The Second Bridge on Artificial Intelligence for Scholarly Communication (AAAI-26)

Contributions

https://doi.org/10.52825/ocp.v8i.3174

Published: 18 Dec. 2025

Automated Scientific Narrative Generation Through Computational Provenance and Dynamic Authoring Frameworks

Augustus Ellerm[1,*], Benjamin Adams[2]
Mark Gahegan[1]

[1]University of Auckland, Auckland, New Zealand

[2]University of Canterbury, Christchurch, New Zealand

*Correspondence: Augustus Ellerm, gus.ellerm@auckland.ac.nz

Abstract. Despite the growing quantity of born-digital research, scholarly articles remain tethered to flattened PDFs—unable to expose or verify the computations they summarize. We present a publication container that couples provenance-generating eScience infrastructures with Dynamic Authoring Frameworks (DAFs). Using this method, an article is modelled as a set of symbolic operations over provenance. The result is a procedural narrative whose claims are anchored to deterministic combinations of operations and provenance, enabling verification. To enhance this method we place LLM inference tasks inside the DAF for small, constrained tasks. This mitigates hallucination, supports granular attribution, and enables researchers to move "warrant" making away from LLM inference, to the explicit operations within the DAF. Using a remote-sensing case study (CoastSat), we show how this method can produce consistent, accurate, and generative methodological descriptions from provenance. We argue that for modern scholarly communication to support generative text, it must move beyond "flat" scholarly articles towards more formal representations of authorship.

Keywords: Scholarly Publishing, Large Language Models, Dynamic Authoring Framework, Provenance

1. Introduction

Despite repeated calls for innovation [1], [2], [3], publishers still default to *flattened* views of research—typically PDFs—that compress born-digital research (research that is performed entirely within digital environments) into a single narrative artifact. Three decades ago, communication over the internet was expected to transform scholarly publishing, not merely reproduce print online. As Harnad wrote:

> ... the most important factor in hastening the onset of the fourth cognitive revolution will surely be the unique capabilities of the medium [networked electronic communication] itself. Electronic journals should not and will not be mere clones of paper journals, ghosts in another medium. [1]

Yet today's scholarly articles remain restricted by print-era conventions, and Harnad's vision of "scholarly skywriting" (a form of agile publishing, similar to micropublication [4]) remains atypical of scholarly communication [5]. Our narrative containers fail to take advantage of the rich, structured information generated by computational research infrastructure, such as data repositories, workflow systems, and provenance tracking. In doing so, they present static representations of research, which quickly go out of date and do not support reproducibility [6], [7].

Large language models (LLMs) may be the catalyst needed to motivate systemic change within our publication containers. Generative text is already detectable within the scientific corpus, with roughly twice the number of article submissions in 2023 likely to contain LLM generated text than those from 2021 [8]. However, unconstrained generative text is at odds with the epistemic standards of science and the scholarly record [9], [10], [11]. Hallucination [12], bias [13], attribution [14], and the non-determinism [15] of these models undermine trust in science communication. In order to apply LLMs responsibly, modern publication containers must supplement the weaknesses of generative text with additional information, enabling readers to verify their understanding of reported results. We do not attempt to remove the stochastic nature of LLM generation itself; rather, we propose coupling **provenance-generating** eScience infrastructure with **Dynamic Authoring Frameworks** (DAFs) in order to express narratives as symbolic operations over an experiment. In doing so, we capture and embed part of the external interpretative logic used by authors when writing articles within the DAF, and couple LLM inference tasks tightly to local contexts—verifiable against resolvable entities within the provenance record.

2. Provenance and eScience Technologies

As eScience has advanced, digital research is performed across modular platforms: scientific gateways [16] and virtual research environments [17] that centralize access to tools and data; literate programming environments that integrate code and narrative [18]; workflow managers that orchestrate multi-step analyses; and containerized environments deployed on cloud and HPC infrastructure [19]. Using these systems leaves a *digital footprint*—provenance of what was done, with which inputs, and by whom [20]. Some of this information is captured implicitly as part of the platforms' operation (e.g., UUIDs, job IDs, timestamps), and some is captured explicitly to improve transparency and reproducibility (e.g., structured metadata and PIDs).

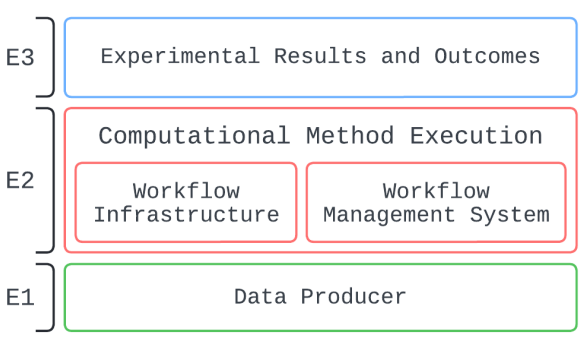
Workflow Management Systems (WMSs) [21] are the most suitable technology for provenance-aware eScience infrastructures. Their declarative workflow definitions (directed acyclic graphs of tasks with explicit dependencies) define a machine-readable record of the method's structure, parameters, and dataflow—a form of *prospective* provenance. They modularize complex methods, providing step-level provenance records of instantiated parameters and inputs/outputs—a form of *retrospective* provenance [22].

