Superscientist: A Durability Contract for AI-Driven Computational Physical Science

Anonymous Authors[1]

Abstract

Autonomous AI scientists in HPC labs require what we call a *durability contract*: a machine-verifiable, versioned, human-signed agreement over *what* is computed, *how* it is verified, and *which* classes of change require re-approval. Existing harnesses optimize for tight, low-stakes loops — minutes-long ideation, sandbox benchmarks, small reversible experiments — but computational physical science breaks every premise of that design: a single molecular-dynamics stage runs hours to days on shared queues, campaigns span sessions and human handoffs, and a late-caught method error silently discards days of compute.

Superscientist realizes the contract via three pillars — **Specify** (Socratic pre-flight), **Persist** (workflow materialized as three on-disk files), **Dispatch** (clean-context subagents, DPDispatcher fanout, deterministic verification) — with a second Socratic gate reserved for mid-campaign amendments and the deliberate absence of a third per-stage gate. Three end-to-end MD campaigns evidence the contract: polymer-melt viscosity via Green–Kubo ($\eta = 16.02 \pm 0.44$ LJ units), 3C-SiC thermal conductivity via NEMD executed across two sessions after a mid-campaign method amendment, and an in-design Bayesian-optimization run over sequence-specific polymers. Moving from agents that propose experiments to agents that uphold an inspectable contract over time is the practical path to AI scientists in HPC labs.

1. Introduction

Autonomous "AI-scientist" harnesses have made remarkable progress on research tasks with tight, low-stakes feedback loops: ideation pipelines propose hypotheses and draft papers in minutes (Lu et al., 2024; Yamada et al., 2025); machine-learning agents iterate on benchmarks in controlled sandboxes (Nathani et al., 2025); chemistry agents reason over small, reversible experiments (M.B̃ran et al., 2024). Workflow-specification systems (AgentSPEX Authors, 2026) and reactive controllers (Yao et al., 2023) treat the agent as a fast closed-loop decision-maker. Computational physical science breaks that assumption. A single molecular-dynamics stage can run for hours to days on shared HPC queues; realistic campaigns span multiple interactive sessions and human handoffs; a method-applicability error caught late silently discards days of compute.

A second wave of agentic systems targets computational chemistry and materials directly: El Agente for quantum chemistry (Zou et al., 2025), ChemGraph for computational chemistry workflows (Pham et al., 2025), hierarchical multi-agent reasoners for functional-materials discovery (Rothfarb et al., 2025), and end-to-end atomistic-simulation frameworks (Vriza et al., 2026). These shift the target from sandbox benchmarks to project-level campaigns on real HPC compute, contributing task-graph orchestration, reusable tool registries, and artifact-first evidence trails. What remains underspecified across this wave is the *governance contract* itself: which transitions require human re-approval, how mid-campaign method changes are recorded, and how downstream stages are invalidated without silent loss. Superscientist is built around exactly that contract.

In this regime the bottleneck is not ideation but durable, auditable execution across wall-clock time. We argue that autonomy here requires a *durability contract*: a machine-verifiable, versioned, human-signed agreement that states *(a)* what is being computed, *(b)* how results will be verified, and *(c)* which classes of change require re-approval. The contract's value is realized at exactly two moments of Socratic human input — a pre-flight specification and mid-campaign amendment — while all other transitions, including per-stage verification, are deterministic and auditable. The deliberate *absence* of a third Socratic gate is part of the claim: per-stage "confirm?" prompts produce human-in-the-loop theater without the audit trail of a versioned state file.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

We make three contributions. **(1)** A three-pillar framing of the contract — *Specify* / *Persist* / *Dispatch* — with an honest delineation between two Socratic gates and one deterministic verification gate. **(2) Superscientist**, a ten-skill harness realizing the contract over three on-disk checkpoint artifacts (`workflow-state.json`, `progress.log`, `init.sh`) and a DPDispatcher (DeepModeling Team, 2024) backend spanning Shell, Slurm, PBS, LSF, and Bohrium. **(3)** Three end-to-end MD campaigns: polymer-melt viscosity via Green–Kubo ($\eta = 16.02 \pm 0.44$ LJ units), 3C-SiC thermal conductivity via NEMD (executed across two sessions with a mid-campaign method amendment), and an in-design Bayesian-optimization campaign over sequence-specific polymers; a supplementary radius-of-gyration study across Shell, Slurm-GPU, and Slurm-MPI backends (within SEM) evidences the Dispatch pillar.

