. Cost: one column. Catches: mixed terms when the builder notices them. Misses: wrong credits, unconfirmable rights holders, people in photographs, wording that is not a license, since none of these changes the license name.
- *Per-file evidence with a blind second review*, this paper. Cost: the saved pages, the quoted sentences, the commands, and a second pass of a few hours per family. Catches: the 30 files above, and it leaves a record a reader can re-derive.

For a corpus meant to be cited for years, the third is the shorter path, because every file the first two publish wrongly has to be withdrawn later, with the benchmarks that used it. For a private test set that never leaves one team, the first is enough.

One family, about one gigabyte. The procedure has so far been applied to one data family. The records total 1.08 GB and are narrow in content. They test the procedure; they do not replace the classic corpora. This prevents any claim about compression ratios or about other families; it does not change the recommendation, which concerns the evidence, not the content.

Single-license records narrow the content. Requiring CC0 first, and excluding share-alike and non-commercial terms throughout, leaves out much redistributable data. A corpus built this way over-represents publishers who choose public-domain dedications. A builder who needs broader content can add attribution and share-alike records under the same gate; the procedure does not forbid them, it sequences them.

A license check is a point-in-time check. The quoted sentences were verified on the live pages on given dates; pages change. Saved copies and dates document what was true at publication. They cannot guarantee that the upstream terms were themselves correct. The procedure is not legal advice. It records evidence so that a reader can judge for themselves.

The audit reads what is published. Table 1 scores the information a reader can find on a corpus's pages and in its archives. Authors may hold more, and some corpora are described in papers or theses not read here. A corpus author who has the evidence can close the gap by publishing it; the audit measures the gap, not the authors.

Independence has limits. The second review did not see the first classification, but both worked from the same lineage rows, so an error there (a wrong source URL, say) can pass both. Re-downloading sources and comparing checksums closes much of that gap where it was done. The remaining exposure is a wrong row that both reviews accept; the post-publication audit is the control for it.

Neutral descriptions are a choice. The records describe what each file is and how it was made. They contain no measurements or analytical annotations, which keeps them independent of any one compressor but leaves the characterization of the data to the user.

What any published archive needs. The results point to a short list of properties that a corpus, or any archive meant to be trusted by strangers, should have. These are properties an organization can write into a data policy or a task order:

- one record per file that states origin, license evidence and, for a generated file, the command that made it;
- checksums at two levels, the file and the container, with every writer parameter that reaches the container bytes;
- a versioned persistent identifier, and corrections only as new versions;
- descriptions of what each file is, without annotations that favor one consumer;
- verification that a reader can run without trusting the publisher.

8. A checklist for corpus builders and benchmark authors

If you are building or maintaining a corpus, start here:

1. **For every test file, write down where it came from and the sentence that lets you share it.** No sentence, no redistribution: link to the source instead.

2. **Quote, date and save the license page.** A license name alone is not evidence.

3. **For every file you generated, record the tool, its version and the exact command.** Re-run it once to see whether the bytes repeat, and say so.

4. **Have a second person, or a second process, check eligibility without seeing the first list.** Publish only what both accept.

5. **Exclude by default.** Credits that disagree, permissive wording without a license, people in photographs and files from locally modified tools stay out until resolved.

6. **Publish checksums for files and for archives, and the archive writer's parameters.** Otherwise the archive checksum cannot be reproduced.

7. **Use a versioned DOI, and correct by new versions.** Never replace a published file silently.

8. **When you report a benchmark, cite the corpus by DOI and version, and state which files you used.**

If you do only one of the eight, do the first. It is the one the 76 files of Section 3 are missing.

The conclusion is narrow, but practical. A test corpus can be published with per-file license evidence, and on this first family the independent second review removed 30 of 774 files that the first pass would have published. New corpus builders should therefore keep the per-file record, use the gate and require an independent second review. Authors who use the classic corpora should state exactly which files they used and what redistribution evidence they relied on. The next step is a second data family, this time recording review hours per file so that the cost can be measured beside the 3.9% removal rate.

Declarations

Data and code availability

- CC0 set: https://doi.org/10.5281/zenodo.23110891 (version 1: 10.5281/zenodo.23110892; version 1.1: 10.5281/zenodo.23112969) [CorpusCC0].
- Attribution set: https://doi.org/10.5281/zenodo.23112971 (version 1: 10.5281/zenodo.23112972) [CorpusAttr].
- Manifests, license tables, recipes and the fetch-and-verify script: https://github.com/innozip/innozip-corpus, commit 189d3eb (scripts MIT, documentation CC BY 4.0) [CorpusRepo].

