


Applying Large Language Models in Hydraulic and Hydrologic Engineering

William Katzenmeyer, P.E., C.F.M



Why Large Language Models

Large Language Models are at the frontier of AI/ML research and capability, with widely available consumer interfaces



Hundreds of Billions of continued capital investment is underway to continue improving capabilities.



Test Time Compute and Reinforcement Learning on Reasoning Tasks push the frontier for science reasoning and coding

What Am I Doing with LLM's?

Building HEC-RAS Automation Tools

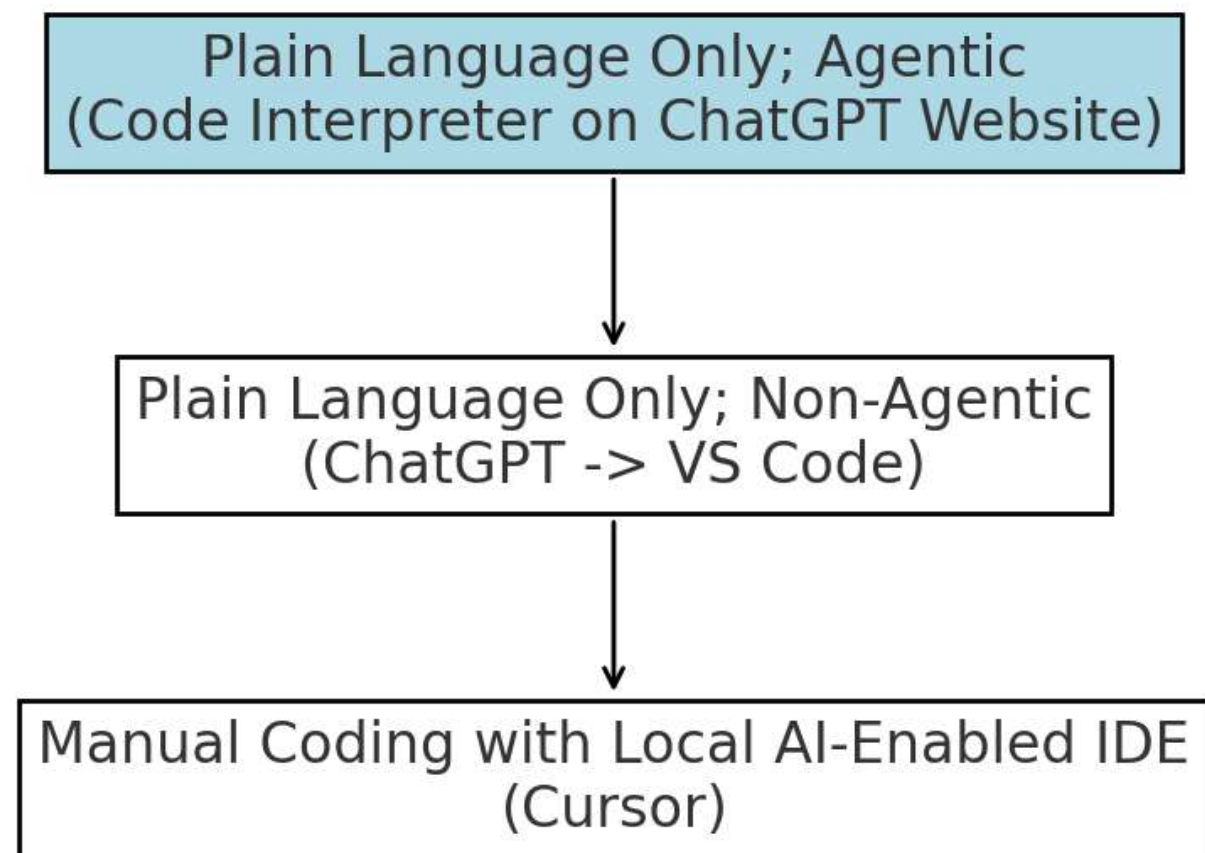
- HEC-Commander Tools
- RAS-Commander Library
- Lots More!

Daily Driver Tasks

- Writing Scripts
- Web Research
- Software Assistance
- Report Editing and Revision



3 Levels of Interaction for Leveraging Code with LLM's



- Writing and Research Assistant
 - Distilling and Summarizing
 - Addressing Grammar, Voice and Structure
 - Drafting and Editing Reports
 - Create QR Codes and Perform Research
 - Data Analysis and Figures
- Software Documentation Expert
 - User Guide with Active Reasoning
 - "How do I do this in Software Y"
- Fully Code-Enabled Assistant
 - Code Interpreter (Generic Assistant)
 - Small Helper Scripts
 - Converting GUI Workflows to Python
 - Agentic Task Automation
 - Running Simple Scripts Locally
 - Coding in your local Development Environment

HEC-Commander Repository

Open-Source Notebooks



HEC-Commander
Repository (GitHub)

