
An AI-Native Protocol and Ecosystem for Computational Earth and Space Science

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Computational models are central to Earth and space science, from land surface
2 hydrology and atmospheric dynamics to ocean circulation and space weather pre-
3 diction. Yet their development and use remain confined to small communities of
4 specialists who can navigate codebases of 200,000 or more lines of Fortran and
5 C. The vast majority of domain scientists, those with deep expertise in hydrol-
6 ogy, biogeochemistry, atmospheric dynamics, cryosphere physics, space plasma
7 physics, or related fields, lack the software engineering skills to modify, configure,
8 or confidently run these models. As a result, they are restricted to qualitative obser-
9 vations rather than the rigorous quantitative simulations their research demands.
10 We propose two complementary responses. First, a simple protocol that constrains
11 AI-assisted model development by inheriting the hierarchical, process-oriented
12 workflow already practiced in physical model work: grounded repository tracing,
13 component-level tests, conservation and bounds checks, and mandatory expert
14 review before acceptance of AI-mediated modifications. Second, an AI-native
15 ecosystem that allows domain scientists to design, execute, and analyze simulations
16 entirely through natural language interaction with AI coding agents, removing the
17 software engineering barrier without compromising scientific rigor. The protocol
18 and ecosystem are illustrated using Earth System Models but apply to any compu-
19 tational science codebase where physical correctness must be preserved: weather
20 prediction, ocean modeling, space weather simulation, hydrology, and beyond.

21 1 Protocol

22 **Premise.** Physical model development already follows a disciplined structure: scientists isolate a
23 process, identify the responsible component, modify the relevant code path, run targeted diagnostics,
24 and only then evaluate coupled behavior. Climate-model assessment is similarly hierarchical, with
25 component-aware and process-aware evaluation preceding broad claims about full-system skill [1–3].
26 The same discipline governs ocean models, space weather codes, and hydrological frameworks:
27 changes must be validated at the process level before being trusted in coupled configurations. We
28 argue that AI development for scientific modeling should inherit this protocol instead of bypassing it.

29 **Step 1: define the scientific target.** Every AI task should begin with an expert-specified process ques-
30 tion such as runoff generation, boundary-layer turbulence, sea-ice thermodynamics, land-atmosphere
31 flux exchange, magnetospheric reconnection, or ionospheric electron density evolution. The unit of
32 work is not “improve the model,” but “inspect or modify the mechanism responsible for this process.”

33 **Step 2: ground the task in repository structure.** The AI system must map the scientific target
34 onto actual model interfaces: documentation, namelists or parameter tables, driver code, component
35 modules, and diagnostics [4–6]. This is realistic in current repositories. Noah-MP separates docs,

36 drivers, parameters, src, and utility logic; WRF exposes distinct physics, registry, run, and test
37 layers; E3SM separates components, drivers, configuration, and shared test infrastructure. Space
38 weather codes such as BOUT++ similarly organize physics modules, input configurations, and
39 diagnostic outputs into navigable layers [7]. This structure should guide the agent’s search space.

40 **Step 3: run hierarchical checks before coupled claims.** Proposed AI changes should first pass the
41 narrowest appropriate checks: compilation, smoke tests, unit or component tests, and process-oriented
42 diagnostics. Only after these pass should the workflow escalate to coupled or longer integrations.
43 This mirrors how physical models are debugged in practice, whether in land surface, atmosphere,
44 ocean, or magnetosphere codes, and limits silent scientific failure [8].

45 **Step 4: require physical validity checks.** Success cannot be defined by compilation or benchmark
46 completion alone. Expert-designed checks such as conservation (mass, energy, momentum, magnetic
47 flux), boundedness, stability, and regime-specific diagnostics must be explicit acceptance criteria [4, 9].
48 In Noah-MP, AI-assisted edits should remain tied to existing balance-check logic. In ocean models,
49 AI must verify tracer conservation. In space weather codes, AI must preserve divergence-free
50 magnetic field constraints. The principle is universal: the physics dictates the acceptance test.