- The archive-rebuild check (Section 6.3), the manifest comparison (Section 6.4) and the composition statistics (Section 5.2) are short standard-library Python scripts in the repository's scripts/ folder [CorpusRepo].
- The audit (Table 1) used the pages and archives listed in the references, retrieved 2026-10-03.

Competing interests.

The author develops commercial data-compression software. No other competing interests.

References

1. All DOIs and arXiv identifiers were checked against Crossref or the arXiv API on 2026-10-03; web resources were retrieved on that date.
2. [Akhtar2024] M. Akhtar, O. Benjelloun, C. Conforti, et al. Croissant: A Metadata Format for ML-Ready Datasets. In Proc. Eighth Workshop on Data Management for End-to-End Machine Learning (DEEM '24), pp. 1-6, 2024. doi:10.1145/3650203.3663326
3. [AmbientCG] ambientCG. License. https://docs.ambientcg.com/license/
4. [Amiri2026] M. Amiri, J. Mohammad Jafari, S. Mostafapour, T. Bocklitz. Lit2Vec: A Reproducible Workflow for Building a Legally Screened Chemistry Corpus from S2ORC for Downstream Retrieval and Text Mining. arXiv:2604.12498, 2026.
5. [Arnold1997] R. Arnold, T. Bell. A corpus for the evaluation of lossless compression algorithms. In Proc. DCC '97, Data Compression Conference, pp. 201-210, 1997. doi:10.1109/DCC.1997.582019
6. [Bandy2021] J. Bandy, N. Vincent. Addressing "Documentation Debt" in Machine Learning Research: A Retrospective Datasheet for BookCorpus. arXiv:2105.05241, 2021.
7. [Bell1989] T. Bell, I. H. Witten, J. G. Cleary. Modeling for text compression. ACM Computing Surveys 21(4):557-591, 1989. doi:10.1145/76894.76896
8. [Bender2018] E. M. Bender, B. Friedman. Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science. Transactions of the ACL 6:587-604, 2018. doi:10.1162/tacl_a_00041
9. [Bennet2025] K. Bennet, G. K. Rajbahadur, A. Suriyawongkul, K. Stewart. Implementing AI Bill of Materials (AI BOM) with SPDX 3.0: A Comprehensive Guide to Creating AI and Dataset Bill of Materials. arXiv:2504.16743, 2025.
10. [Boffa2025] A. Boffa, R. Di Cosmo, P. Ferragina, A. Guerra, G. Manzini, G. Vinciguerra, S. Zacchiroli. On the compressibility of large-scale source code datasets. Journal of Systems and Software 227:112429, 2025. doi:10.1016/j.jss.2025.112429
11. [CC0] Creative Commons. CC0 1.0 Universal, legal code. https://creativecommons.org/publicdomain/zero/1.0/legalcode
12. [CCBY4] Creative Commons. Attribution 4.0 International, legal code. https://creativecommons.org/licenses/by/4.0/legalcode
13. [Chen2024] X. Chen, J. Tian, I. Beaver, C. Freeman, Y. Yan, J. Wang, D. Tao. FCBench: Cross-Domain Benchmarking of Lossless Compression for Floating-Point Data. Proc. VLDB Endowment 17(6):1418-1431, 2024. doi:10.14778/3648160.3648180
14. [CorpusAttr] E. Spivak. INNOZIP compression corpus: game assets, attribution set (version 1) [Data set]. Zenodo, 2026. doi:10.5281/zenodo.23112971
15. [CorpusCC0] E. Spivak. INNOZIP compression corpus: game assets, CC0 set (versions 1 and 1.1) [Data set]. Zenodo, 2026. doi:10.5281/zenodo.23110891
16. [CorpusRepo] innozip-corpus, public repository, commit 189d3eb. https://github.com/innozip/innozip-corpus
17. [Gebru2021] T. Gebru, J. Morgenstern, B. Vecchione, J. Wortman Vaughan, H. Wallach, H. Daumé III, K. Crawford. Datasheets for datasets. Communications of the ACM 64(12):86-92, 2021. doi:10.1145/3458723
18. [Holub2011] J. Holub, J. Reznicek, F. Simek. Lossless Data Compression Testbed: ExCom and Prague Corpus. In Proc. 2011 Data Compression Conference, p. 457, 2011. doi:10.1109/DCC.2011.61
19. [Jain2024] N. Jain, M. Akhtar, J. Giner-Miguelez, R. Shinde, et al. A Standardized Machine-readable Dataset Documentation Format for Responsible AI. arXiv:2407.16883, 2024.
20. [Jewitt2026] J. Jewitt, G. K. Rajbahadur, H. Li, B. Adams, A. E. Hassan. Permissive-Washing in the Open AI Supply Chain: A Large-Scale Audit of License Integrity. arXiv:2602.08816, 2026.
21. [Kandpal2025] N. Kandpal, B. Lester, C. Raffel, S. Majstorovic, et al. The Common Pile v0.1: An 8TB Dataset of Public Domain and Openly Licensed Text. arXiv:2506.05209, 2025.
22. [Kenney] Kenney. Support (license statement). https://kenney.nl/support
23. [Kim2025] J. Kim, S. Sohn, G. J. Jo, J. Choi, et al. Do Not Trust Licenses You See: Dataset Compliance Requires Massive-Scale AI-Powered Lifecycle Tracing. arXiv:2503.02784, 2025.
24. [Kocetkov2022] D. Kocetkov, R. Li, L. Ben Allal, J. Li, et al. The Stack: 3 TB of permissively licensed source code. arXiv:2211.15533, 2022.
25. [Kodak] R. Franzen (host). Kodak Lossless True Color Image Suite. https://r0k.us/graphics/kodak/
26. [Kryukov2020] K. Kryukov, M. T. Ueda, S. Nakagawa, T. Imanishi. Sequence Compression Benchmark (SCB) database -- A comprehensive evaluation of reference-free compressors for FASTA-formatted sequences. GigaScience 9(7), 2020. doi:10.1093/gigascience/giaa072
27. [Longpre2024] S. Longpre, R. Mahari, A. Chen, N. Obeng-Marnu, et al. A large-scale audit of dataset licensing and attribution in AI. Nature Machine Intelligence 6(8):975-987, 2024. doi:10.1038/s42256-024-00878-8

