

The Truth Contract: Pre-commit Verification as Research Integrity Infrastructure for Independent ML Researchers

Chen, Ho Yiing¹

¹Independent researcher, charenix.com, ORCID: 0009-0006-6816-9891
abbychen981008@gmail.com

2026-05-21

Abstract

Independent ML researchers do not get peer review, adversarial collaborators, or institutional ethics boards. The gap shows up as inflated claims, hallucinated citations, and silent rewrites of past failure. I describe a lightweight integrity layer for this setting, the **Truth Contract**: a project-level `S0UL.md` document declaring forbidden phrasings, a pre-commit hook `truth-lint` that scans diffs and commit messages for violations, and an auto-citation verifier that blocks publish until every bibliography entry resolves against a primary source. I describe the Mercury project's 24-hour sprint as a case study, including a real failure mode where two preprints went to permanent DOIs with bibliography errors that the verifier would have caught had it been wired in at the start. The Truth Contract did not prevent the failure; it caught the failure within hours and forced a corrigendum. That is what I claim it is for.

1 The problem

The institutional researcher writes a paper, sends it out to two anonymous reviewers, and gets back a sheet of objections that sometimes includes “citation 17 does not exist” or “you cannot say ‘achieves state of the art’ without showing the comparison.” The independent researcher writing the same paper at home has none of that. The error goes to the arXiv, the Zenodo, the company blog, and stays.

Two specific failure modes account for most of the visible damage:

1. **Hallucinated or sloppy citations.** Authors cite work they have not read. Author lists, years, venues, and DOIs are wrong. When the writing is assisted by a language model, the rate of these errors rises sharply, because language models confidently produce plausible-looking citations that do not exist.
2. **Inflated claims that survive into the published record.** A bench number is described as “achieves” something it does not. A feature is described as “deployed” when it is in a branch. A failure is described as “mixed results” rather than as a failure. None of these are lies in the strict sense; all of them rewrite reality.

The institutional defence against both is the reviewer. The independent researcher needs a substitute. This paper describes the one I built for the Mercury project [1, 2, 3].

2 Mechanism

The Truth Contract has three pieces.

2.1 SOUL.md

A markdown file at the project root, kept short. It lists forbidden phrases, each with a reason. Sample entries from the Mercury project’s `SOUL.md`:

```
- "deployed" -- only allowed if the URL returns 200
- "achieves" -- requires a comparison table in the same commit
- "we trained" -- forbidden in this project (no training is done)
- "state of the art" -- forbidden, period
- "hallucinated" -- if you mean the model produced wrong output, say so
```

The list is project-specific. The Mercury list grew over the sprint as new failure modes appeared.

2.2 truth-lint hook

A git pre-commit hook scans the staged diff and the commit message for any phrase listed in `SOUL.md`. A hit blocks the commit and prints which phrase, which line. The author either rewrites or, explicitly, adds the phrase to an exemption list in the same commit. Either path is recorded.

The hook is roughly thirty lines of Python. This is not the technical novelty of this paper. The novelty is that the hook exists.

2.3 Auto-citation verifier

For papers in particular, a separate script `paper-check.py` runs eight rules over a `.tex` file before publish. Rule 8 walks every `\bibitem` entry, extracts any DOI or URL, and resolves it against a primary source: CrossRef [4], DataCite [5], ArXiv, or, for software references, the canonical `CITATION.cff` or BibTeX block in the GitHub README. The script compares the canonical title, first-author surname, and publication year against what the bibliography entry says. A mismatch blocks the publish step. In shape, the verifier and the pre-commit ecosystem [6] share the same shape: cheap automated checks at a chokepoint where humans are about to do something irreversible. Concretely, the citation pass blocks publish unless

$$\forall b \in \mathcal{B} : \quad \text{verify}(b) \in \{ \text{ok}, \text{exempt} \},$$

where exempt covers entries explicitly marked “in preparation” or “forthcoming”.

3 Case study: the Mercury 24-hour sprint

Between 2026-05-19 and 2026-05-21 I built the Mercury observability pipeline, drafted five preprints, and shipped two of them to Zenodo with permanent DOIs. The Truth Contract was wired in for code (`truth-lint` blocked four commits during the sprint) but *not* for citations until late on day two.

The result is a clean failure case for this paper. Both v1 preprints (Mercury-Viewer [2], Mercury method [1]) went to permanent DOIs carrying two bibliography errors I introduced by writing the references list from memory without primary-source verification. Specifically: a third-party tool was cited with an incorrect publication year (off by one) and a self-reference used an inconsistent author name. Both errors are minor on the scale of academic misconduct; neither would be visible to a casual reader; both are exactly the type of slip the Truth Contract claims to prevent.

