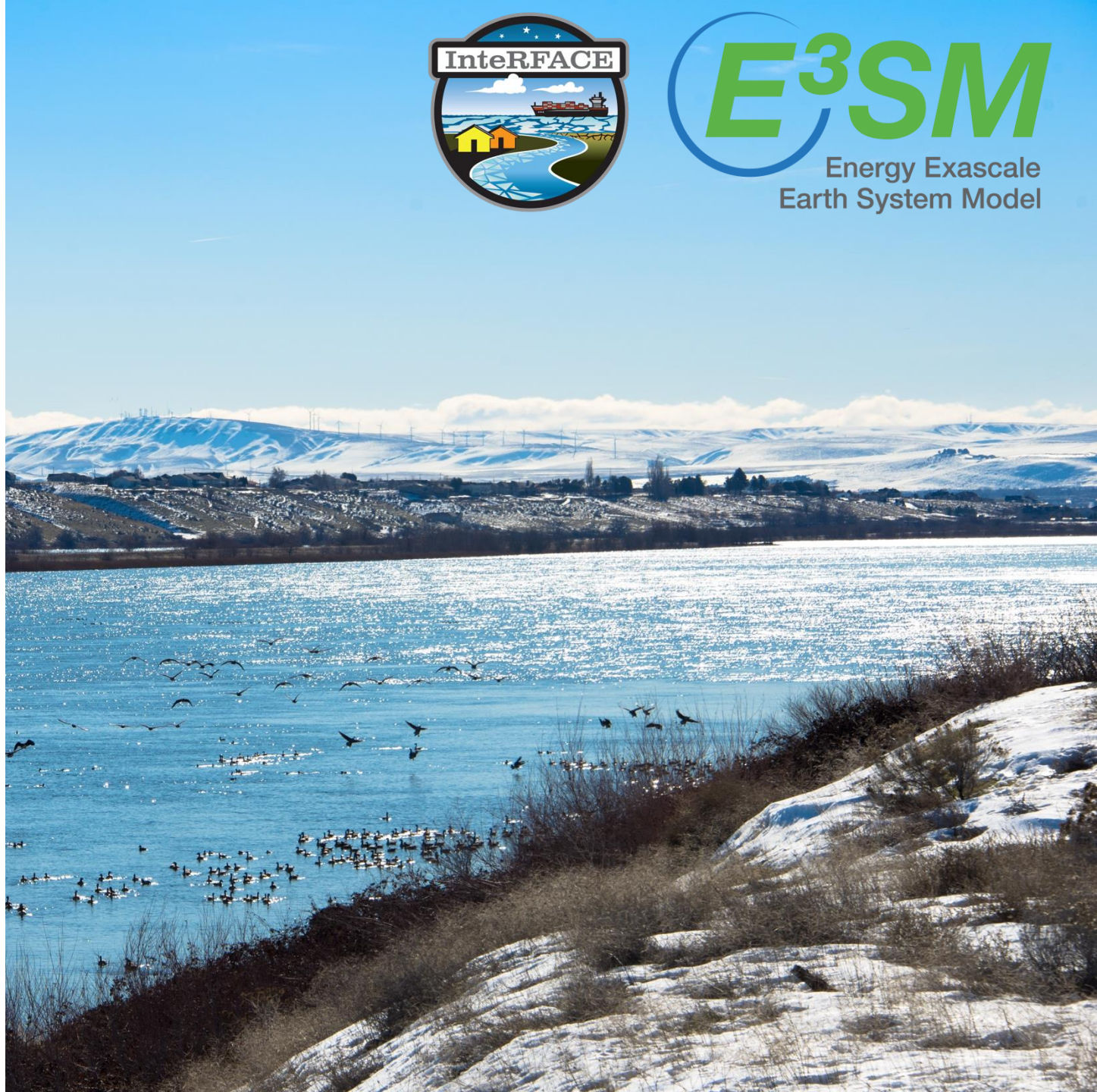
Pacific Northwest NATIONAL LABORATORY

# Can We Trust LLMs for Complex Earth System Model Analysis?

**Tian Zhou**

**HydroML26**

5/19/2026




U.S. DEPARTMENT OF ENERGY BATTELLE

PNNL is operated by Battelle for the U.S. Department of Energy

Pacific Northwest
NATIONAL LABORATORY

# LLMs can now write working scientific scripts.

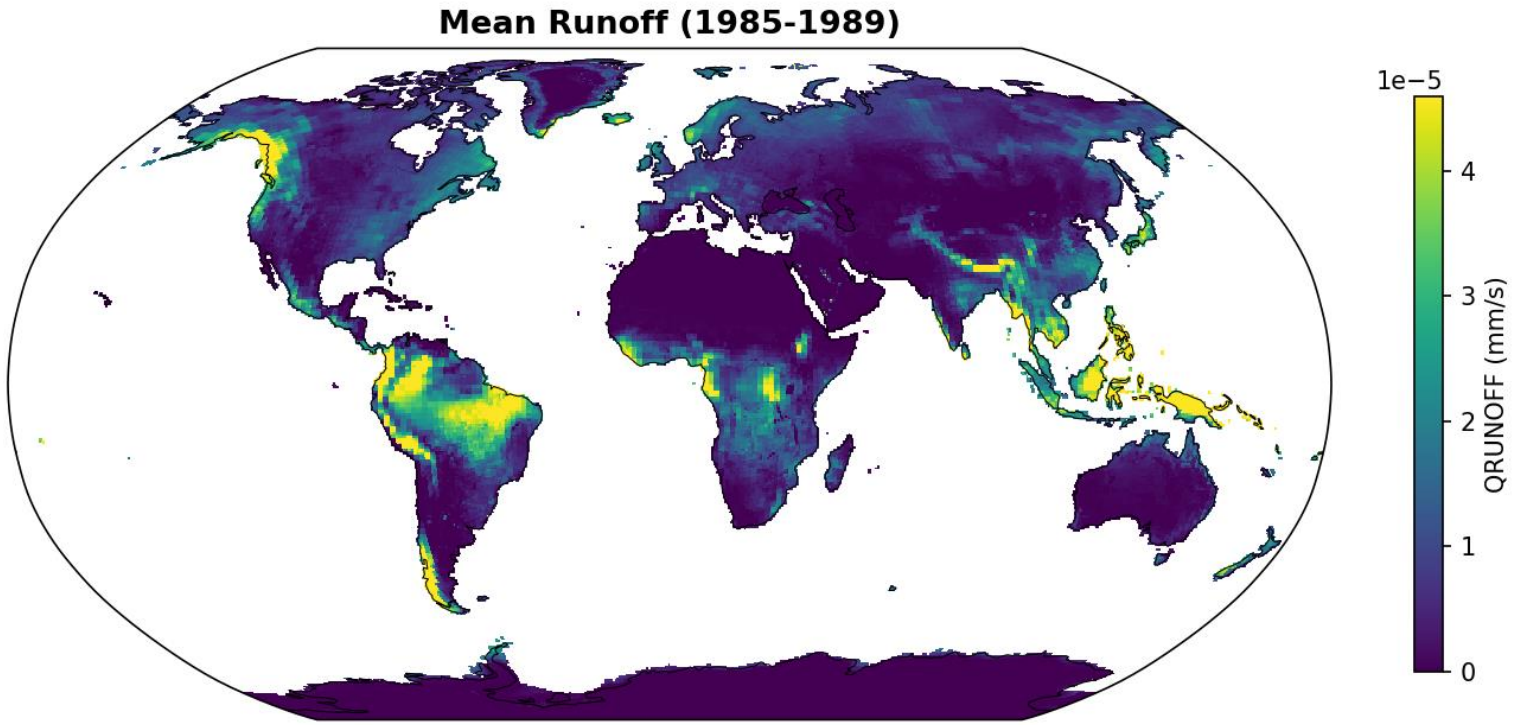
Pacific Northwest
NATIONAL LABORATORY

LLMs can now write working scientific scripts.

**But is the method scientifically valid?**

You are a scientific programmer analyzing ESM output. Please write a complete, self-contained Python script that performs the requested analysis.