28. [LTCB] M. Mahoney. Large Text Compression Benchmark; About the test data. https://www.mattmahoney.net/dc/text.html; https://www.mattmahoney.net/dc/textdata.html
29. [Missier2013] P. Missier, K. Belhajjame, J. Cheney. The W3C PROV family of specifications for modelling provenance metadata. In Proc. EDBT 2013, pp. 773-776, 2013. doi:10.1145/2452376.2452478
30. [PizzaChili] P. Ferragina, G. Navarro. Pizza&Chili corpus: compressed indexes and their testbeds, text collection. https://pizzachili.dcc.uchile.cl/texts.html
31. [PolyHaven] Poly Haven. License. https://polyhaven.com/license
32. [Pushkarna2022] M. Pushkarna, A. Zaldivar, O. Kjartansson. Data Cards: Purposeful and Transparent Dataset Documentation for Responsible AI. In Proc. FAccT 2022, pp. 1776-1826, 2022. doi:10.1145/3531146.3533231
33. [PythonZipfile] Python Software Foundation. zipfile -- Work with ZIP archives. Python 3.14 documentation. https://docs.python.org/3.14/library/zipfile.html
34. [Rajbahadur2021] G. K. Rajbahadur, E. Tuck, L. Zi, D. Lin, et al. Can I use this publicly available dataset to build commercial AI software? -- A Case Study on Publicly Available Image Datasets. arXiv:2111.02374, 2021.
35. [REUSE] Free Software Foundation Europe. REUSE Specification. https://reuse.software/spec/
36. [Rousseau2020] G. Rousseau, R. Di Cosmo, S. Zacchiroli. Software provenance tracking at the scale of public source code. Empirical Software Engineering 25(4):2930-2959, 2020. doi:10.1007/s10664-020-09828-5
37. [Silesia] S. Deorowicz. Silesia compression corpus. https://sun.aei.polsl.pl/~sdeor/index.php?page=silesia
38. [SPDX] The Linux Foundation. SPDX License List. https://spdx.org/licenses/
39. [Squash] E. Nemerson. Squash Compression Corpus (proposal). https://github.com/nemequ/squash-corpus
40. [Wilkinson2016] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, et al. The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data 3:160018, 2016. doi:10.1038/sdata.2016.18
41. [Wilkinson2026] S. R. Wilkinson, P. Shpilker, W. Brewer. SetGo: Metadata Readiness for Scientific AI Datasets. arXiv:2607.22677, 2026.
42. [Zhao2020] K. Zhao, S. Di, X. Lian, S. Li, D. Tao, J. Bessac, Z. Chen, F. Cappello. SDRBench: Scientific Data Reduction Benchmark for Lossy Compressors. In Proc. IEEE BigData 2020, pp. 2716-2724, 2020. doi:10.1109/BigData50022.2020.9378449
43. [ZipAppnote] PKWARE Inc. APPNOTE.TXT -- .ZIP File Format Specification (sections on "version made by" and the Zip64 extended information extra field). https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT
44. [ZlibngCorpora] zlib-ng. corpora (README). https://github.com/zlib-ng/corpora