Blogs



Chat GPT



HMS-Commander

AI-Coded Jupyter Notebook Supporting

- Subbasin Parameter Editing
- DSS Output File Renaming
- Impervious Grid Scaling > 1.0
- Calibration Regions by shapefile
- CSV File Input
- Enables Linked HMS>RAS Calibration Workflows
- Modular Script Ready for AI-Editing for Bespoke Applications



HMS – Commander



Example CSV Input for HHS-commander & RAS-Commander 2D Infiltration Overrides

User run number From csv	Initial Deficit scale	Maximum Deficit scale	Percolation rate scale	Impervious area scale	Recession factor	Initial flow area ratio	Threshold Flow to Peak ratio	Time of concentration scale	Storage Coefficient scale
1	1.0	1	0.06	1	0.1	1	0.1	1	1
2	0.9	1	0.06	1	0.1	1	0.1	1	1

RAS-Commander

Parallelizing HEC-RAS across multiple networked machines via a Jupyter Notebook

RAS-Commander 1.0 (GUI Version)

- Parallel Execute Local + Remote Machines using Psexec
- Build Plans from DSS (HMS>RAS1D, RAS1D>RAS2D)
- Override Infiltration Base Parameters by CSV

Author: William (Bill) Katzenmeyer, P.E., C.F.M. (C.H. Fenstermaker)

Notable Contributions: Sean Micek, P.E. Prototype Infiltration HECRAS

Source: <https://github.com/billk-FM/HEC-Commander-Tools>

#1 ----- USER INPUTS (These inputs are defaults and are not to be changed)

#2 ----- Additional Settings, Paths and Variables ...

#3 Required Import Statements ...

#4 ----- TKINTER GUI ----- ...

#4 Check for psexec64.exe, get password, and test connection

#5 Populate RAS Project Name and Other Paths ...

#6 If Build from DSS, copy template folder to output folder

#7 If Build from DSS, Create Proposed Plan and Unsteady

#8 If "Build from DSS", update the plan (.pxx), unsteady

#9 If Enable_Infiltration_Overrides = True, Copy Geomtr

#10 Delete Existing Files and Directories in HECRAS_Depl

#11 Create Absolute Path Batch File for each Plan File f

#12 Queue and Execute each plan via batch file ...

#13 RESULTS POSTPROCESSING: Copy all matching files from the list of directories to a single directory, replace if newer Then, prompt user and delete temporary

AI-Coded Jupyter Notebook Supporting:

- Parallel HEC-RAS Execution
- Windows Native: Supports All Versions
- Leverage Multiple Workstations in Parallel
- Open Source, MIT License

Basic Components

- User Input and Settings
- New!** Tkinter GUI
- File Deploy and Copy
- Batch File Creation
- Command Line Execution
- Results Collection

Flexible Operation:

- Bring Your Own Project
- Create Plans from HMS DSS Input Files
- Optional 2D Infiltration Overrides

RAS-Commander is ready for AI-Assisted editing by end users to support their bespoke applications

GPT Assistants



HEC-Commander Repository Assistant

By William Katzenmeyer +1

Expert in HEC-Commander Scripts and Automation of HEC-HMS, HEC-RAS, and Visualization of DSS Files

How do I use a specific Python notebook in HEC-...

Can you explain this section of the README?

What does this part of the documentatio...

Help me understand this code snippet from...



Water Resources Python Notebook Assistant

By William Katzenmeyer +1

Water resources engineer and python coding expert, oriented towards typical office workstation setups

Help me debug this Python code

How do I use pathlib in Python?

Suggestions for naming Python variables

Best practices in organizing Python notebooks



Voice Notes Transcription Assistant

By William Katzenmeyer +1

Transcription and editing assistant for voice notes.

Transcribe and edit my voice note:



GIS Autonomous Assistant

By William Katzenmeyer +1

Helpful GIS Assistant using Code Interpreter

Merge the polygons in the uploaded shapefile

Provide Land Use Statistics from the provided land use...

Scan the uploaded raster file and identify any data...

Create a pandas dataframe using the uploaded...



Python HDF File Assistant

By William Katzenmeyer +1

Python assistant with HDF Python library documentation



USGS API Assistant

By William Katzenmeyer +1

The knowledge base of this assistant is based primarily on the [kapadia/usgs github library](https://github.com/kapadia/usgs)