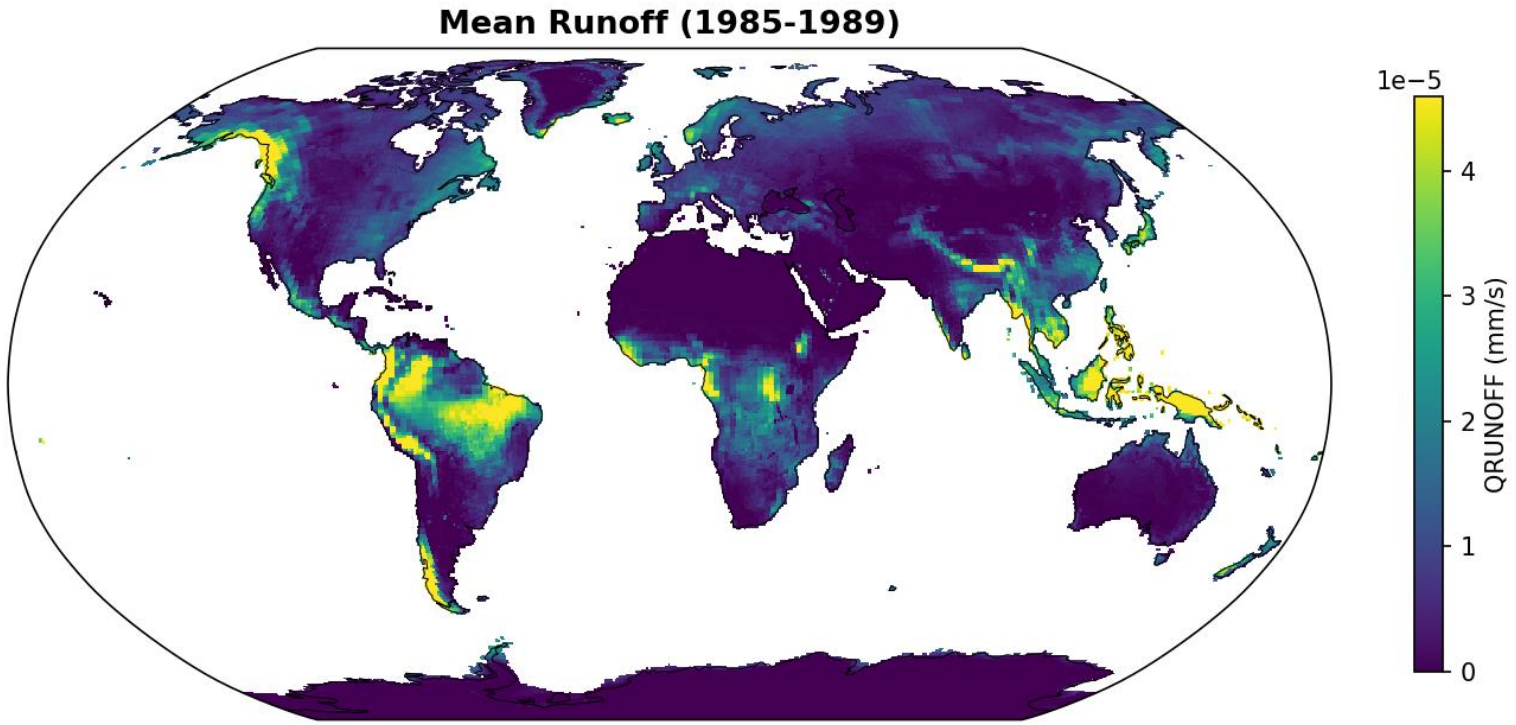
Task: Using E3SM land model monthly output, extract the climatological mean runoff over 1985–1989. The ELM data path is XXX. Extract the global QRUNOFF field for 1985–1989 and compute the area-weighted global mean. Produce a map of the mean runoff field with the global statistics overlaid.




```
global: 6.75e-06 mm/s
```

You are a scientific programmer analyzing ESM output. Please write a complete, self-contained Python script that performs the requested analysis.