With provenance, WMSs support reproducibility/reuse, auditability, and debugging/optimization within computational research workflows. This value has motivated developments in provenance standards and tools across common ecosystems (e.g., Nextflow [23]; Toil [24]; Arvados [25]) improving the accessibility and interoperability of these records. As a result of these advancements, detailed provenance information is increasingly available, providing an interface for publication containers to couple narrative claims to the science performed. WMSs represent one mature provenance-aware technology in use by researchers today. However, they are only one aspect of born-digital

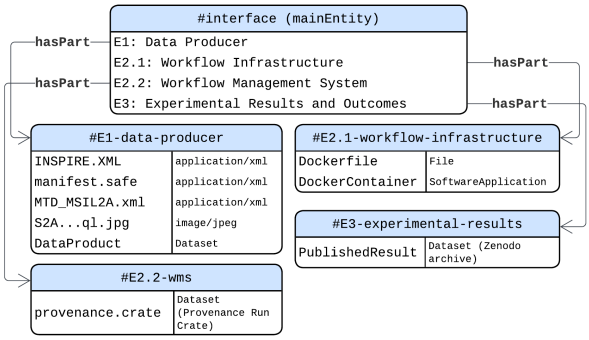
research. When conceptualizing the provenance of an experiment, we must also consider the origin of data and the storage of results. Towards that end, we introduce a minimal provenance-aware Experiment Infrastructure model.

2.1 Provenance-aware Experiment Infrastructures

We model the Experiment Infrastructure as a sufficient and generalizable pattern for computational research (Figure 1a): **E1** Data Producer (structured data inputs with identifiable sources and metadata), **E2** Computational Method Execution (method execution with structured provenance capture), and **E3** Experimental Results and Outcomes (archival outputs supporting traceability and reuse). For each execution of an experiment, these layers produce provenance records that we aggregate into an `interface.crate`—a linked-data Research Object (RO) whose parts correspond to each layer. The model is schema-agnostic across layers, but assumes that the WMS exports step-level provenance (e.g., that of the Provenance Run Crate schema [26]).



***(a) E1**: structured inputs with identifiers and metadata. **E2**: WMS infrastructure with step-level provenance capture. **E3**: results archival with identifiers and metadata.*



***(b)** An example interface RO-Crate, aggregating layers of provenance metadata derived from the Experiment Infrastructure. Find an interactive representation of this crate here.*

***Figure 1.** Structuring the provenance of an Experiment Infrastructure.*

Figure 1b shows a simplified `interface.crate` generated for a remote-sensing experiment. For clarity, an interactable version of this crate can be explored here, and may be downloaded here. The WMS step-level provenance (E2.2), generated by CWLProv [27], is embedded within the `interface-crate` and is similarly interactable here. The resulting crate is an automated portable record of **what ran**, **with what**, **and where**.

3. Dynamic Authoring Framework

We introduce the idea here of a Dynamic Authoring Framework (DAF) which compiles operations over this crate's graph (i.e., information on steps, data, parameters, environments) and resolves an article to one of many interpretative states. DAFs consist of two components: a Document Schema, and a set of Document Operations.

The Document Schema is a structured, machine-readable model over which operations modify narrative content. It specifies the elements an article may contain (e.g., sections, paragraphs, sentences), their hierarchical and semantic relations, and how these elements are operated upon by procedural logic. The instantiated schema results in a document graph $D = (N, A)$ of nodes N and hierarchy edges A. Nodes are either narrative elements E (authored narrative content) or operation nodes O. The Document Schema must provide (1) embeddings for operations (O), (2) stable identifiers

of narrative elements (E) for addressability, and (3) at least paragraph-level granularity. Paragraphs are identified as the minimum granularity for these models, as they reflect a cognitive and rhetorical unit of thought in academic writing [28], [29]. Each operation $o \in O$ is described as a tuple (q_0, ϕ_0, τ_0): a query q_0 over the `interface.crate` returning bindings, a condition ϕ_0 (e.g., if/else, for, goto) evaluated on those bindings, and a target set of narrative elements $\tau_0 \subseteq N$ whose inclusion is dependent on the value of the condition.