Geospatial Python Notebook Assistant using WBT

By William Katzenmeyer +1

Python Notebook Assistant with access to WBT User Manual and Python



Python Expert: Hydroclimate Data Retriever Tools

By William Katzenmeyer +1

Expert in Use of Hydroclimate Data Retriever Python Packages

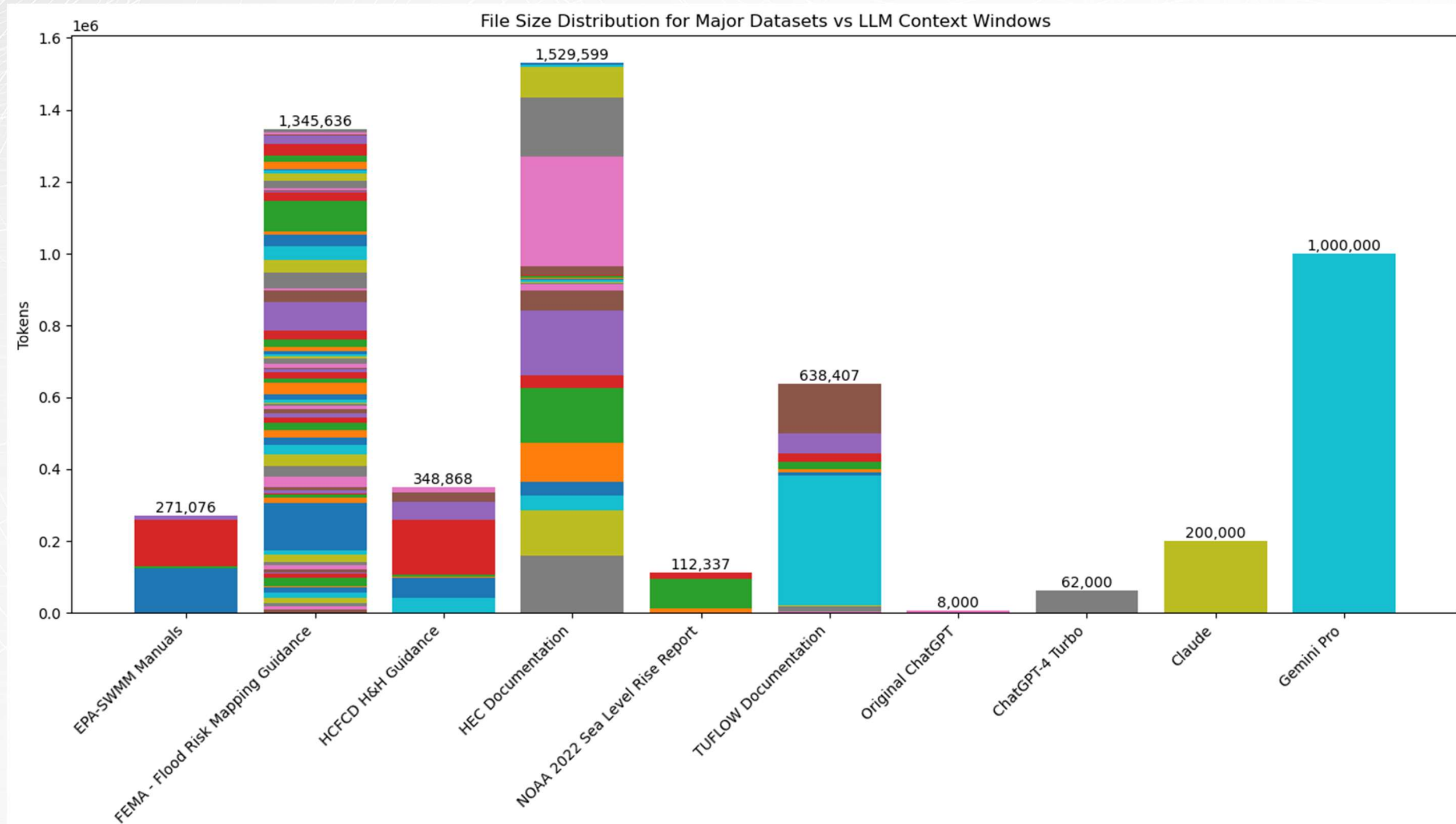


RAS Commander Library Assistant

By William Katzenmeyer +1

Helpful assistant for the RAS Commander (ras-commander) Library and Expert at HEC-RAS Automation

Context Window Limitations



Avoiding the Pitfalls and Limitations of LLM's



1
False Recall (Hallucinations)



2
Confident, Plausible
Errors/ High Error Rates



3
Non-Deterministic
Operation



4
Limited Input and Output



5
Rapidly Changing Technologies,
Terminology and Platforms



6
Data Privacy and Confidentiality

Neural Networks are incredible at Compression

BUT Compression leads to Errors and Hallucinations

Neural Networks compress language data into latent representations, but it's a lossy compression. A model's weights are best relied upon for reasoning and not its recall.