2. System Overview

Skills, not a DSL. Superscientist is a Claude Code plugin of ten Markdown skills: seven phase skills implement a five-stage pipeline (Design → Plan → Execute → Verify → Complete), and three cross-cutting skills (`session-resume`, `systematic-debugging`, `checkpoint-management`) span all phases. Figure 1 shows the full harness. Skills are natural-language prompts, not Python code — a deliberate contrast with workflow-specification systems (AgentSPEX Authors, 2026) that couple a YAML schema to a Python runtime. Two consequences follow. A domain scientist authors or edits workflow logic without touching orchestration code. And the agent reasons about its workflow in the same medium it executes, removing a code-to-prompt translation layer that becomes a common source of silent semantic drift during long campaigns.

Three-file durability substrate. State lives on disk in three files. `workflow-state.json` owns the stage DAG, per-stage statuses and success criteria, amendment records, and running-process metadata — the full execution graph. `progress.log` is append-only and ISO-8601 timestamped via `$(date -Iseconds)` at the *tool-call* level, never LLM-typed; this removes a common failure mode in which models post-hoc invent times. `init.sh` is an idempotent environment bootstrap re-executed at every session start. Every skill reads and writes these files, and no hidden in-memory state persists between sessions. A fresh process with no prior conversational context therefore recovers a campaign entirely by reading the three files: the harness is session-agnostic by construction.

Dispatch and execution. Each stage executes in a fresh clean-context subagent spawned via the Claude Agent SDK (Anthropic, 2024), reading only the state, log, and plan entries it needs. Compute is submitted through DPDispatcher (DeepModeling Team, 2024), which unifies Shell, Slurm, PBS, LSF, and Bohrium behind one Python API. In asynchronous mode a `tmux` wrapper writes a `DPDISP_DONE` marker on job completion, so the orchestrator can lose its own session and resume monitoring when a later pass reads the marker from disk. For fully autonomous runs, an optional `run-workflow.sh` wrapper chains fresh `claude -p` sessions stage-by-stage; multi-day campaigns proceed without any interactive shell attached.

3. The Durability Contract

Specify: the Socratic pre-flight gate. The `experiment-design` skill drives a Socratic dialogue, one question at a time: what is the physical quantity of interest, what method computes it, what are the stage-wise success criteria, and which method-applicability assumptions must hold. Binaries or inputs that cannot be checked at design time — HPC compile flags, potential-file paths — are tagged `[UNVERIFIED]` and must be resolved before the gate closes. The skill enumerates method pitfalls (e.g., centroid-stress incompatibility with many-body potentials; §4) and attaches a safeguard to each. Output is a Markdown spec committed to git plus a stub `workflow-state.json`. The gate is intentionally *two-layer*: verbal approval during the dialogue *and* written approval of the persisted spec; the two have different failure modes and both are required.

Persist: amendments as the second Socratic gate. On-disk state, not the ongoing conversation, is the source of truth. Any definitional change — parameters, success criteria, stage dependencies, stage add/remove — routes through the amendment protocol: *propose → user approves → version increments → amendment record appended → downstream stages marked* `invalidated`. The `invalidated` status is first-class, not a silent deletion: the previous attempt and its failure mode remain visible in `progress.log` and referenceable from later records. This contrasts with reactive controllers (Yao et al., 2023) that mutate an in-context plan without a durable trace of what changed, when, or why.