The novelty of a DAF does not lie in inventing new control structures (i.e., conditions that o may take), but in formalizing the relationship between an experiment (`interface.crate`) and a set of narrative outcomes. By doing so, a DAF provides a formal description of the workflow and the computational setting(s) used. But it can do more than this: it can also define the possible interpretations an article may take, and in doing so define the **warrants**—in Toulmin's sense [30], the rule that *licenses movement from data to claim*—of a narrative claim. For example, a DAF could define rules that determine the narrative's perspective, dependent on experimental results—e.g., a DAF Operation in the form: `if result in range a-b`. This defines a warrant for the subseqent narrative claim: "This result is in the high range and indicates a significant deviation from the norm." In this sense, a DAF can also capture some of the inferential logic that researchers use to interpret or contextualise their results. It embeds multiple narrative possibilities within a single container—a framework that we take advantage of to enable trustworthy LLM inference. Within this paper we use Stencila [31] as the DAF—leveraging its rich document model and associated operations to procedurally edit an article dependent on an `interface.crate`. Find an example of an authored DAF document here, its JSON serialisation here, and a compiled instance here.

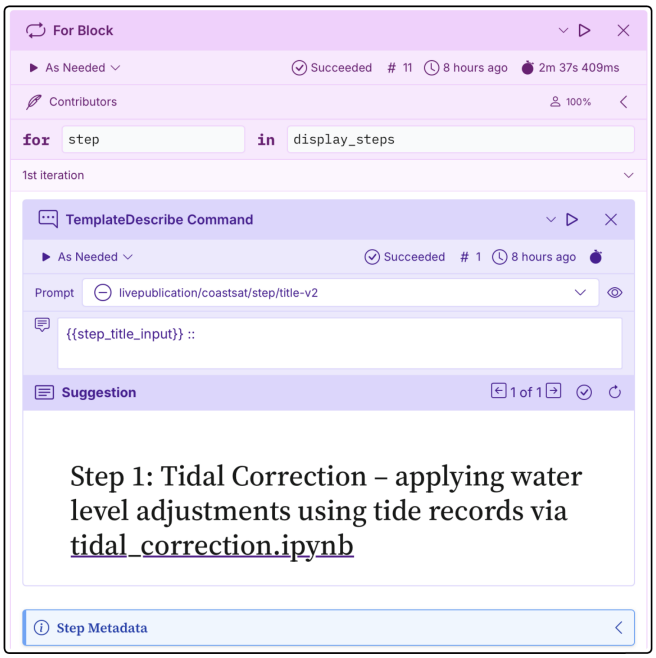
4. Applying LLM inference to Dynamic Authoring Frameworks

As an example, we apply LLM inference within a DAF to generate a *provenance-dependent* methodological narrative for a coastal shoreline analysis experiment (CoastSat). The CoastSat experiment is represented by a previously published `interface.crate`—a linked-data RO-Crate that aggregates the provenance information generated by CoastSat when it executes. This object is used as a portable, queryable index of CoastSat's most recent execution. An interactive version of this `interface.crate` can be explored here.

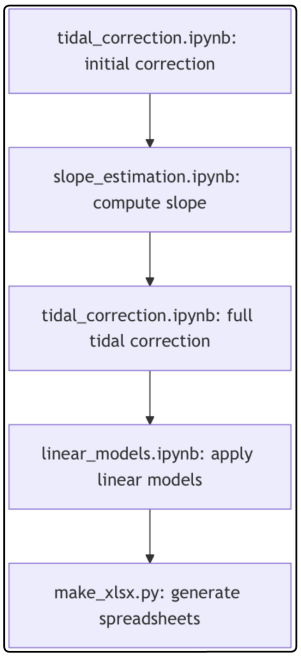
By combining the `interface.crate` with a simple DAF, we procedurally generate a description of the experimental processes using an LLM, working in small, constrained steps. Each generated fragment (titles, objectives, operations, inputs/outputs) is grounded in crate entities and linked for validation. The example provided in this paper produces:

- **Step-title Generation** (Figure 2a): Titles that encode links to the versioned source code used during the execution of that step.
- **Workflow overview and Diagram** (Figure 2b): A description of the CoastSat workflow itself, and a generative Mermaid diagram for a high-level overview.
- **Objectives** (Figure 2c): A set of objectives inferred from each step, with citations to source code cell blocks for traceability.
- **Input/Output Parameter Descriptions** (Figure 2d): A description of each step's parameters, with links to example artifacts produced during the last CoastSat execution.

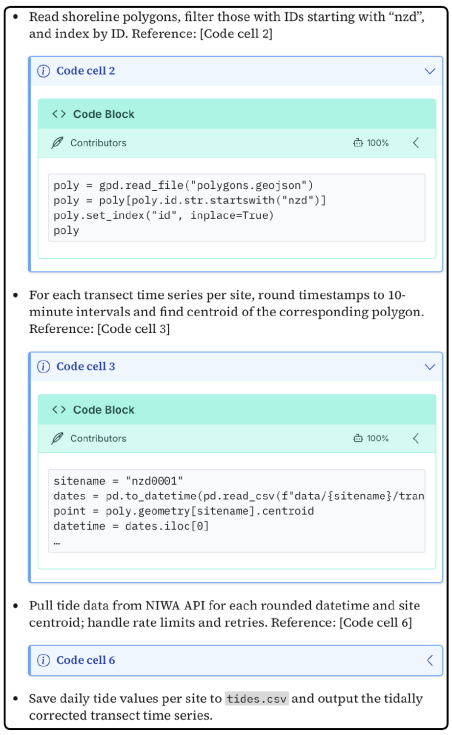
The following Stencila snippet (Listing 1) illustrates how we can procedurally build methodological descriptions by stepping through provenance entities and performing these small inference tasks. Each inference consumes a restricted context window built from the `interface.crate`. The result is a generative methodological description that directly references the steps represented within provenance and enumerates on their function and parameter settings (Figure 3a).