This lossy recall manifests as made-up references, misquotes and other incorrect information

The best LLM platforms provide the models with specific context and valid links for the model to formulate a response, such as:

- **Perplexity**
- **OpenAI and Google “Deep Research” Tools**

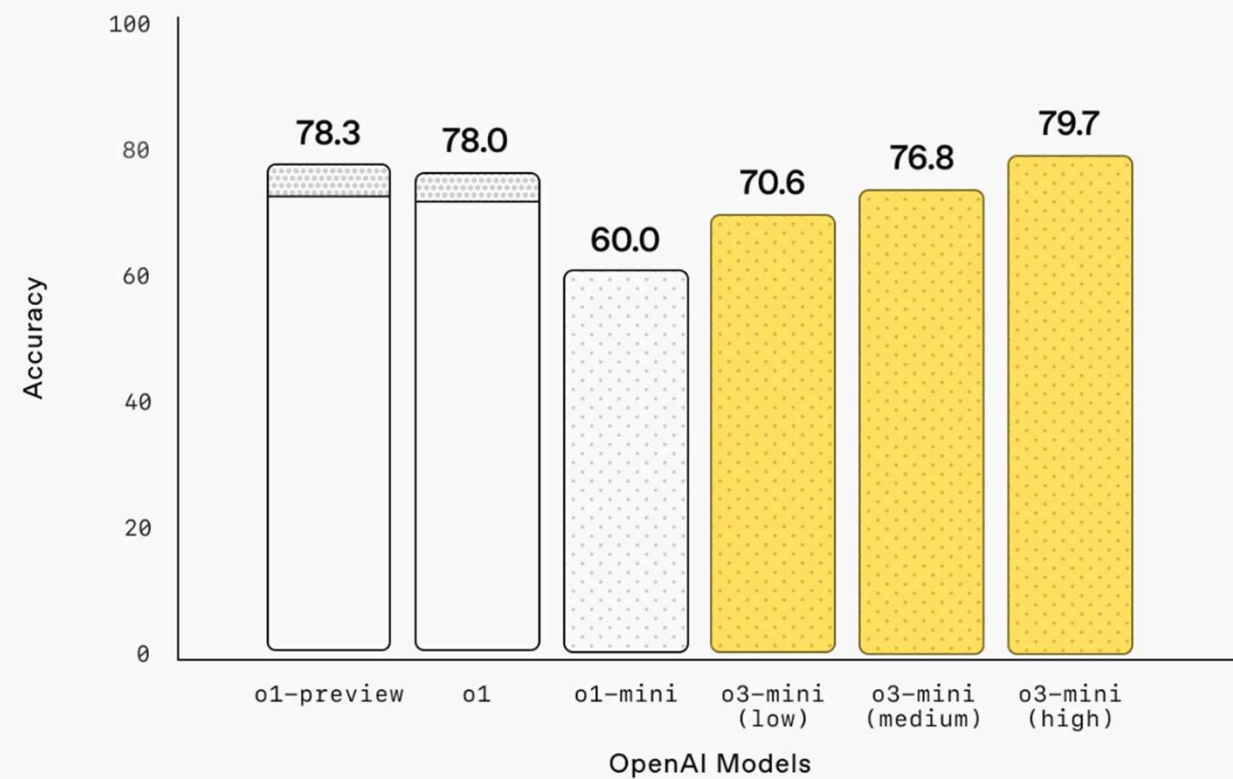


This Photo by Unknown Author is licensed under CC BY

To reduce errors and hallucinations, providing accurate context is key!