Dispatch: deterministic verification and portability. Each stage runs in a fresh subagent that reads only the state, log, and plan entries it needs. Verification is *deterministic* — success-criteria numbers, file existence, physical-reasonableness checks — not a conversational "looks good." On failure, the `systematic-debugging` skill fires automatically and escalation to the user occurs only after retry exhaustion (three local, five remote). We argue explicitly against a third Socratic gate here: per-stage "con-

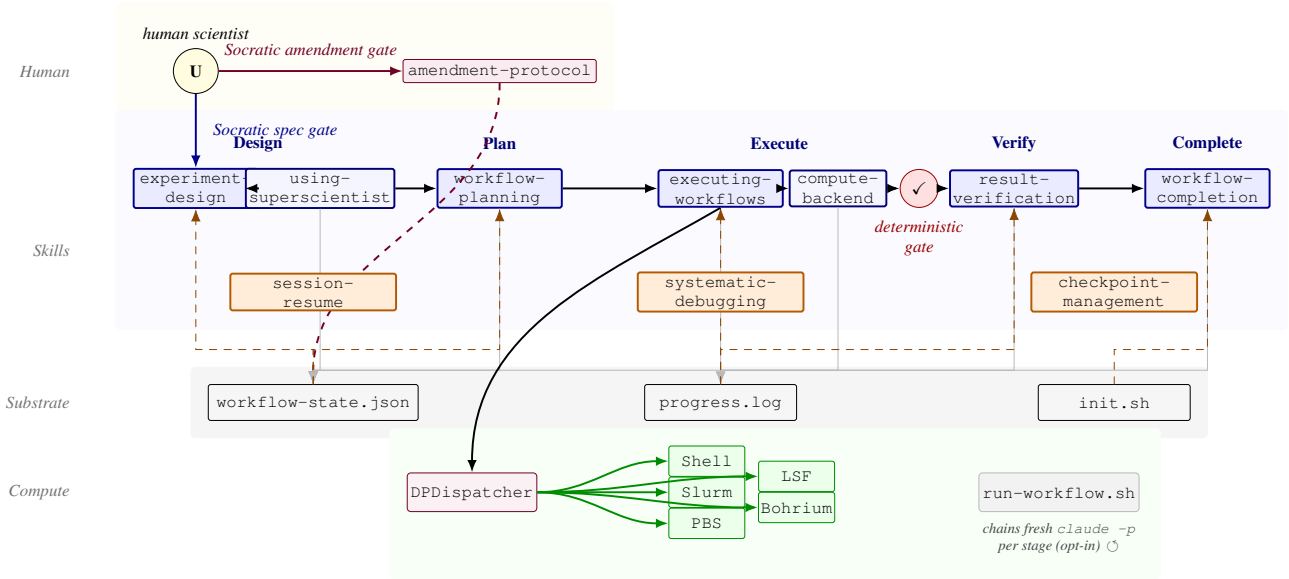


Figure 1. **Superscientist as a harness of ten skills over three persistent checkpoint artifacts.** Socratic human-in-the-loop gates (top) bound both spec time and mid-campaign amendments; all other transitions — including the *deterministic* verification gate between Execute and Verify — proceed without human input and are auditable via `progress.log`. Blue nodes: the seven phase skills of the five-phase pipeline (Design → Plan → Execute → Verify → Complete). Orange nodes: three cross-cutting skills (`session-resume`, `systematic-debugging`, `checkpoint-management`) spanning all phases. Grey files: the three-file durability substrate (`workflow-state.json`, `progress.log`, `init.sh`). Green: DPDispatcher fans compute to Shell / Slurm / PBS / LSF / Bohrium; an optional `run-workflow.sh` wrapper chains fresh `claude -p` sessions for multi-day hands-off runs.