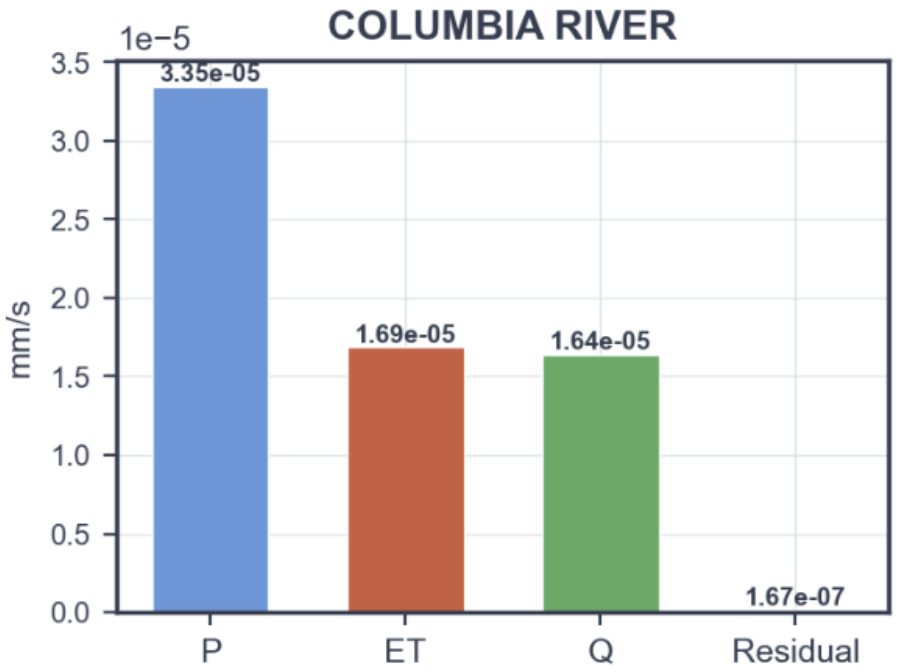
Task: Using E3SM land model monthly output, extract the climatological mean runoff over 1985–1989. The ELM data path is XXX. Extract the global QRUNOFF field for 1985–1989 and compute the area-weighted global mean. Produce a map of the mean runoff field with the global statistics overlaid.

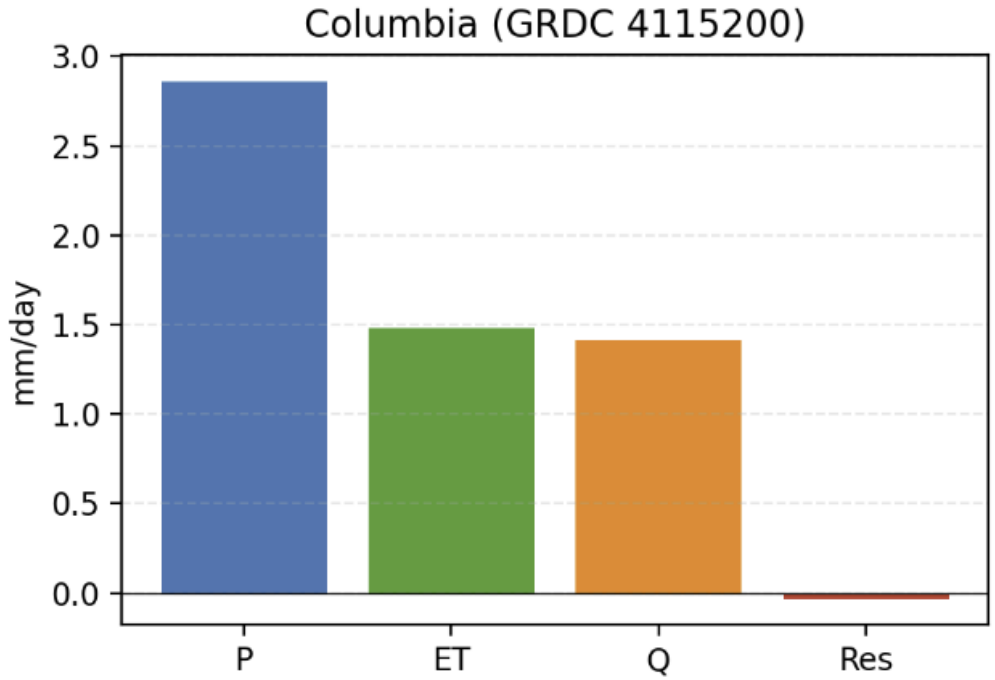



**Silent Failure!**

```
global: 6.75e-06 mm/s
```

# Silent Failure Could Lead to Opposite Conclusions




The auto-generated script omits land-fraction weighting, leading to negative water budget residual calculation.