51 **Step 5: preserve expert sign-off through structured review.** The final decision remains with
52 domain scientists. AI can accelerate localization, explanation, patch drafting, and test execution,
53 but it should not decide that a scientifically consequential change is correct. Concretely, every
54 AI-mediated modification should produce a review artifact: a pull request or equivalent that contains
55 the proposed diff, the validation results from Steps 3–4, and a plain-language summary of what was
56 changed and why. A domain expert must approve this artifact before the change enters any shared
57 branch. This mirrors existing community governance in projects like CESM and E3SM, where code
58 changes require reviewer approval before merging [10, 11]. The protocol is therefore human-led and
59 AI-assisted by design.

60 2 Ecosystem: AI as the Interface to Computational Science

61 The protocol above governs how AI should behave when modifying model code. But limiting AI to
62 the developer workflow misses a larger opportunity. The deeper problem across computational Earth
63 and space science is not that model development is slow; it is that the models are inaccessible to most
64 of the scientists who need them.

65 **The accessibility gap.** Consider the current landscape. Earth System Models such as Noah-MP,
66 WRF, CESM, E3SM, and MOM6 are written in Fortran or C, with codebases ranging from tens
67 of thousands to hundreds of thousands of lines [12, 5, 6, 13, 14]. The same is true for ocean
68 models, coupled hydrology frameworks like ParFlow [15], and space weather codes. Learning to
69 use any one of them requires attending specialized tutorials, workshops, or short courses [16–22].
70 Entry is competitive and expensive. The WRF tutorial is strictly limited to 60–70 participants per
71 session [16]. The CESM tutorial selects approximately 80 students from a larger applicant pool [17].
72 Professional short courses for integrated modeling can exceed \$1,100 in registration fees [22]. Even
73 after training, the gap between understanding a model conceptually and being able to modify its code,
74 configure its parameters, interpret its outputs, and debug its failures remains vast. A hydrologist who
75 understands infiltration excess runoff at the equation level may still be unable to change a single
76 parameterization option in Noah-MP without weeks of code archaeology. A space physicist who
77 derives magnetohydrodynamic equations on a whiteboard may be unable to add a new boundary
78 condition to a simulation code.

79 **What an AI-native ecosystem would look like.** We envision a pipeline where domain scientists
80 interact with computational science codes entirely through natural language, mediated by AI coding
81 agents such as Claude Code [23], Gemini CLI [24], or similar tools. The pipeline would test whether
82 AI can:

- 83 1. **Translate hypotheses into configurations.** A scientist states a high-level research question
84 (“How does switching from a simple water table scheme to the MMF aquifer model affect
85 baseflow in arid regions?” or “What happens to ring current dynamics if we double the
86 plasmasphere refilling rate?”), and the AI translates this into the concrete parameter files,
87 namelist changes, or code modifications required to set up the experiment.

- 88 2. **Prepare and validate input data.** In practice, 60–70% of the effort in a modeling study is not
89 code modification but data preparation: acquiring and reformatting meteorological forcing data,
90 soil and land cover parameters, initial and boundary conditions, and observational datasets for
91 calibration [25, 26]. The AI should assist with downloading, regridding, quality-checking, and
92 formatting these inputs, and flag mismatches between forcing data resolution, model grid, and
93 simulation period before the run begins.
- 94 3. **Autonomously debug runtime errors.** Model compilation and execution are notoriously fragile
95 across platforms. The AI handles environment configuration, library dependencies, compiler flag
96 issues, segmentation faults from out-of-bounds arrays, and the cascading failures that routinely
97 block non-expert users for days or weeks.
- 98 4. **Process outputs into scientific visualizations.** Raw model outputs (typically NetCDF files
99 containing hundreds of variables across space and time) must be post-processed into meaningful
100 diagnostics: spatial maps, time series, vertical profiles, budgets, and regime-conditional analyses.
101 The AI should produce accurate, publication-quality visualizations directly from natural language
102 descriptions of what the scientist wants to see.
- 103 5. **Guide experimental design without hallucinating physics.** The AI helps scientists design
104 controlled experiments, explains the physical assumptions behind different parameterization
105 options, and identifies potential confounders, all without fabricating mechanisms or citing
106 nonexistent literature. This is where the protocol (Section 1) becomes essential: the AI’s
107 guidance must be grounded in the actual model code and documentation, not in plausible-
108 sounding but unverifiable claims.