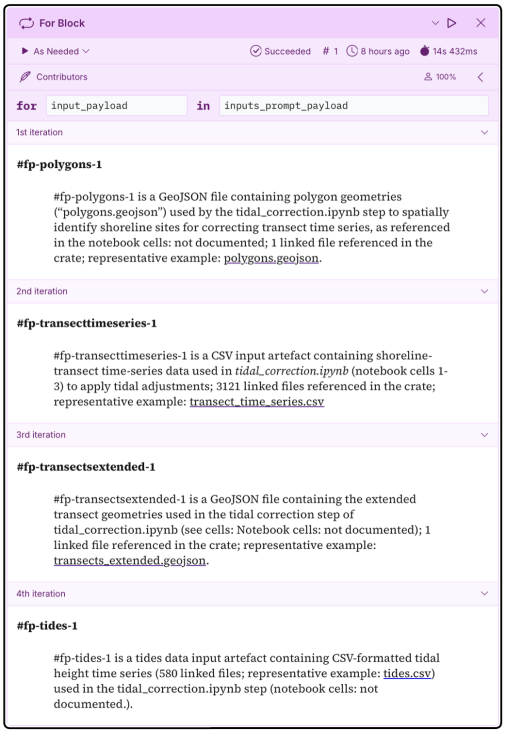
a. *Procedural step-title generation. For each step we provide the step's metadata, producing consistent numbered headings that link back to the executed notebook (prompt context).*



b. *Generative mermaid flowchart. Describes the notebooks and scripts executed for the shoreline workflow (prompt context).*

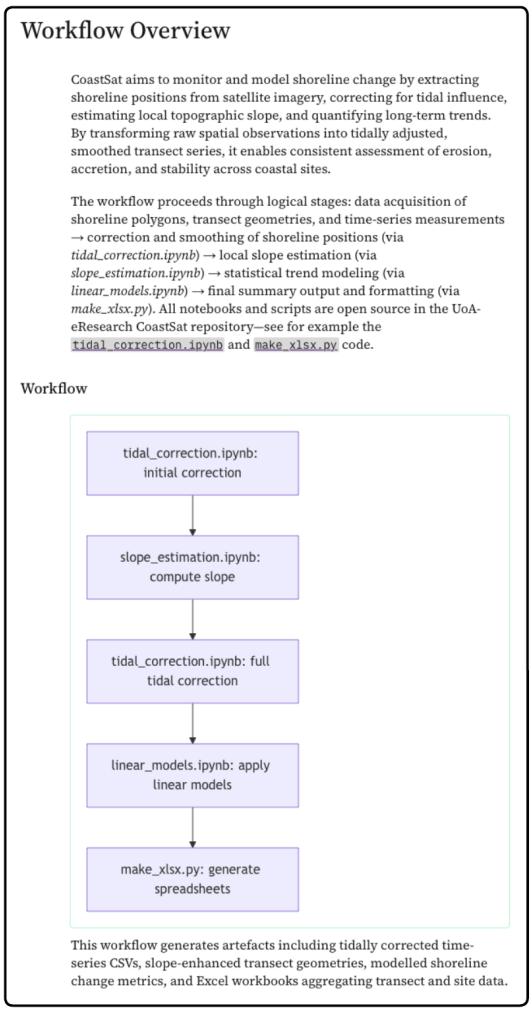


c. *Generative step objectives and goals. The content cites supporting notebook cells to enable readers to trace the method in context (prompt context).*



d. *Enumerated step inputs. Each description links back to the corresponding artifact in the CoastSat provenance (prompt context).*

Figure 2. *Generative methodology description derived from experiment provenance (interface.crate) and Dynamic Authoring Framework (template). (a) Procedural title generation. (b) Generative Mermaid diagram. (c) Step objective descriptions. (d) Enumerated input descriptions. View the entire description here.*



Workflow Overview

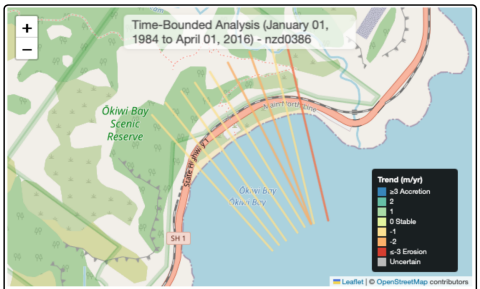
CoastSat aims to monitor and model shoreline change by extracting shoreline positions from satellite imagery, correcting for tidal influence, estimating local topographic slope, and quantifying long-term trends. By transforming raw spatial observations into tidally adjusted, smoothed transect series, it enables consistent assessment of erosion, accretion, and stability across coastal sites.