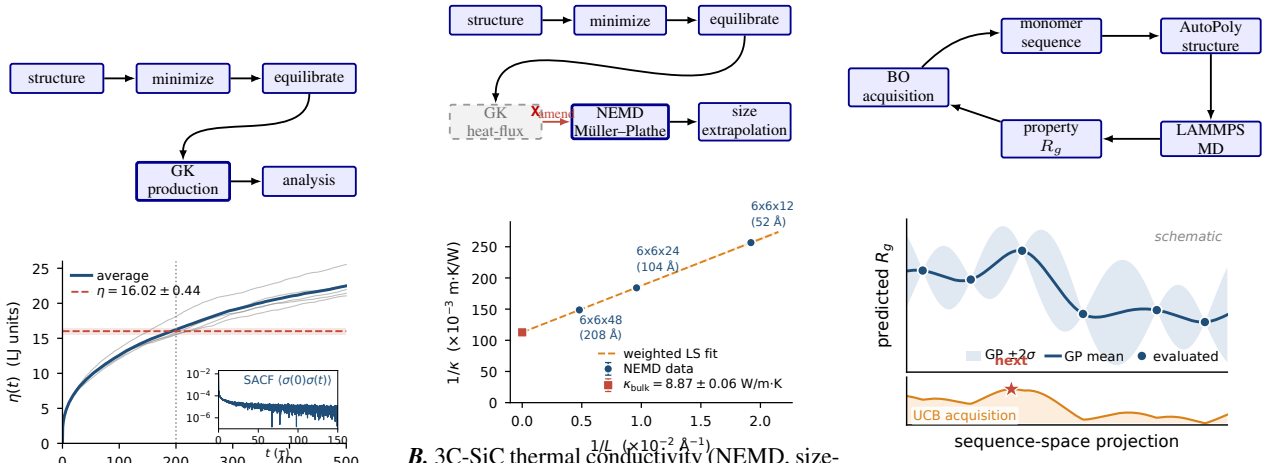
firm?" prompts produce human-in-the-loop theater without the audit trail of a versioned state file. As evidence for portability, the same radius-of-gyration workflow yielded $R_g = 4.847/4.798/4.903\,\sigma$ on local Shell, Slurm-GPU, and Slurm-MPI backends (within SEM); a missing `module load openmpi` triggered two Slurm failures that auto-recovered with corrected submissions, with no human input.

4. Evidence: Three MD Campaigns

Polymer-melt viscosity (Panel A, executed). A five-stage Green–Kubo (Green, 1954; Kubo, 1957) workflow on a Kremer–Grest bead-spring melt (Kremer & Grest, 1990) of volume 5832 σ^3 ran to completion in a single local Shell session (9 h wall-clock, LAMMPS v. Aug 2023 (Thompson et al., 2022; Plimpton, 1995)). The stress autocorrelation running integral reaches a plateau at 100–300 τ, giving $\eta = 16.02 \pm 0.44$ LJ units averaged over five independent stress components. This is the quiet case: no amendments, no verification failures, no cross-session handoff — evidence that contract overhead is negligible when the campaign proceeds as specified.

3C-SiC thermal conductivity (Panel B, executed — anchor). The initial design specified Green–Kubo with a Tersoff (Tersoff, 1989) potential. At stage 4 the harness detected that LAMMPS `compute centroid/stress/atom` — required for correct heat flux with many-body potentials — is not implemented for the Tersoff pair style; the gate fired, stages 4–5 were marked `invalidated`, and an amendment pivoted the method to NEMD Müller–Plathe (Müller-Plathe, 1997) over three box sizes ($6 \times 6 \times \{12, 24, 48\}$). Session 1 ended mid-preparation; session 2 read `workflow-state.json`, found the live `tmux` job via its `DPDISP_DONE` marker, and resumed monitoring at ∼67% of stage-4 runtime. Size extrapolation ($1/\kappa$ vs. $1/L_z$, $R^2 = 0.9999$) gave $\kappa_{\text{bulk}} = 8.9$ W/mK — ∼40× below the Tersoff literature range of 200–400 W/mK (Li et al., 2012), a finite-size and temperature-gradient caveat that we report rather than tune around.

Bayesian-optimized polymer sequences (Panel C, in-design). An active BO loop (Frazier, 2018) iterates monomer sequences against simulated radius of gyration: AutoPoly (AutoPoly Authors, 2024) generates coarse-grained structures, LAMMPS evaluates R_g, and a UCB acquisition over a 500-candidate pool selects the next batch. The campaign is specified but not yet executed; the paper's most honest claim is that *the spec is the artifact*. Figure 2C shows the persisted `workflow-state.json` excerpt alongside the loop — what the Specify pillar delivers before any compute is burned.



A. Polymer-melt viscosity (Green–Kubo). Running integral plateau at 100–300 τ. $\eta = 16.02 \pm 0.44$ **LJ units.**
5 stages, local Shell, 9 h, single session.