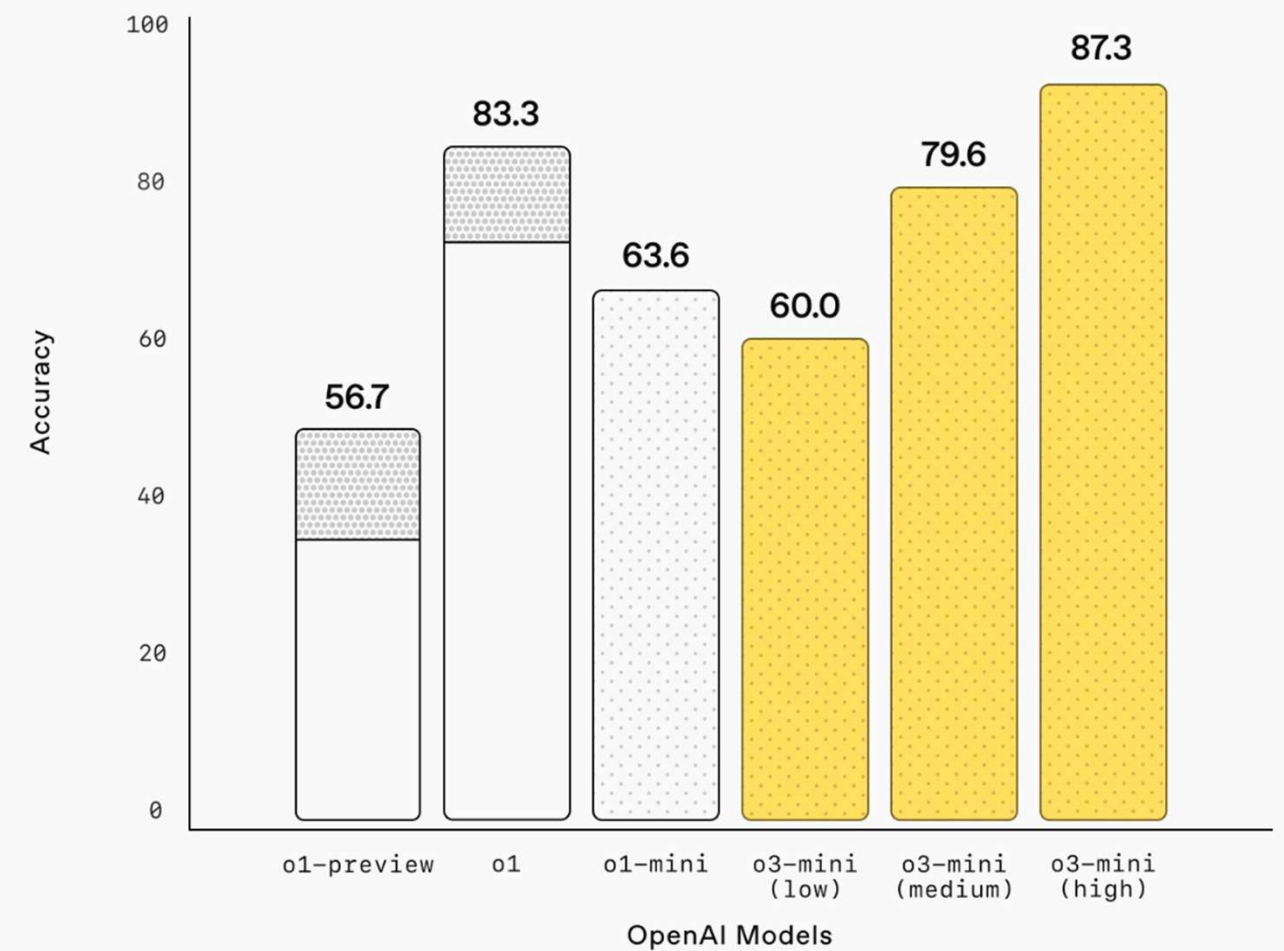
Tempering Expectations

PhD-level Science Questions (GPQA Diamond)



PhD-level science: On PhD-level biology, chemistry, and physics questions, with low reasoning effort, OpenAI o3-mini achieves performance above OpenAI o1-mini. With high effort, o3-mini achieves comparable performance with o1.

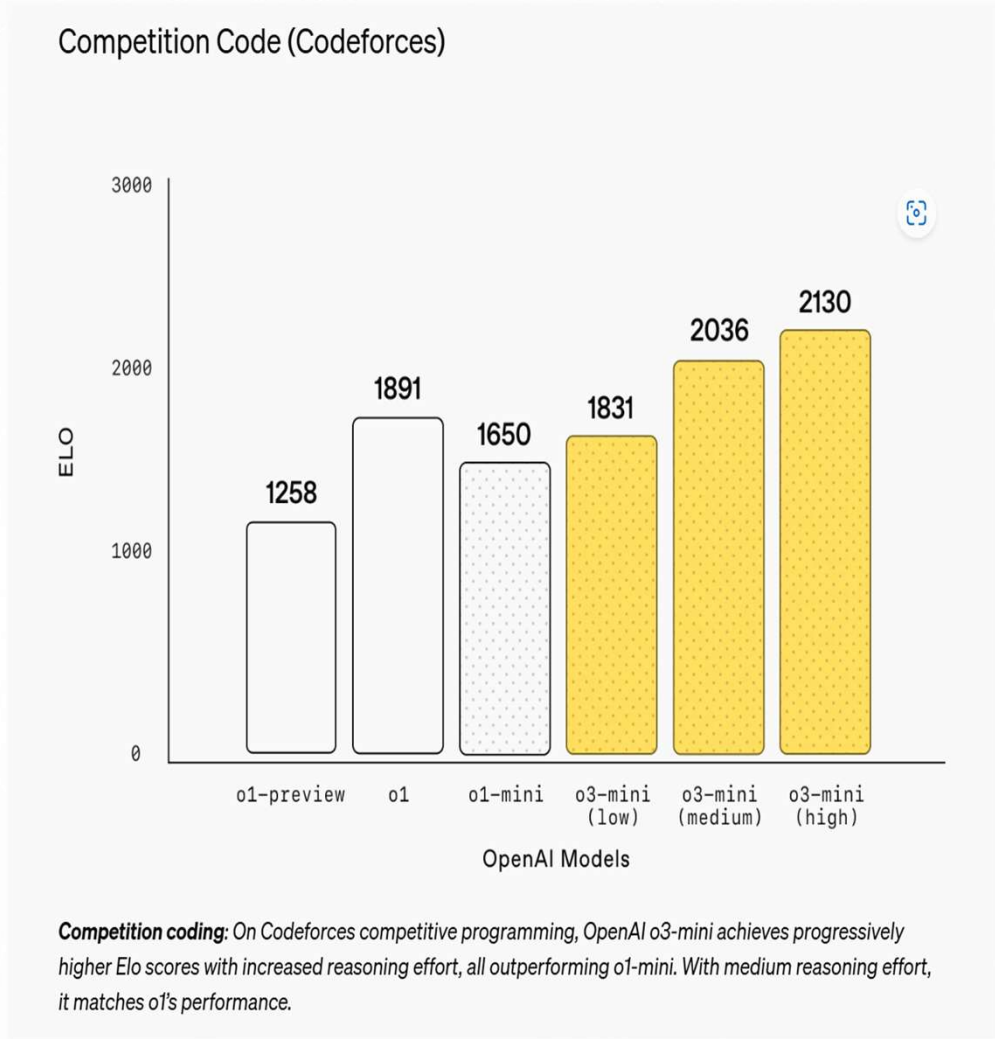
Competition Math (AIME 2024)