109 **Measuring success.** The ecosystem’s value should be measured by outcomes that matter to working
110 scientists: (a) reduction in time-to-experiment, from weeks or months of setup to hours of interactive
111 sessions; (b) accuracy of AI-generated configurations, verified against expert-prepared reference
112 configurations; (c) reliability of autonomous debugging, measured by the fraction of common build
113 and runtime errors resolved without human intervention; and (d) scientific correctness of the AI’s
114 explanations and experimental guidance, evaluated by domain experts for factual accuracy and
115 absence of hallucinated mechanisms.

116 **The Fortran gap.** A practical obstacle deserves explicit acknowledgment. The vast majority of
117 Earth System Models and space weather codes are written in Fortran, a language dramatically
118 underrepresented in LLM training corpora relative to Python or JavaScript [27, 28]. Current AI
119 coding agents produce syntactically invalid Fortran at substantially higher rates than for mainstream
120 languages, particularly around array semantics, intent declarations, module interfaces, and implicit
121 typing. The protocol’s hierarchical checks (Step 3) partially mitigate this by catching compilation
122 failures early, but the ecosystem must also adopt compiler-in-the-loop validation: every AI-generated
123 Fortran edit should be compiled and tested before being presented to the scientist, not after.

124 3 Example, Scope, and Implication

125 **A concrete example.** Consider surface runoff parameterization in Noah-MP. A scientist wants to com-
126 pare VIC-style and TOPMODEL-style runoff schemes [29]. In the current workflow, this requires un-
127 derstanding how the option propagates through `ConfigVarType.F90`, `NoahmpReadTableMod.F90`,
128 `NoahmpTable.TBL`, and the relevant `RunoffSurface*Mod.F90` implementations [4]. Under the
129 AI-native ecosystem, the scientist simply describes the comparison in natural language. The AI,
130 constrained by the protocol, traces the option through the repository, prepares a narrowly scoped con-
131 figuration change, runs short validation and balance checks, and presents the results as interpretable
132 diagnostics. The scientist reviews the AI’s work at each stage, retaining full scientific authority while
133 being freed from the software engineering burden.

134 **Scope: anywhere there is code, there is AI.** The protocol and ecosystem are not limited to
135 Earth System Models. Any computational science domain where large legacy codebases encode
136 physical knowledge faces the same accessibility barrier: space weather models (SWMF, BATS-
137 R-US, BOUT++), hydrological frameworks (SUMMA, ParFlow), ocean models (MOM6, NEMO,
138 POP), atmospheric chemistry models, seismic wave propagation codes, and astrophysical simulation
139 packages. The common pattern is a codebase that is physically rich but software-engineering-intensive,

maintained by a small group but needed by a much larger scientific community. Wherever this pattern holds, the protocol provides the safety constraints and the ecosystem provides the accessibility bridge.

Implication. The combination of protocol and ecosystem redefines both who can do computational science and how it is done. The protocol ensures that AI-assisted development remains trustworthy, auditable, and physically grounded. The same staged logic (scientific intent → repository grounding → component validation → coupled evaluation → expert judgment) transfers across repositories because these codebases are already organized around components, drivers, configurations, and validation artifacts [30, 5, 6]. The ecosystem removes the software engineering barrier that currently excludes the majority of domain scientists from computational investigation. If a hydrologist can run a controlled Noah-MP experiment through conversation rather than code, a biogeochemist can set up a CESM carbon cycle perturbation without learning Fortran, and a space physicist can modify a magnetosphere boundary condition without mastering C++, then the community of scientists who can do rigorous quantitative analysis expands by an order of magnitude. Benchmarks such as ESM-BENCH can then evaluate not only whether an agent changed code correctly, but whether it enabled a scientist to go from hypothesis to result without sacrificing physical correctness [31].