Pacific Northwest
NATIONAL LABORATORY

**Silent failures** are runs that succeed at the software level but fail at the scientific level.

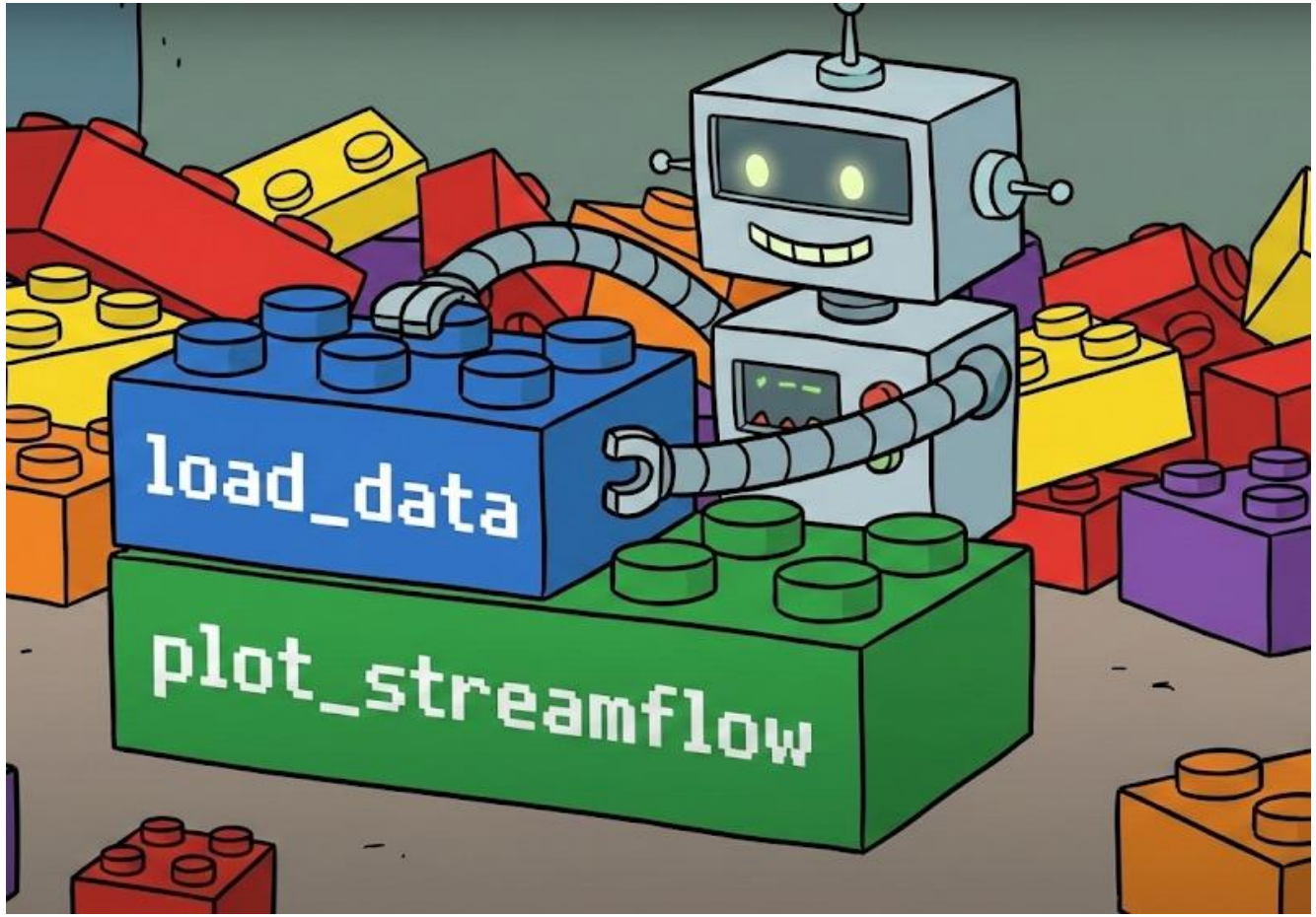
Pacific Northwest
NATIONAL LABORATORY

**Silent failures** are runs that succeed at the software level but fail at the scientific level.

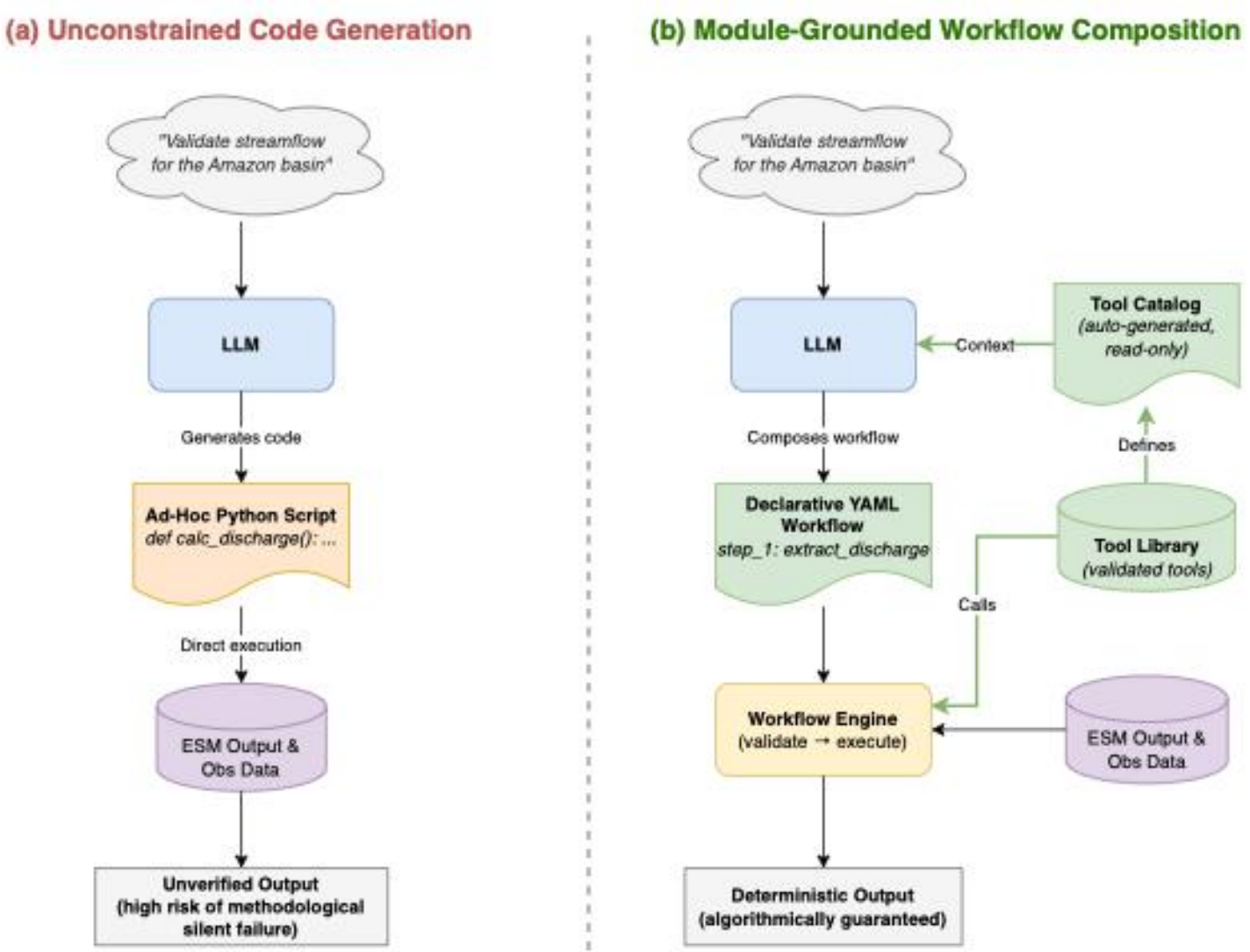
**What should we let LLM to decide?**

Pacific Northwest
NATIONAL LABORATORY

# Let the LLM compose, not code



# ESFlow: a Module-Grounded Workflow

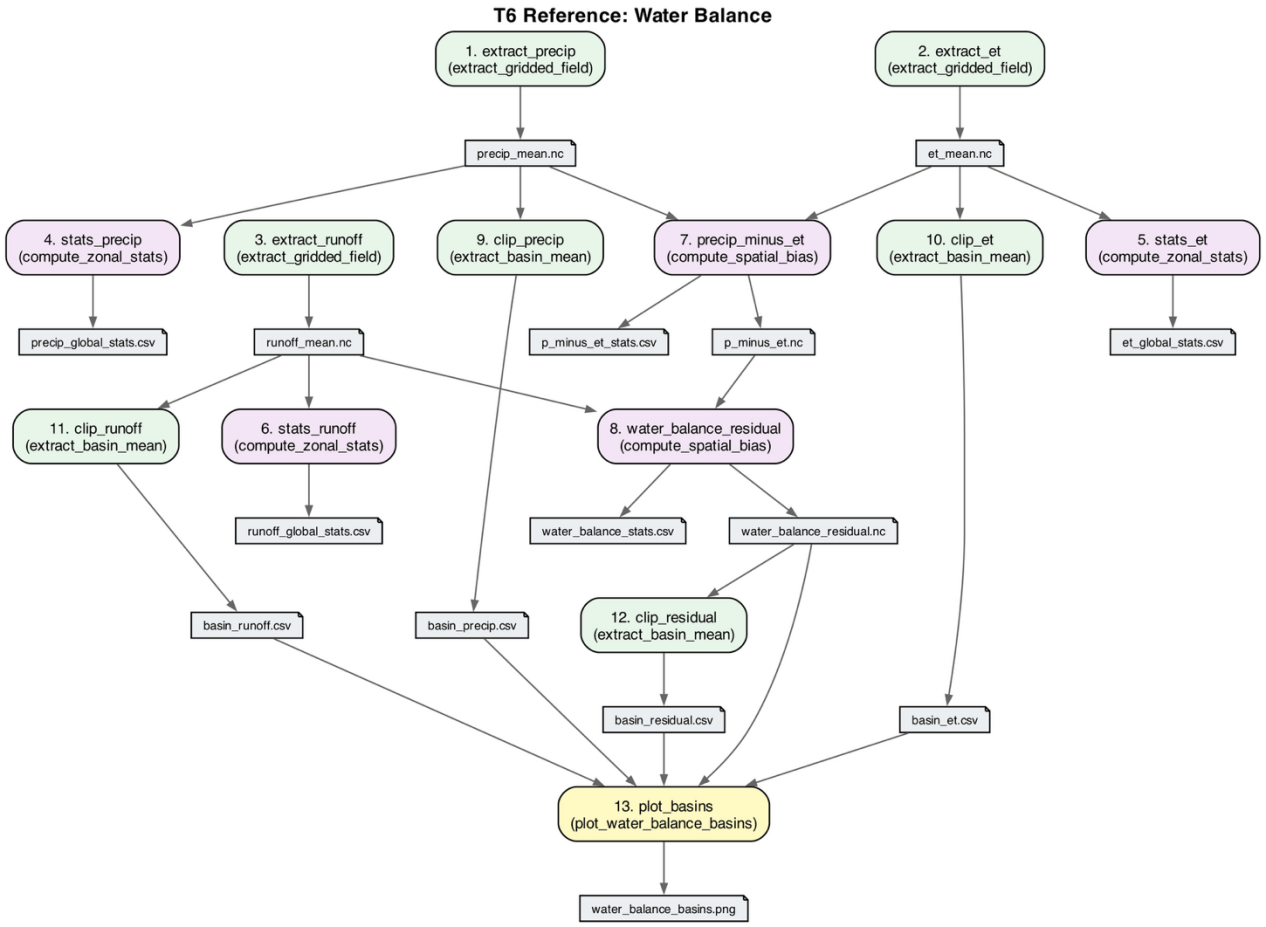