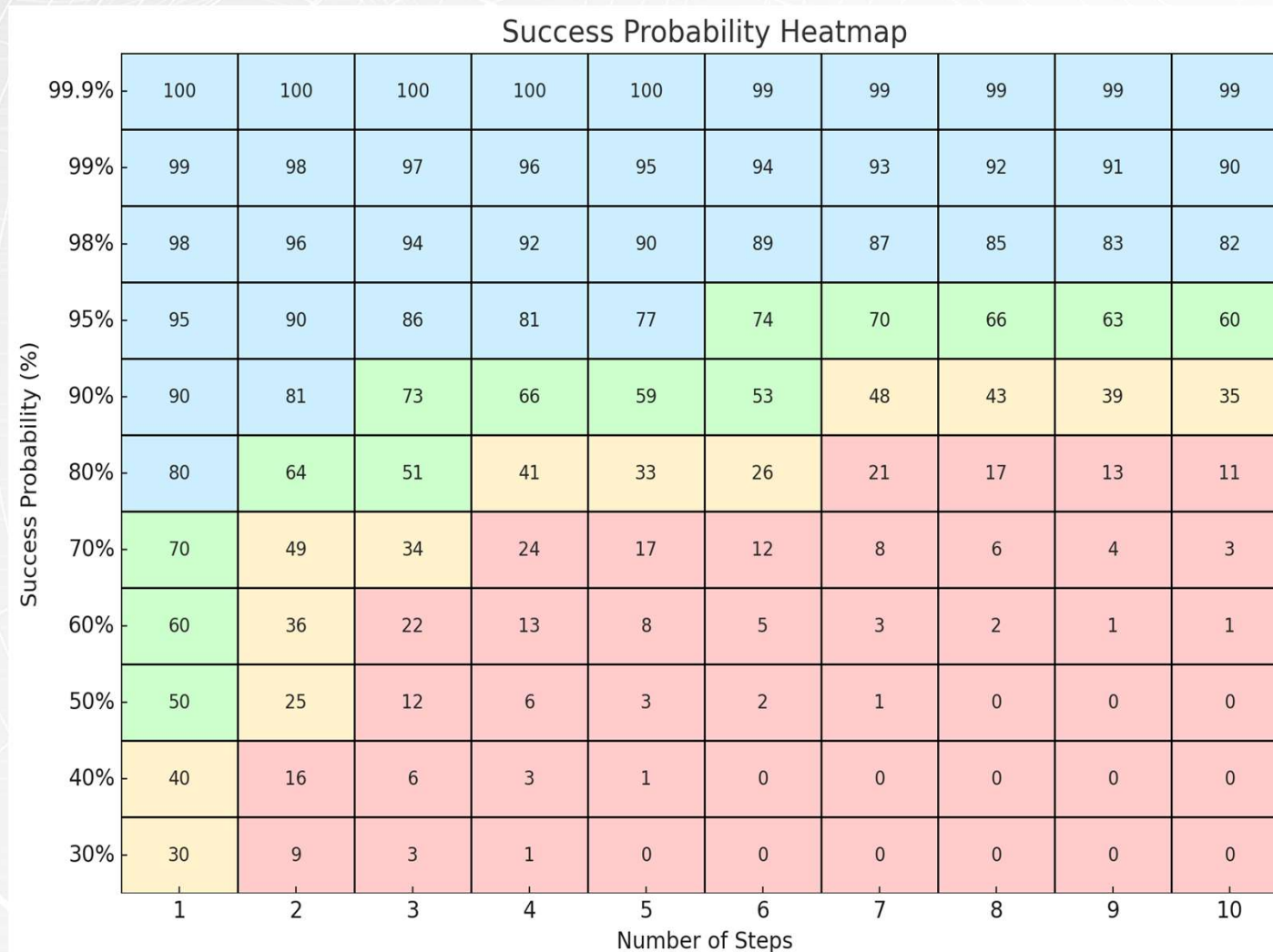
Coding Performance Scores

Benchmark (Metric)	DeepSeek V3
Architecture	MoE
# Activated Params	37B
# Total Params	671B
MMLU (EM)	88.5
MMLU-Redux (EM)	89.1
MMLU-Pro (EM)	75.9
DROP (3-shot F1)	91.6
IF-Eval (Prompt Strict)	86.1
GPQA-Diamond (Pass@1)	59.1
SimpleQA (Correct)	24.9
FRAMES (Acc.)	73.3
LongBench v2 (Acc.)	48.7