References

- [1] Veronika Eyring, Sandrine Bony, Gerald A Meehl, Catherine A Senior, Bjorn Stevens, Ronald J Stouffer, and Karl E Taylor. Overview of the coupled model intercomparison project phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development*, 9(5): 1937–1958, 2016. URL <https://gmd.copernicus.org/articles/9/1937/2016/>.
- [2] Mattia Righi, Bouwe Andela, Veronika Eyring, Axel Lauer, Valeriu Predoi, Manuel Schlund, Javier Vegas-Regidor, Lisa Bock, Björn Brötz, Lee de Mora, et al. ESMValTool v2.0 – technical overview. *Geoscientific Model Development*, 13(3):1179–1199, 2020. URL <https://gmd.copernicus.org/preprints/gmd-2019-226/gmd-2019-226.pdf>.
- [3] Frédéric Hourdin, Thorsten Mauritsen, Andrew Gettelman, Jean-Christophe Golaz, Venkatramani Balaji, Qingyun Duan, Doris Folini, Duoying Ji, Daniel Klocke, Yun Qian, Florian Rauser, Catherine Rio, Lorenzo Tomassini, Masahiro Watanabe, and Daniel Williamson. The art and science of climate model tuning. *Bulletin of the American Meteorological Society*, 98(3):589–602, 2017. doi: 10.1175/BAMS-D-15-00135.1. URL <https://journals.ametsoc.org/view/journals/bams/98/3/bams-d-15-00135.1.xml>.
- [4] Cenlin He, Prasanth Valayamkunnath, Michael Barlage, Fei Chen, David Gochis, Ryan Cabell, Tim Schneider, Roy Rasmussen, Guo-Yue Niu, Zong-Liang Yang, Dev Niyogi, and Michael Ek. The community Noah-MP land surface modeling system technical description version 5.0. Technical Report NCAR/TN-575+STR, National Center for Atmospheric Research, 2023. URL https://www2.mmm.ucar.edu/wrf/users/physics/phys_refs/LAND_SURFACE/noah_mp_tech_note.pdf.
- [5] Jordan G. Powers, Joseph B. Klemp, William C. Skamarock, Christopher A. Davis, Jimmy Dudhia, David O. Gill, Janice L. Coen, David J. Gochis, Ravan Ahmadov, Steven E. Peckham, Georg A. Grell, John Michalakes, Samuel Trahan, Stanley G. Benjamin, Curtis R. Alexander, Geoffrey J. Dimego, Wei Wang, Craig S. Schwartz, Glen S. Romine, Zhiqian Liu, Chris Snyder, Fei Chen, Michael J. Barlage, Wei Yu, and Michael G. Duda. The weather research and forecasting model: Overview, system efforts, and future directions. *Bulletin of the American Meteorological Society*, 98(8):1717–1737, 2017. doi: 10.1175/BAMS-D-15-00308.1. URL <https://journals.ametsoc.org/view/journals/bams/98/8/bams-d-15-00308.1.xml>.
- [6] Jean-Christophe Golaz, Peter M. Caldwell, Luke P. Van Roekel, Mark R. Petersen, Qi Tang, Jonathan D. Wolfe, Gebre Abeshu, Valentine Anantharaj, Xylar S. Asay-Davis, David C. Bader, Scott A. Baldwin, Gautam Bisht, Peter A. Bogenschutz, Marcia Branstetter, Michael A. Brunke, Steven R. Brus, Susannah M. Burrows, Philip J. Cameron-Smith, Aaron S. Donahue, Michael Deakin, Richard C. Easter, Katherine J. Evans, Yan Feng, Mark Flanner, James G. Foucar, Steven J. Ghan, Brian M. Griffin, Cecile Hannay, Bryce E. Harrop, Forrest M. Hoffman, Andrew Hunter, Robert Jacob, Douglas W. Jacobsen, Nicole Jeffery, Philip W. Jones, Noel D. Keen, Stephen A. Klein, Vincent E. Larson, L. Ruby Leung, Hong-Yi Li, Wuyin Lin, William H. Lipscomb, Po-Lun Ma, Salil Mahajan, Mathew E. Maltrud, Azamat Mametjanov, Julie L.