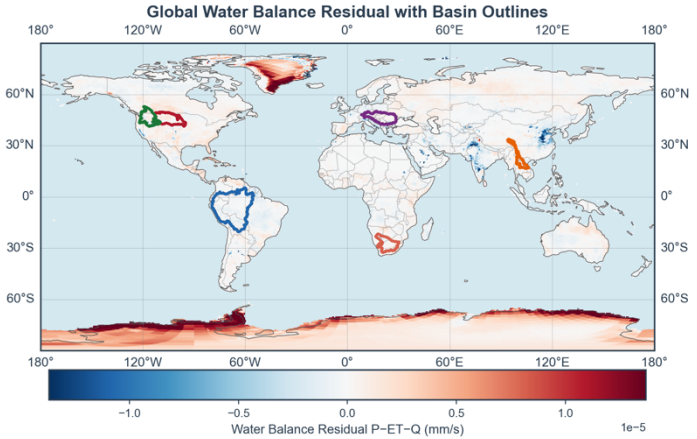

- Validated Python tools (LLMs never see)
- Tool Library describes tool function and I/O (LLM context)
- LLM generates workflow in YAML format
- A `run_workflow` engine executes the workflow by calling tools

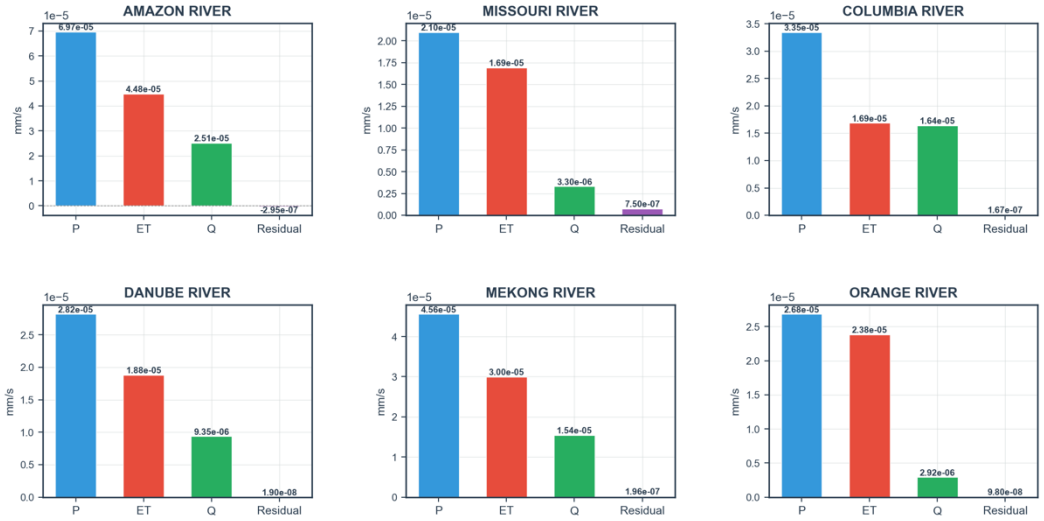
# Benchmark tests

- Two test types: baseline (direct Python generation) & module-grounded (ESFlow framework).
- Six LLMs evaluated: Phi-4, Haiku 4.5, o4-mini, Gemini 2.5 Flash, GPT-5, Opus 4.6.
- Seven hydrology diagnostic tasks – ranging from simple to complex.
- Four repeats per test run.