HumanEval-Mul (Pass@1)	82.6
LiveCodeBench (Pass@1-COT)	40.5
LiveCodeBench (Pass@1)	37.6
Codeforces (Percentile)	51.6
SWE Verified (Resolved)	42.0
Aider-Edit (Acc.)	79.7
Aider-Polyglot (Acc.)	49.6
AIME 2024 (Pass@1)	39.2
MATH-500 (EM)	90.2
CNMO 2024 (Pass@1)	43.2
CLUEWSC (EM)	90.9
C-Eval (EM)	86.5
C-SimpleQA (Correct)	64.1



Agentic Operation and the March of 9s



Benchmarks are starting to represent the difficulty that we encounter in practice and are beginning to approach success rates that allow multi-step agentic behavior.

- *X Axis:*

Number of Sequential Steps

- *Y Axis:*

Success Probability for Single Response

- *Heatmap shows probability of success of multiple steps @ similar difficulty.*

Prompt Engineering

- The most impactful tips and tricks for improving your prompts generally revolve around providing the AI custom instructions and context

When prompting an LLM, focus on

- Providing clear, well-structured directions
- Use Delineators to Separate Instructions from Context
- **Understand the Limitations**
 - Limited Context Windows
 - Limited Retrieval from Large Documents
 - Probabilistic Operation, not Deterministic
 - File size and library limitations in Code Interpreter
 - No internet access (blame the AI safety patrol)

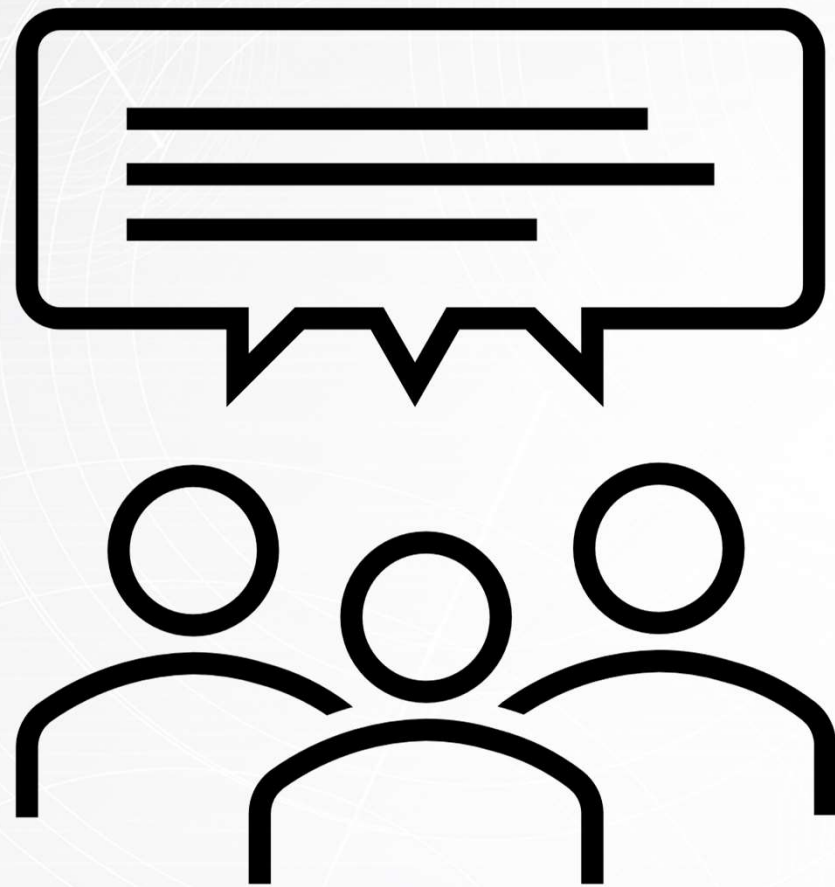
Basic “prompt engineering” is typically:

- Role (Persona)
 - Constraints
 - Contextual Data
 - Instructions
 - Desired Output
 - Examples
- Prompts can also be structured as code



Your AI assistant does not know what it is doing here, unless you tell it.