Once added to `paper-check.py`, caught both errors on the first invocation. Catching them required:

- Fetching the third-party tool’s GitHub README, parsing the `@misc` BibTeX block, comparing the canonical year and author list to the bibitem.
- Resolving every DOI against CrossRef or DataCite and checking that the title returned matched the title written.

Within four hours of the error being detected, both v1 records were superseded by v2 corrigenda. The v1 records remain permanent (Zenodo does not allow deletion), but the v1 pages now display a banner

pointing to v2 as the canonical version. This is the standard academic erratum workflow, and it cost the project nothing except embarrassment.

The lesson is small but exact: *the Truth Contract does not prevent failure; it prevents persistent failure*. The window between commit and discovery is what closes.

4 Costs and benefits

The cost side is genuinely small. Let c_{contract} be the total lines of code in the system:

$$c_{\text{contract}} = c_{\text{SOUL}} + c_{\text{hook}} + c_{\text{verifier}} \lesssim 200 + 50 + 500 = 750.$$

The per-publish runtime for a paper with $|\mathcal{B}|$ bibliography entries is

$$T_{\text{verify}} \approx |\mathcal{B}| \cdot t_{\text{api}}, \quad t_{\text{api}} \in [0.3, 0.8] \text{ s},$$

dominated by network latency to CrossRef, DataCite, and GitHub raw endpoints. For $|\mathcal{B}| \leq 30$, this puts the total publish-time check well under 30 seconds.

The benefit side is harder to measure but easy to motivate. Inflated claims and hallucinated citations are the single most common reason independent ML preprints lose credibility. Catching them at the publish step, before the DOI is minted, is orders of magnitude cheaper than catching them as a comment on the public record.

I am not claiming the Truth Contract is a substitute for peer review. My claim: this is the cheapest non-trivial integrity layer I know how to install for someone working alone. It would not have caught a fabricated bench result. It does catch the everyday small dishonesties of solo writing, especially when the writing is co-produced with a language model that produces plausible text without producing true text.

5 Limitations

The verifier is only as good as the canonical sources it consults. CrossRef does not index Zenodo DOIs; DataCite does, but slower. Some references (older preprints, personal websites without metadata) fall through to a “needs manual” bucket, which the human author then has to confirm by hand. The script flags but does not auto-fix.

The forbidden-phrase list is brittle. A clever rewrite of an inflated claim that avoids the exact wording in `SOUL.md` will pass the hook. The hook is a tripwire, not an oracle.

Finally, the case study is a single project, a single sprint, and a single author. The numbers in section 3 are anecdotes, not evidence. The framework is offered as a small piece of infrastructure to be picked up and modified, not as a proof of efficacy.

6 Conclusion

For someone writing ML papers alone, the difference between a clean record and a stained one is often not whether they made mistakes (they did, I did) but whether the mistakes were caught fast enough to land as “corrigendum within hours” instead of “error on the permanent record for years.”

The Truth Contract is a 700-line lever for closing that window. It is cheap, it is auditable, and it works the same way the rest of computational science increasingly works [6]: cheap automated checks at chokepoints. I am releasing the Mercury project’s `SOUL.md`, `truth-lint`, and `paper-check.py` alongside this paper for anyone who wants to copy the pattern.

Acknowledgements

Drafted in continuation of the Mercury sprint, in collaboration with an Anthropic Claude Code session as pair-engineer. The failure case in section 3 is a real one and is described here without embellishment.

References

- [1] Chen, Ho Yiing. Mercury: An addressable million-cell sensor grid for browser-accessible LLM observability on consumer hardware. Zenodo, 2026. DOI: 10.5281/zenodo.20313154.
- [2] Chen, Ho Yiing. Mercury-Viewer: A self-contained browser-based interactive 3D visualization of LLM internal sensor maps. Zenodo, 2026. DOI: 10.5281/zenodo.20313150.
- [3] Chen, Ho Yiing. Unsupervised discovery of language and topic lanes in transformer models via multilingual co-firing signatures. In preparation, 2026.
- [4] Hendricks, G., Tkaczyk, D., Lin, J., and Feeney, P. Crossref: The sustainable source of community-owned scholarly metadata. *Quantitative Science Studies*, 1(1):414–427, 2020. DOI: 10.1162/qss_a_00022.
- [5] Brase, J. DataCite – A global registration agency for research data. In *Fourth International Conference on Cooperation and Promotion of Information Resources in Science and Technology*, pages 257–261, 2009. DOI: 10.1109/COINFO.2009.66.
- [6] Yelp et al. pre-commit: A framework for managing and maintaining multi-language pre-commit hooks. <https://pre-commit.com>, 2014–present.