B. 3C-SiC thermal conductivity (NEMD, size-extrapolated).
$\kappa_{\mathbf{bulk}} = 8.87 \pm 0.06$ **W/mK** *(∼40× below Tersoff literature; finite-size caveat).*
Amendment fired at stage 4 (17:45 day 1); session 2 resumed at ∼67% the next morning (09:27) without manual recovery.

C. Sequence-optimized polymers via Bayesian optimization.
Design approved; execution pending.
Matérn-5/2 GP, UCB (β=0.5), batch 5, 16 rounds, Δ<0.5% over 3-round convergence; ablation vs. random search planned.

Figure 2. **Three end-to-end MD campaigns.** Panels A and B are completed runs; Panel C is the in-design Bayesian-optimization campaign, shown as its persisted spec. Panel B illustrates the paper's core claim: when the original method failed a pitfall check at stage 4 (`centroid/stress/atom` unsupported for Tersoff), the Socratic amendment gate fired, downstream stages were invalidated on record, and execution resumed across two sessions without loss of traceability.

5. Governance, Safeguards, and Roadmap

Attribution as a log, not a label. Every state transition is logged with an ISO-8601 timestamp and the acting tool; timestamps are generated via `$(date -Iseconds)` at tool-call time, never by the model, removing a common failure mode in which agents post-hoc invent times. The amendment schema is intentionally minimal — {`version`, `changes`, `rationale`, `invalidated_stages`, `approved_by`} — capturing the minimum disclosure needed to reconstruct who decided what, when, and with what downstream effect. This does not resolve authorship norms for AI-assisted science (Birhane et al., 2022), but it makes human and AI contributions mechanically recordable, a prerequisite for any honest attribution policy.

Safety and dual-use. We see three concerns with matched mitigations. *(a)* Over-claiming from under-converged simulations is controlled by the deterministic verification gate and by requiring honest caveats in the reporting stage — the ∼40× SiC gap in §4 is the worked example. *(b)* Silent drift across long campaigns is prevented by routing definitional changes *only* through the versioned amendment protocol. *(c)* Autonomous dispatch is limited to user-configured DPDispatcher backends; there is no resource discovery. Explicit scope exclusions: no wet-lab control, no biological-sequence generation, and no autonomous literature-to-experiment loops — the Socratic pre-flight gate breaks that loop by design.

Evaluation roadmap. Four planned ablations close the evaluation gap. *(a)* Contract-enforced vs. gate-free baseline on a common MD task, measuring spec-amendment rate, false-completion rate (stages marked complete whose success criteria fail an independent audit), and human-intervention count. *(b)* Autonomous-chaining budget scaling (sessions vs. `claude -p` invocation cost). *(c)* Cross-backend reproducibility extended from R_g to a full MD suite. *(d)* Bayesian-optimization execution with a regret curve against random search. Baselines are workflow-specification systems (AgentSPEX Authors, 2026), fast-loop harnesses (Yamada et al., 2025), and a ReAct (Yao et al., 2023) reactive controller.

References

AgentSPEX Authors. AgentSPEX: A workflow specification and execution framework for autonomous scientific agents, 2026. TODO-verify: confirm authors, venue, DOI from original paper cited in design spec §7.4.

Anthropic. Claude agent SDK and claude code. https://docs.claude.com/en/api/agent-sdk/overview, 2024. TODO-verify: confirm final URL and release dates.

AutoPoly Authors. AutoPoly: Automated polymer structure generation for LAMMPS. Software package, 2024. TODO-verify: confirm authors, release, and DOI/Zenodo record.

Birhane, A., Kalluri, P., Card, D., Agnew, W., Dotan, R., and Bao, M. The values encoded in machine learning research. *2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2022. TODO-verify: exact pages/DOI; check if this is the right Birhane reference for attribution framing per design spec §7.4.

DeepModeling Team. DPDispatcher: A python package for generating HPC scheduler jobs from JSON input. https://github.com/deepmodeling/dpdispatcher, 2024. TODO-verify: confirm release version and attribution.

Frazier, P. I. A tutorial on Bayesian optimization. *arXiv preprint*, 2018.

Green, M. S. Markoff random processes and the statistical mechanics of time-dependent phenomena. II. irreversible processes in fluids. *Journal of Chemical Physics*, 22(3): 398–413, 1954. doi: 10.1063/1.1740082.