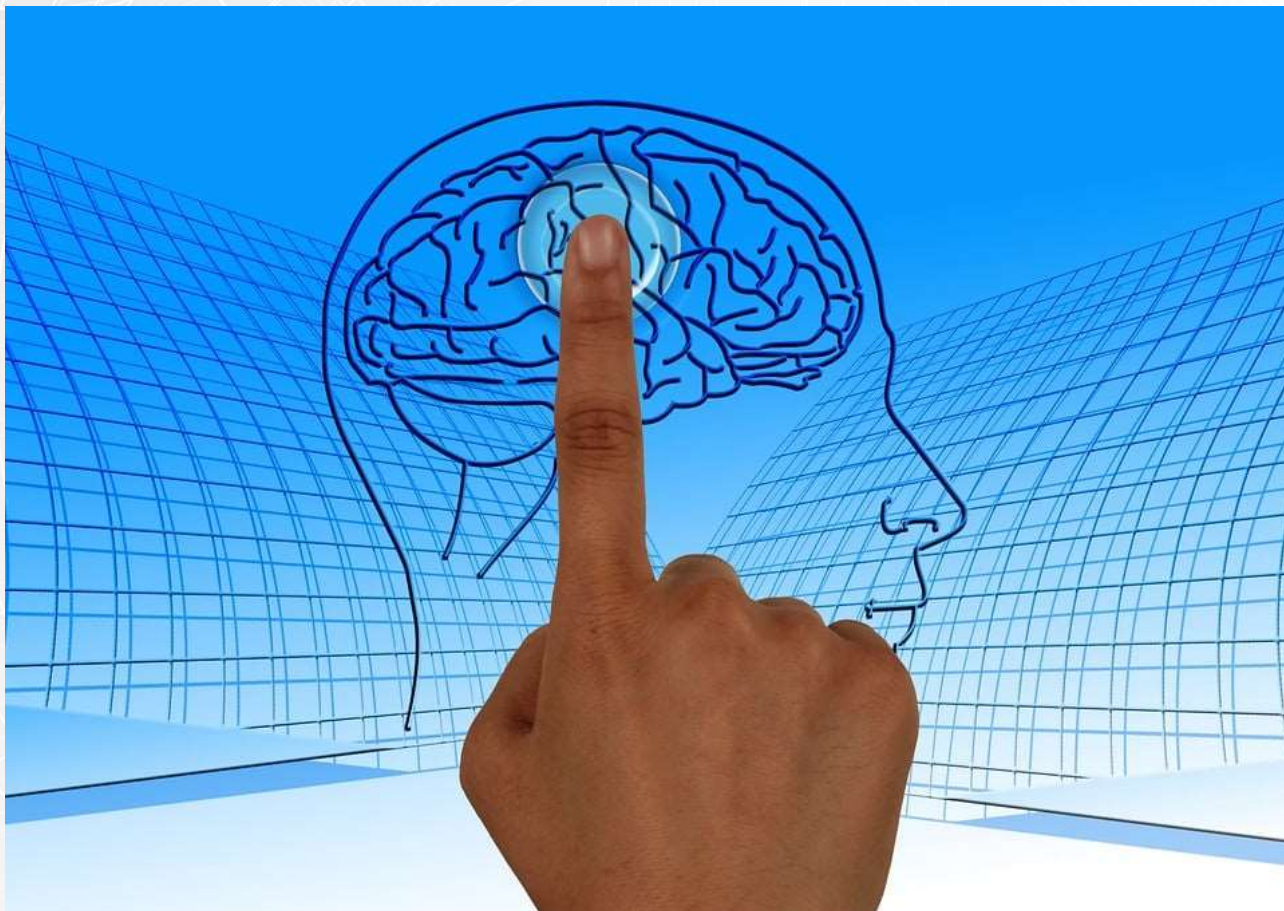
Maintaining Responsible Charge of Your Work



None of our tools are perfect. As a licensed professional, you are solely responsible for maintaining direct control and responsible charge of all aspects of your work product. Human review of all outputs will remain a steadfast requirement regardless of the advancements in Large Language Model capabilities.

As LLM's become more and more reliable, spotting the errors will require additional diligence. With better tools at our disposal, we get to spend more time on higher value tasks by directing agents to perform repetitive, verifiable tasks, data analysis, and bespoke software development.

Learn more



[HEC-Commander Tools Repository on Github](#)

[RAS-Commander Library on GitHub \(Coming Soon\)](#)

[Full Momentum Vodcast](#)

[Full Momentum Episode 33: The Future of leveraging HEC-RAS with Automation and AI](#)

[Australian Water School Free Webinars on Youtube](#)

[AI Tools for Modeling Innovation](#)

[Applying AI to HEC-RAS Modeling Workflows](#)

[Australian Water School Paid Webinars and Courses](#)

[Short Course: AI Applications to HEC-RAS modelling workflows](#)

[Full Course: AI-assisted coding for water modellers](#)

[Youtube Demonstrations](#)

https://www.youtube.com/@GPT_Commander

Q&A Discussion