The workflow proceeds through logical stages: data acquisition of shoreline polygons, transect geometries, and time-series measurements → correction and smoothing of shoreline positions (via *tidal_correction.ipynb*) → local slope estimation (via *slope_estimation.ipynb*) → statistical trend modeling (via *linear_models.ipynb*) → final summary output and formatting (via *make_xlsx.py*). All notebooks and scripts are open source in the UoA-eResearch CoastSat repository—see for example the `tidal_correction.ipynb` and `make_xlsx.py` code.

Workflow

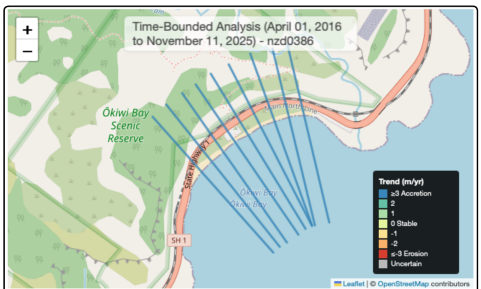
This workflow generates artefacts including tidally corrected time-series CSVs, slope-enhanced transect geometries, modelled shoreline change metrics, and Excel workbooks aggregating transect and site data.

(a) *The resulting authored description of the CoastSat workflow, as it might appear in a publication, automatically generated from the* `interface.crate` *by the Dynamic Authoring Framework.*



The time-bounded assessment for site nzd0386 — Canterbury, New Zealand / Aotearoa spans July 1999 to approximately March 2016, covering about 16.7 years of observations. Nine transects contributed 1,791 valid shoreline measurements, averaging 199.0 points per transect; any missing counts are reported as unknown. Mean shoreline change is −0.55 m/yr with values ranging from −1.23 to −0.38 m/yr, indicating consistent erosion across the site. Model agreement is low, with an average R^2 of 0.07, suggesting that the estimated trends have limited confidence. Overall, the shoreline exhibits marked retreat throughout the analysed period, pointing to stability concerns for coastal landforms in this region.

(b) *Generated prose of a pre-earthquake Kaikōura coastline*



The time-bounded assessment for site nzd0386 — Canterbury, New Zealand / Aotearoa spans April 01, 2016 to November 11, 2025, covering roughly 9.4 years of observations. Nine transects contributed 1,472 valid shoreline measurements, averaging 163.6 points per transect. Mean shoreline change is +8.53 m/yr with values ranging from +6.22 to +10.00 m/yr, indicating consistent accretion across the site. Model agreement is moderate, with an average R^2 of 0.38. Overall, the shoreline exhibits stable accumulation at significant rates during the analysed period.

(c) *Generated prose of post-earthquake Kaikōura coastline*

Figure 3. *Example outputs of the LLM-enabled DAF applied to CoastSat: (a) a snippet of the generative methodology, (b-c) example comparison between two time-delimited CoastSat provenance records.*

```
### Step Inputs
:: if inputs_prompt_payload
::: for input_payload in inputs_prompt_payload
**{{input_payload["parameter"]}}**
:::: template_describe @livepublication/coastsat/step/input-v2 [openai/gpt-5] ↩
    ↪ {{input_payload}} ::::
:::
::
### Step Outputs
:: if outputs_prompt_payload
::: for output_payload in outputs_prompt_payload
**{{output_payload["parameter"]}}**
:::: template_describe @livepublication/coastsat/step/output-v2 [openai/gpt-5] ↩
    ↪ {{output_payload}} ::::
:::
::
```

Listing 1. *LLM inference loop extract from coastsat_llm.smd illustrating how provenance-derived payloads drive generative summaries. The `for` blocks iterate over provenance representations of CoastSat's steps inputs and outputs, passing curated artifact metadata to the inference step.*

```
::: if (site_data.n_points_nonan > min_points_for_valid_regression) and ↩
    ↪ ((abs(site_data.trend) > trend_threshold)) and (site_data.rmse < ↩
    ↪ noise_threshold)
:::: template_describe @livepublication/coastsat/time/positive-findings-v2 ↩
    ↪ [openai/gpt-5] {{meta_data}} ::::
:::
```

Listing 2. *Narrative classification example extracted from coastsat_results_llm.smd. The `if` block defines a narrative warrant for a specific conclusion.*

Finally, we apply this method to the "Results" section of the CoastSat experiment (rather than the methods), and by doing so, can explore changes in CoastSats result's over time. Here, we generate two versions of the `interface.crate` using time-bounded datasets resulting in two different reports. Figures 3b and 3c illustrate how, by taking advantage of this method, we can express change within the generative narrative. This example highlights the effect on the coastline of the 2016 Kaikōura earthquake (in New Zealand). This magnitude 7.8 earthquake uplifted sections of the seabed by up to 8 meters, dramatically altering the shoreline [32]. By embedding simple classifications within the DAF (listing 2), we model a simple **warrant** regarding accretion/erosion narrative cases. The resulting changes in the generated description of the shoreline assessment (both text and images) can be seen in Figure 3b and 3c. Using this approach, we have more control over the consequent inference, allowing us to closely govern the LLM's role in authoring scholarly communication.