- McClean, Renata B. McCoy, Richard B. Neale, Stephen F. Price, Yun Qian, Philip J. Rasch, J. E. Jack Reeves Eyre, William J. Riley, Todd D. Ringler, Andrew F. Roberts, Andrew G. Salinger, Zeshan Shaheen, Xiaoying Shi, Balwinder Singh, Jinyun Tang, Mark A. Taylor, Peter E. Thornton, Adrian K. Turner, Milena Veneziani, Hui Wan, Hailong Wang, Shanlin Wang, Dean N. Williams, Phillip J. Wolfram, Patrick H. Worley, Shaocheng Xie, Yang Yang, Jin-Ho Yoon, Kai Zhang, Yuying Zhang, and Xue Zheng. The DOE E3SM coupled model version 1: Overview and evaluation at standard resolution. *Journal of Advances in Modeling Earth Systems*, 11(7):2089–2129, 2019. doi: 10.1029/2018MS001603.
- [7] B. D. Dudson, M. V. Umansky, X. Q. Xu, P. B. Snyder, and H. R. Wilson. BOUT++: A framework for parallel plasma fluid simulations. *Computer Physics Communications*, 180(9):1467–1480, 2009. doi: 10.1016/j.cpc.2009.03.008. URL <https://doi.org/10.1016/j.cpc.2009.03.008>.
- [8] Thomas Clune and Richard Rood. Software testing and verification in climate model development. *IEEE Software*, 28(6):49–55, 2011. URL <https://ieeexplore.ieee.org/document/5999647>.
- [9] Derek Hansen, Danielle C. Maddix, Shima Alizadeh, Gaurav Gupta, and Michael W. Mahoney. Learning physical models that can respect conservation laws. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*. PMLR, 2023. URL <https://arxiv.org/abs/2302.11002>.
- [10] Community Earth System Model Project. CESM2 release series information. <https://www.cesm.ucar.edu/models/cesm2/releases>, 2026. Accessed 2026-04-26.
- [11] E3SM Project. E3SM releases. <https://github.com/E3SM-Project/E3SM/releases>, 2026. Accessed 2026-04-26.
- [12] Guo-Yue Niu, Zong-Liang Yang, Kenneth E Mitchell, Fei Chen, Michael B Ek, Michael Barlage, Anil Kumar, Kevin Manning, Dev Niyogi, Enrique Rosero, Mukul Tewari, and Youlong Xia. The community Noah land surface model with multiparameterization options (Noah-MP): 1. model description and evaluation with local-scale measurements. *Journal of Geophysical Research: Atmospheres*, 116(D12):D12109, 2011. doi: 10.1029/2010JD015139.
- [13] Gokhan Danabasoglu, Jean-François Lamarque, Julio Bacmeister, David A Bailey, Alice K DuVivier, Jim Edwards, Louisa K Emmons, John Fasullo, Rolando Garcia, Andrew Gettelman, et al. The community earth system model version 2 (CESM2). *Journal of Advances in Modeling Earth Systems*, 12(2):e2019MS001916, 2020. URL <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2019ms001916>.
- [14] Alistair Adcroft, Whit Anderson, V. Balaji, Chris Blanton, Mitchell Bushuk, Carolina O. Dufour, John P. Dunne, Stephen M. Griffies, Robert Hallberg, Matthew J. Harrison, Isaac M. Held, Malte F. Jansen, Jasmin G. John, John P. Krasting, Amy R. Langenhorst, Sonya Legg, Zhi Liang, Colleen McHugh, Aparna Radhakrishnan, Brandon G. Reichl, Tony Rosati, Bonita L. Samuels, Andrew Shao, Ronald Stouffer, Michael Winton, Andrew T. Wittenberg, Baoqiang Xiang, Niki Zadeh, and Rong Zhang. The GFDL global ocean and sea ice model OM4.0: Model description and simulation features. *Journal of Advances in Modeling Earth Systems*, 11(10):3167–3211, 2019. doi: 10.1029/2019MS001726.
- [15] Stefan J. Kollet and Reed M. Maxwell. Integrated surface–groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model. *Advances in Water Resources*, 29(7):945–958, July 2006. doi: 10.1016/j.advwatres.2005.08.006. URL <https://www.sciencedirect.com/science/article/pii/S0309170805002101>.
- [16] NCAR Mesoscale & Microscale Meteorology Laboratory. WRF tutorial. <https://www2.mmm.ucar.edu/wrf/users/tutorial/tutorial.html>, 2026. Accessed 2026-04-26. Specialized training strictly limited to 60–70 participants per session.
- [17] NCAR Climate & Global Dynamics Laboratory. CESM tutorial. <https://www.cesm.ucar.edu/events/tutorials>, 2026. Accessed 2026-04-26. Highly competitive selection process with 80 students selected from a larger applicant pool.