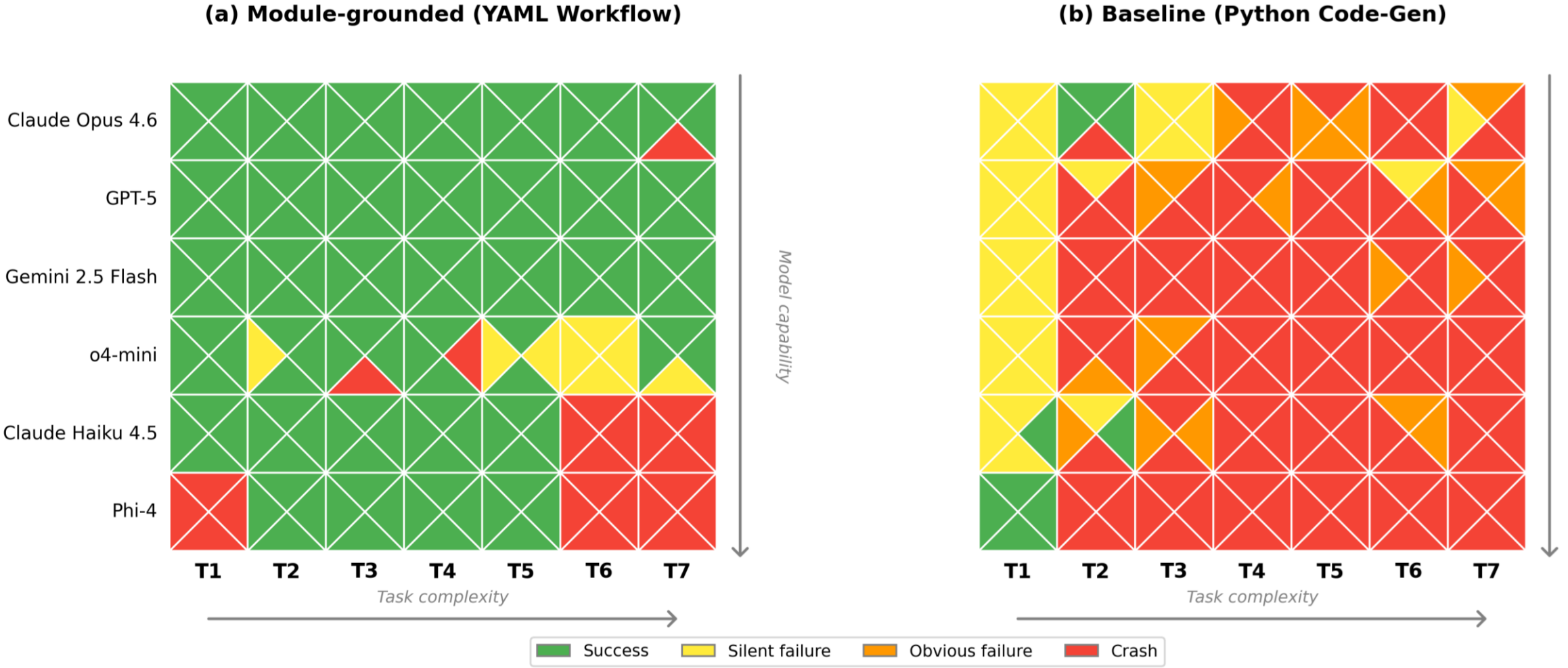
# Benchmark tests (Task 6)