During an inference task (see `template_describe` calls in Listing 1), Stencila builds a structured prompt from (1) a versioned slug (e.g. @livepublication/coastsat/step/output-v2) which resolves to a reusable prompt template that embeds a shared CoastSat context and specifies the expected form of the output, and (2) a selection of provenance data derived from the `interface.crate`. The composed prompt is then sent to a pinned LLM for inference. GPT-5 was the model for this example; however, smaller, more efficient models may better suit the inference strategy described in this paper. The project source code that generated this example can be found in this repository.

5. Discussion and Conclusion

By treating an `interface.crate` as the machine-readable *ground truth* of an experiment and the DAF as the procedural narrative layer, we separate the **warrant** from the **words** (narrative inference) describing that claim. This enables three primary benefits:

- **Verification and Attribution**: Each generated claim is predicated on a combination of deterministic DAF operations (O) and the provenance context. Readers may cross-reference prose and provenance to verify statements and attribute them to concrete artifacts.
- **Hallucination Reduction**: Hallucination remains a challenge for LLMs [33]. Prior work establishes that constraining contexts and using RAG methods reduces the likelihood of hallucination during inference [34], [35], [36]. Our approach applies inference over small, highly constrained contexts designed to reduce the likelihood of unsupported claims. The codebase used in this paper is open, and we encourage researchers to re-execute the pipeline to form an opinion on consistency. Anecdotally, we have not experienced any hallucination issues in our experiments thus far. However, we have not conducted a controlled evaluation; demonstrating empirical reduction in hallucination is future work.
- **Reproducibility and Stability**: By moving warrant-bearing decisions into the DAF and away from an LLM, we define a paper's *admissible* claims. For a given experiment's `interface.crate` and DAF, any regeneration is licensed only to those claims that the DAF warrants. While different runs may still vary in wording (and thus how a claim is interpreted), we introduce the ability to audit this wording against explicit links between prose and provenance. This makes DAF-generated narratives a more reliable source of truth, arguably more reliable than relying on the author's memory and technical understanding.

The example presented in this paper generates methodological prose: DAF operations select steps, parameters, and artifacts to be expressed, and inference via LLM supplies the words. In a separate experiment focused on using the framework to report on dynamic systems, we developed a results-focused account based on the CoastSat example above, but concentrating on outcome warrants. In that experiment, the DAF operations apply inclusion/exclusion criteria over trend estimates and other factors drawn from the `interface.crate`. And while this does not invoke an LLM for text generation, the same **warrants**→**words** pattern applies; once the DAF warrants a conclusion, an LLM can verbalize it over a constrained context.

Finally, a strength of this approach is that the `interface.crate` can be versioned. Each new execution of an experiment produces a new `interface.crate` reflecting the current data, parameters, and infrastructure used for the experiment's run. Re-compiling the DAF against this crate yields a new narrative with updated **warrants**, and, where the DAF operations license them, updated **conclusions**. This means that a report (or even a 'live' research article) can be automatically updated as experiments are updated or modified. Automating this regeneration reduces the opportunity for errors and inconsistencies to be introduced within the prose, helping to keep reported methods and structured results aligned with the underlying computational processes. Additionally, the difference between two crates can be measured and expressed via the DAF to readers—highlighting changes in methodology or results. This reflects the iterative nature of scientific research and anticipates changes within the underlying computational methodology.

The method presented in this paper makes the publication container genuinely *born-digital*: the `interface.crate` carries an evolving provenance record of an experiment, the DAF translates this record into warranted claims, and the LLM supplies the prose over well-constrained contexts. In doing so, the container moves beyond print-era constraints and towards a live and epistemologically sound record. This opens the door to a new type of publication—a LivePublication [37]—which lives alongside a computational experiment as an evolving narrative representation of scientific research performed. This category of publication reduces manual writing effort for longitudinal studies, maintains alignment between the research narrative and the experimental workflow, and bundles reproducible accounts of both within a single container.