- [18] E3SM Project. E3SM tutorial workshop. <https://e3sm.atlassian.net/wiki/spaces/ET/overview>, 2024. Accessed 2026-04-26. Virtual training with recordings and materials hosted on Confluence.
- [19] GFDL/NCAR MOM6 Development Team. MOM6 community tutorial. <https://github.com/mom-ocean/MOM6/wiki>, 2025. Accessed 2026-04-26. Intensive 3–5 day workshops for regional and global configurations.
- [20] NCAR Research Applications Laboratory. Noah-MP community workshop and tutorial. <https://ral.ucar.edu/model/noah-multiparameterization-land-surface-model-noah-mp-lsm>, 2012. Accessed 2026-04-26.
- [21] American Meteorological Society. AMS short courses. <https://annual.ametsoc.org/2026/attend/registration/short-course-registration/>, 2026. Accessed 2026-04-26. Technical tutorials with registration fees ranging from \$40 to \$360.
- [22] International Ground Water Modeling Center. ParFlow short course. <https://igwmc.princeton.edu/short-courses/>, 2024. Accessed 2026-04-26. Professional registration for integrated modeling tracks can exceed \$1,100.
- [23] Anthropic. Claude code: An agentic coding companion, 2025. URL <https://anthropic.com/claude-code>.
- [24] Google. Gemini CLI: An interactive command-line ai assistant, 2026. URL <https://github.com/google/gemini-cli>.
- [25] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020. URL <https://rmets.onlinelibrary.wiley.com/doi/10.1002/qj.3803>.
- [26] Gilberto Pastorello, Carlo Trotta, Eleonora Canfora, Housen Chu, Danielle Christianson, You-Wei Cheah, Cristina Poindexter, Jiquan Chen, Abdelrahman Elbashandy, Marty Humphrey, et al. The FLUXNET2015 dataset and the ONEFlux processing pipeline for eddy covariance data. *Scientific Data*, 7(1):225, 2020. URL <https://www.nature.com/articles/s41597-020-0534-3>.
- [27] Milan Curcic, Ondrej Certík, Brad Richardson, Sebastian Ehlert, Laurence Kedward, Arjen Markus, Ivan Pribec, and Jérémie Vandenplas. Toward modern Fortran tooling and a thriving developer community. *CoRR*, abs/2109.07382, 2021. URL <https://arxiv.org/abs/2109.07382>.
- [28] Dragos Chirilă and Gerrit Lohmann. *Introduction to Modern Fortran for the Earth System Sciences*. Springer, 2015. doi: 10.1007/978-3-662-44910-5. URL <https://paleodyn.uni-bremen.de/study/FortranBook.pdf>.
- [29] Xu Liang, Dennis P. Lettenmaier, Eric F. Wood, and Stephen J. Burges. A simple hydrologically based model of land surface water and energy fluxes for general circulation models. *Journal of Geophysical Research: Atmospheres*, 99(D7):14415–14428, 1994. doi: 10.1029/94JD00483.
- [30] Gregory Flato, Jochem Marotzke, Babatunde Abiodun, Pascale Braconnot, Sin Chan Chou, William D. Collins, Peter M. Cox, Fatima Driouech, Seita Emori, Veronika Eyring, Chris Forest, Peter Gleckler, Eric Guilyardi, Christian Jakob, Vladimir Kattsov, Chris Reason, and Markku Rummukainen. Evaluation of climate models. *Climate change 2013: the physical science basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 741–866, 2013. URL https://www.ipcc.ch/site/assets/uploads/2018/02/WG1AR5_Chapter09_FINAL.pdf.
- [31] Koutian Wu et al. ESM-BENCH: A benchmark for evaluating whether AI agents understand earth system model physics and code. <https://zenodo.org/records/19802836>, 2026.