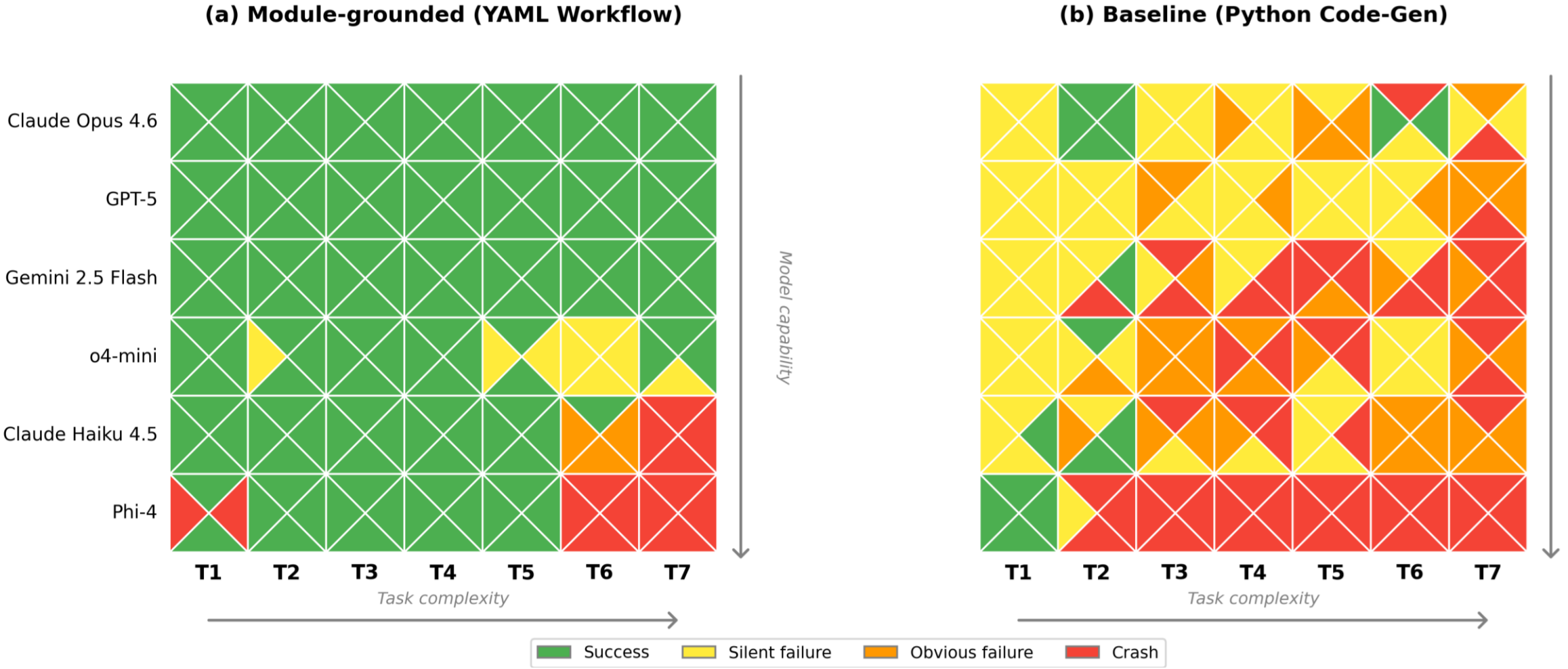
# Single shot results




**82% Success, 5% Silent Failure**

**5% Success, 16% Silent Failure**

# Self debug (up to 3 rounds) results




**85% Success, 5% Silent Failure**

**9% Success, 40% Silent Failure**

Pacific Northwest NATIONAL LABORATORY

# Can we trust LLMs for complex ESM analysis?

- Unconstrained code generation is unreliable for ESM analysis.
- Module-grounded composition achieves high-fidelity results.
- Self-debugging is safe only when the LLM is constrained.

Pacific Northwest NATIONAL LABORATORY

# Thanks for your attention!

https://doi.org/10.5194/egusphere-2026-2237
Preprint. Discussion started: 27 April 2026


EGUsphere Preprint repository

## Can We Trust LLMs for Complex Earth System Model Analysis? Silent Failure and Evidence from Module-Grounded Benchmarking

Tian Zhou[1], Yun Qian[1], and L. Ruby Leung[1]
[1]Pacific Northwest National Laboratory, Richland, WA, USA
**Correspondence:** Tian Zhou (tian.zhou@pnnl.gov)