Limitations and Future Work

This is an early, single-domain demonstration: we report structural guarantees of the framework but have not yet run controlled evaluations. Practical concerns remain: (1) maintaining persistent links between provenance and published artifacts, (2) testing generality across provenance models and domains, (3) adressing the additional upfront effort required by authors to define a DAF, and (4) measuring performance and cost. Future work consists of evaluating prose variation across regenerations, replication of the method across different WMSs, and research on the relationship between warrants, DAFs, and generative tasks. We will also publish standardized profiles for `interface.crate`s.

Author contributions

Augustus Ellerm: Conceptualization, Methodology, Software, Investigation, Writing.
Benjamin Adams: Conceptualization, Supervision, Writing - Review and Editing.
Mark Gahegan: Conceptualization, Supervision, Writing - Review and Editing, Funding acquisition.

Competing interests

The authors declare no competing interests.

Funding

The authors are grateful for funding from the New Zealand government via the “Beyond Prediction: explanatory and transparent data science” project supported by the Strategic Science Investment Fund, administered by the Ministry of Business Innovation and Employment, Aotearoa/New Zealand.

References

[1] S. Harnad, “Post-Gutenberg galaxy: The fourth revolution in the means of production of knowledge”, *The Public-Access Computer Systems Review*, vol. 2, no. 1, pp. 39–53, 1991.

[2] P. E. Bourne et al., “Improving the future of research communications and e-scholarship (dagstuhl perspectives workshop 11331)”, *Dagstuhl Manifestos*, vol. 1, no. 1, pp. 41–60, 2012. DOI: 10.4230/DagMan.1.1.41.

[3] D. Shotton, “Semantic publishing: The coming revolution in scientific journal publishing”, *Learn. Publ.*, vol. 22, no. 2, pp. 85–94, 2009. DOI: 10.1087/2009202.

[4] T. Clark, P. N. Ciccarese, and C. A. Goble, "Micropublications: A semantic model for claims, evidence, arguments and annotations in biomedical communications", *J. Biomed. Semantics*, vol. 5, p. 28, 2014. DOI: 10.1186/2041-1480-5-28.

[5] C. Tenopir, D. W. King, L. Christian, and R. Volentine, "Scholarly article seeking, reading, and use: A continuing evolution from print to electronic in the sciences and social sciences", *Learn. Publ.*, vol. 28, no. 2, pp. 93–105, 2015. DOI: 10.1087/20150203.

[6] M. Baker, "1,500 scientists lift the lid on reproducibility", *Nature*, vol. 533, no. 7604, pp. 452–454, 2016. DOI: 10.1038/533452a.

[7] M. D. Wilkinson et al., "The FAIR guiding principles for scientific data management and stewardship", *Scientific Data*, vol. 3, no. 1, p. 160 018, 2016. DOI: 10.1038/sdata.2016.18.

[8] F. M. Howard, A. Li, M. F. Riffon, E. Garrett-Mayer, and A. T. Pearson, "Characterizing the increase in artificial intelligence content detection in oncology scientific abstracts from 2021 to 2023", *JCO Clin. Cancer Inform.*, vol. 8, no. 8, e2400077, 2024. DOI: 10.1200/CCI.24.00077.

[9] Nature, "Tools such as ChatGPT threaten transparent science; here are our ground rules for their use", *Nature*, vol. 613, no. 7945, p. 612, 2023. DOI: 10.1038/d41586-023-00191-1.

[10] International Committee of Medical Journal Editors, "Recommendations for the conduct, reporting, editing, and publication of scholarly work in medical journals", *Chin. Nurs. Res.*, vol. 2, no. 4, pp. I–XIII, 2015.

[11] C. Zielinski et al., "WAME recommendations on ChatGPT and chatbots in relation to scholarly publications", *Natl. Med. J. India*, vol. 36, no. 1, pp. 1–4, 2023. DOI: 10.25259/NMJI_365_23.

[12] L. Huang et al., "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions", *ACM Trans. Inf. Syst.*, vol. 43, no. 2, 46:1–46:55, 2025. DOI: 10.1145/3703155.

[13] X. Wei, N. Kumar, and H. Zhang, "Addressing bias in generative AI: Challenges and research opportunities in information management", *Information & Management*, vol. 62, no. 2, p. 104 103, 2025. DOI: 10.1016/j.im.2025.104103.

[14] B. Huang, C. Chen, and K. Shu, "Authorship attribution in the era of LLMs: Problems, methodologies, and challenges", *ACM SIGKDD Explorations Newsletter*, vol. 26, no. 2, pp. 21–43, 2024. DOI: 10.1145/3715073.3715076.

[15] Y. Song, G. Wang, S. Li, and B. Y. Lin, "The good, the bad, and the greedy: Evaluation of LLMs should not ignore non-determinism", in *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Albuquerque, NM, USA: Association for Computational Linguistics, 2025, pp. 4195–4206. DOI: 10.18653/v1/2025.naacl-long.211.

[16] N. Wilkins-Diehr, "Special issue: Science gateways—common community interfaces to grid resources", *Concurr. Comput.*, vol. 19, no. 6, pp. 743–749, 2007. DOI: 10.1002/cpe.1098.