Kremer, K. and Grest, G. S. Dynamics of entangled linear polymer melts: A molecular-dynamics simulation. *Journal of Chemical Physics*, 92(8):5057–5086, 1990. doi: 10.1063/1.458541.

Kubo, R. Statistical-mechanical theory of irreversible processes. I. general theory and simple applications to magnetic and conduction problems. *Journal of the Physical Society of Japan*, 12(6):570–586, 1957. doi: 10.1143/JPSJ.12.570.

Li, W., Lindsay, L., Broido, D. A., Stewart, D. A., and Mingo, N. Thermal conductivity of bulk and nanowire $\mathrm{Mg_2si}_x\mathrm{sn}_{1-x}$ alloys from molecular dynamics simulations. *Physical Review B*, 2012. TODO-verify: confirm exact title and citation target for 3C-SiC Tersoff benchmark anchoring Panel B caveat; consider using Porter et al. or Goel et al. (2016) instead.

Lu, C., Lu, C., Lange, R. T., Foerster, J., Clune, J., and Ha, D. The AI scientist: Towards fully automated open-ended scientific discovery. arXiv preprint, 2024. TODO-verify: confirm v1 vs. v2 citation target.

M.B̃ran, A., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 2024. arXiv:2304.05376. TODO-verify: volume, pages, DOI.

Müller-Plathe, F. A simple nonequilibrium molecular dynamics method for calculating the thermal conductivity. *Journal of Chemical Physics*, 106(14):6082–6085, 1997. doi: 10.1063/1.473271.

Nathani, D., Madaan, L., Roberts, N., et al. MLGym: A new framework and benchmark for advancing AI research agents. arXiv preprint, 2025. TODO-verify: confirm arXiv ID, full author list, version.

Pham, T. D., Tanikanti, A., and Keçeli, M. ChemGraph: An agentic framework for computational chemistry workflows. arXiv preprint, 2025. TODO-verify: confirm arXiv ID and version.

Plimpton, S. Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, 117 (1):1–19, 1995. doi: 10.1006/jcph.1995.1039.

Rothfarb, S., Davis, M. C., Matanovic, I., Li, B., Holby, E. F., and Kort-Kamp, W. J. M. Hierarchical multi-agent large language model reasoning for autonomous functional materials discovery. arXiv preprint, 2025. TODO-verify: confirm arXiv ID and version.

Tersoff, J. Modeling solid-state chemistry: Interatomic potentials for multicomponent systems. *Physical Review B*, 39(8):5566–5568, 1989. doi: 10.1103/PhysRevB.39.5566.

Thompson, A. P., Aktulga, H. M., Berger, R., Bolintineanu, D. S., Brown, W. M., Crozier, P. S., in 't Veld, P. J., Kohlmeyer, A., Moore, S. G., Nguyen, T. D., Shan, R., Stevens, M. J., Tranchida, J., Trott, C., and Plimpton, S. J. LAMMPS — a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications*, 271:108171, 2022. doi: 10.1016/j.cpc.2021.108171.

Vriza, A., Kornu, U., Koneru, A., Chan, H., and Sankaranarayanan, S. K. R. S. Multi-agentic AI framework for end-to-end atomistic simulations. *Digital Discovery*, 2026. TODO-verify: confirm volume, pages, and DOI; advance article at time of cite.

Yamada, Y., Lange, R. T., Lu, C., Hu, S., Lu, C., Foerster, J., Clune, J., and Ha, D. The AI scientist-v2: Workshop-level automated scientific discovery via agentic tree search. Sakana AI technical report, 2025. TODO-verify: confirm arXiv ID and publication venue.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Zou, Y., Cheng, A. H., Aldossary, A., Bai, J., Leong, S. X., Campos-Gonzalez-Angulo, J. A., Choi, C., Ser, C. T., Tom, G., Wang, A., Zhang, Z., Yakavets, I., Hao, H., Crebolder, C., Bernales, V., and Aspuru-Guzik, A. El Agente: An autonomous agent for quantum chemistry. *Matter*, 8(7), 2025. TODO-verify: confirm pages and DOI.