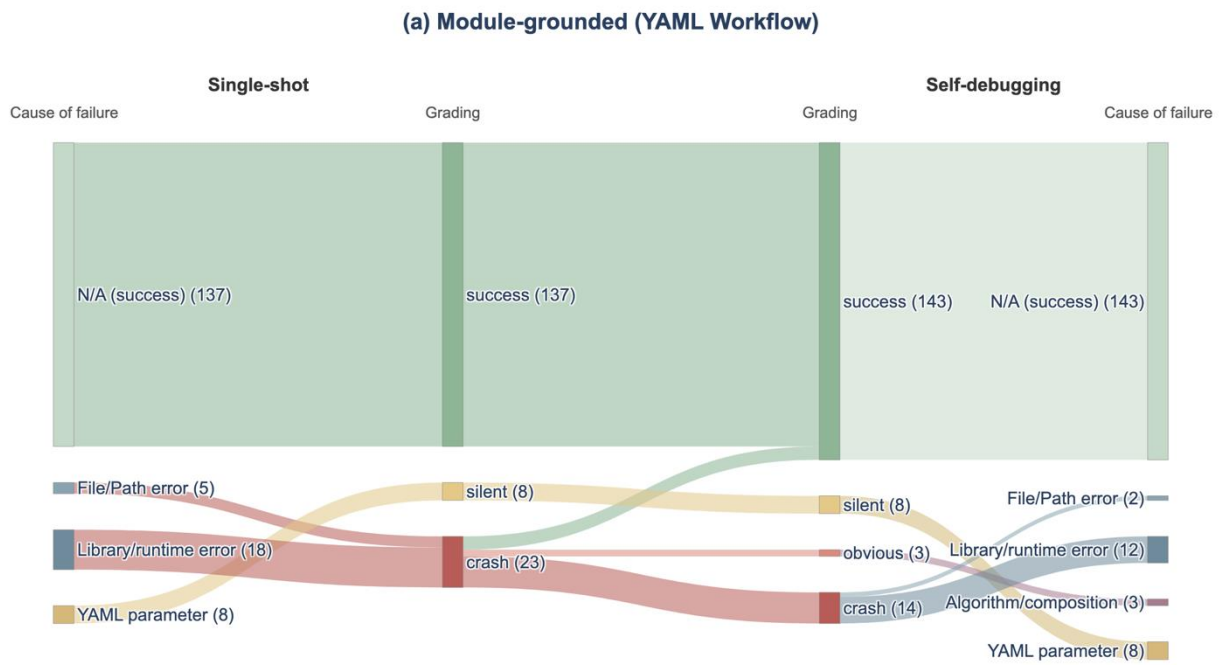
**Abstract.**

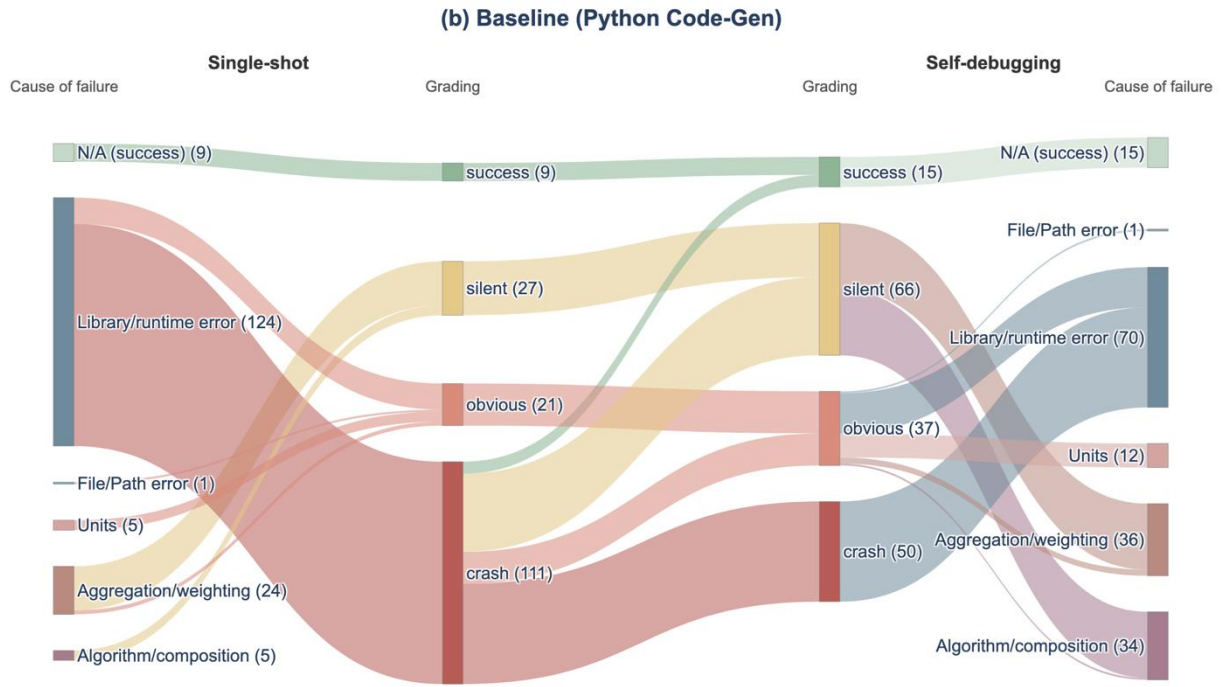
Large language models (LLMs) are becoming increasingly capable of complex scientific scripting, but this growing robustness creates a paradox: the more trustworthy their outputs appear, the more easily scientifically incorrect results can pass unnoticed. In Earth system model (ESM) analysis, such silent failures are more dangerous than visible crashes because they produce plausible figures and statistics that may be accepted without detailed inspection. We address this risk with ESFlow, a module-grounded agentic AI framework that constrains the LLM to compose workflows from validated analysis tools rather than generate arbitrary code. The LLM reads an auto-generated, self-describing catalog and outputs a YAML (human-readable data-serialization) workflow, which is then executed by a deterministic engine. We demonstrate this framework with a validated tool library for Energy Exascale Earth System Model (E3SM) land surface hydrology diagnostics in a benchmark spanning seven analysis tasks and six contemporary LLMs. Across both single-attempt runs and runs augmented with automatic self-debugging, the module-grounded approach attains an overall success rate above 80 %, maintains a low and stable silent-failure rate, and reaches 100 % success for the three high-capability models, whereas unconstrained Python code generation succeeds in only about 5 % of runs and sees its silent-failure rate rise from roughly 16 % to about 40 % under self-debugging. These results suggest that increasing LLM capability does not remove the reliability problem in scientific scripting; it makes silent failures more consequential by making incorrect outputs more convincing. The answer to the trust question posed in the title is therefore conditional: unconstrained code generation is not trustworthy for complex ESM analysis, whereas module-grounded workflow composition can be highly reliable for frontier models and remains substantially more robust under iterative self-debugging. By shifting the LLM's role from code generation to the composition of trusted tools, this framework provides a safer, more scalable architecture for AI-assisted scientific discovery that is aligned with FAIR (findable, accessible, interoperable, and reusable) principles.

Pacific Northwest
NATIONAL LABORATORY

# Backup Slide

Flow of non-success runs through single-shot grading and post-self-debug



Pacific Northwest
NATIONAL LABORATORY

# Backup Slide

## Root cause of the non-success runs

| Category | Description and representative examples |
|---|---|
| File/Path error *(execution-level)* | Crash from a wrong metadata path, missing input file, or unsupported output location.<br>• Wrong gauge-file path (Phi-4, T1)<br>• Output organised into subdirectories the engine does not create (Opus, T7)<br>• Incorrect observation-metadata path (Haiku, T4) |
| Library/runtime error *(execution-level)* | Language-level or library-level failure during script or tool execution.<br>• Calendar-type mismatch when coercing E3SM no-leap dates (multiple models, T4/T5)<br>• NumPy shape mismatch during gauge-to-grid matching, e.g. (360,) vs. (720,) (GPT-5, T4)<br>• Lazy-array evaluation failure on scalar extraction (Gemini, T7)<br>• Polygon-clipping error on self-intersecting basin geometries (multiple models, T7)<br>• Missing or incorrect import statements (Phi-4, T6) |
| Units *(methodological)* | Missing or partial unit conversion that produces plausible but numerically wrong output.<br>• Converts some variables to mm day$^{-1}$ but leaves others in mm s$^{-1}$ (multiple models, T7)<br>• Omits MODIS kg m$^{-2}$ s$^{-1}$ to mm day$^{-1}$ conversion (multiple models, T3)<br>• Applies wrong conversion factor ($\times$86.4 instead of $\times$86 400) (o4-mini, T7) |
| Aggregation / weighting *(methodological)* | Incorrect temporal aggregation, spatial weighting, or land/ocean masking.<br>• Daily-to-monthly aggregation inconsistent between model and observations (multiple models, T2/T4)<br>• Missing land-fraction weighting, causing 1–2 % drift at coastal cells (multiple models, T6)<br>• Area-weighted mean over all cells including ocean zeros, diluting land-only values (multiple models, T3)<br>• Drops cells with negative runoff before averaging, biasing upward (Gemini, T6) |
| Algorithm / composition *(methodological)* | Wrong analytical method, missing workflow step, or incorrect metric definition.<br>• Bilinear interpolation instead of conservative remapping (multiple models, T3)<br>• Un-normalised Wasserstein distances reported as raw values, e.g. 54 144 vs. reference 0.32 (Opus, T7)<br>• P−ET reported instead of P−ET−Q, omitting the runoff-subtraction step (Haiku, T6)<br>• Hand-rolled spherical area weights instead of model-native area variable (GPT-5, T6)<br>• Longitude convention mismatch (0–360° vs. −180–180°) producing all-NaN fields (Phi-4, T3) |
| YAML parameter *(parameter; module-grounded only)* | Structurally valid workflow that invokes a validated tool with incorrect arguments; the tool executes normally but on wrong inputs.<br>• `years: [1985, 1989]` (two-element list) instead of the full five-year range, loading only 2 of 5 years (o4-mini, T5/T6/T7)<br>• Wrong variable name passed to a correctly selected extraction tool (o4-mini, T4) |