[17] L. Candela, D. Castelli, and P. Pagano, "Virtual research environments: An overview and a research agenda", *Data Sci. J.*, vol. 12, GRDI75–GRDI81, 2013. DOI: 10.2481/dsj.GRDI-013.

[18] D. E. Knuth, "Literate programming", *Comput. J.*, vol. 27, no. 2, pp. 97–111, 1984. DOI: 10.1093/comjnl/27.2.97.

[19] M. Alser et al., "Packaging and containerization of computational methods", *Nat. Protoc.*, vol. 19, no. 9, pp. 2529–2539, 2024. DOI: 10.1038/s41596-024-00986-0.

[20] P. Groth and L. Moreau, "PROV-overview: An overview of the PROV family of documents", World Wide Web Consortium, Tech. Rep., 2013, W3C Note 30 April 2013.

[21] C. Goble et al., "FAIR computational workflows", *Data Intelligence*, vol. 2, no. 1-2, pp. 108–121, 2020. DOI: 10.1162/dint_a_00033.

[22] S. B. Davidson and J. Freire, "Provenance and scientific workflows: Challenges and opportunities", in *Proceedings of the 2008 ACM SIGMOD International Conference on*

Management of Data, Vancouver, BC, Canada: ACM, 2008, pp. 1345–1350. DOI: 10.1145/1376616.1376772.

[23] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame, "Nextflow enables reproducible computational workflows", *Nat. Biotechnol.*, vol. 35, no. 4, pp. 316–319, 2017. DOI: 10.1038/nbt.3820.

[24] J. Vivian et al., "Toil enables reproducible, open source, big biomedical data analyses", *Nat. Biotechnol.*, vol. 35, no. 4, pp. 314–316, 2017. DOI: 10.1038/nbt.3772.

[25] Arvados, *Arvados homepage*, https://arvados.org/, Accessed: 2025-06-22.

[26] S. Leo et al., "Recording provenance of workflow runs with RO-crate", *PLoS One*, vol. 19, no. 9, e0309210, 2024. DOI: 10.1371/journal.pone.0309210.

[27] F. Z. Khan, S. Soiland-Reyes, M. R. Crusoe, A. Lonie, and R. Sinnott, "CWLProv: Interoperable retrospective provenance capture and its challenges", in *Proceedings of the 7th International Provenance and Annotation Workshop (IPAW 2018)*, Springer, 2018. DOI: 10.5281/zenodo.1208478.

[28] L. Flower and J. R. Hayes, "A cognitive process theory of writing", *Coll. Compos. Commun.*, vol. 32, no. 4, pp. 365–387, 1981. DOI: 10.2307/356600.

[29] S. P. Witte and L. Faigley, "Coherence, cohesion, and writing quality", *Coll. Compos. Commun.*, vol. 32, no. 2, pp. 189–204, 1981. DOI: 10.2307/356693.

[30] L. Groarke, "Informal logic", in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., Fall 2021 Edition, Metaphysics Research Lab, Stanford University, 2021.

[31] M. Aufreiter, A. Pawlik, and N. Bentley, *Stencila – an office suite for reproducible research*, https://elifesciences.org/labs/c496b8bb/stencila-an-office-suite-for-reproducible-research, Accessed: 2025-09-23, 2018.

[32] R. M. Langridge et al., "Coseismic rupture and preliminary slip estimates for the papatea fault and its role in the 2016 M_W 7.8 kaikōura, new zealand, earthquake", *Bull. Seismol. Soc. Am.*, vol. 108, no. 3B, pp. 1596–1622, 2018. DOI: 10.1785/0120170336.

[33] Z. Xu, S. Jain, and M. Kankanhalli, "Hallucination is inevitable: An innate limitation of large language models", *arXiv [cs.CL]*, 2024. arXiv: 2401.11817 [cs.CL].

[34] N. F. Liu et al., "Lost in the middle: How language models use long contexts", *Trans. Assoc. Comput. Linguist.*, vol. 12, pp. 157–173, 2024. DOI: 10.1162/tacl_a_00638.

[35] H. Jiang, Q. Wu, C.-Y. Lin, Y. Yang, and L. Qiu, "LLMLingua: Compressing prompts for accelerated inference of large language models", in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, 2023, pp. 13 358–13 376. DOI: 10.18653/v1/2023.emnlp-main.825.

[36] Z. Ji et al., "Survey of hallucination in natural language generation", *ACM Comput. Surv.*, vol. 55, no. 12, 2023, ISSN: 0360-0300. DOI: 10.1145/3571730.

[37] A. Ellerm, M. Gahegan, and B. Adams, "LivePublication: The science workflow creates and updates the publication", in *2023 IEEE 19th International Conference on e-Science (e-Science)*, IEEE, 2023, pp. 1–10. DOI: 10.1109/e-Science58273.2023.10254857.