

CPPM Seminar May 18th 2026

Syntax, Not Science: Safeguarding Scientific Integrity via the Scientist-AI-Loop

Based on [arxiv:2603.18145](https://arxiv.org/abs/2603.18145)

Nico Schuster

+ collaborators

Andrés N. Salcedo, Simon Bouchard, Dennis Frei, **Alice Pisani, Julian E. Bautista, Julien Zoubian, Stephanie Escoffier**, Wei Liu, Georgios Valogiannis & Pauline Zarrouk



CPPM Seminar May 18th 2026

Or short:
Vibe-coding for Scientists

Based on [arxiv:2603.18145](https://arxiv.org/abs/2603.18145)

Nico Schuster

+ collaborators

Andrés N. Salcedo, Simon Bouchard, Dennis Frei, **Alice Pisani, Julian E. Bautista, Julien Zoubian, Stephanie Escoffier**, Wei Liu, Georgios Valogiannis & Pauline Zarrouk



Outline

- 1. Introduction:** Scientist vs. AI Competence
- 2. First Look:** Visualizing Gravitational Lensing
- 3. The SAIL Framework:** Scientist-AI-Loops
- 4. The Coding Workflow:** From Prompts to Agents
- 5. Showcase II:** Exploring the Cosmic Web
- 6. Impact:** Outreach, Teaching & Sandboxing
- 7. Conclusions**

Scientists vs. AI

Scientists

- Years of studying a particular topic to gain expert knowledge
- Experts in their field, excel at scientific logic and modeling
- Deductive scientific reasoning

- „Part-time coders“ with „full-time deadlines“

AI/LLMs

- Artificial Intelligence / Large Language Models often trained for a more general purpose
- „Jack of all trades, master of none“
- Probabilistic nature: might prioritize „likely“ over „correct“

And let's face it...

- Master of syntax: generates working codes in mere seconds to minutes

Probabilistic vs. Deductive

- **An Illusion of Competence:** LLMs optimize for working syntax
 - > results might look perfect but could be fundamentally flawed
- **Silent Failures:** AI might hallucinate logic /discard physical laws in favor of optimized code
- **Knowledge Deficit:** LLMs can excel at „textbook“ science, but fail at more advanced cases
- **Context Limitations:** LLMs may prune existing logic to fit context windows

Defining Roles

The scientist as the architect:

- Introduces necessary ideas and concepts
- Defines governing equations and enforces boundaries
- Identifies limitations, simplifications and failures

The AI as the syntax engine:

- Responsible for code implementation
- Can suggest improvements, e.g., graphics rendering
- Performance optimization

Not Scientist vs. AI, but Scientist & AI

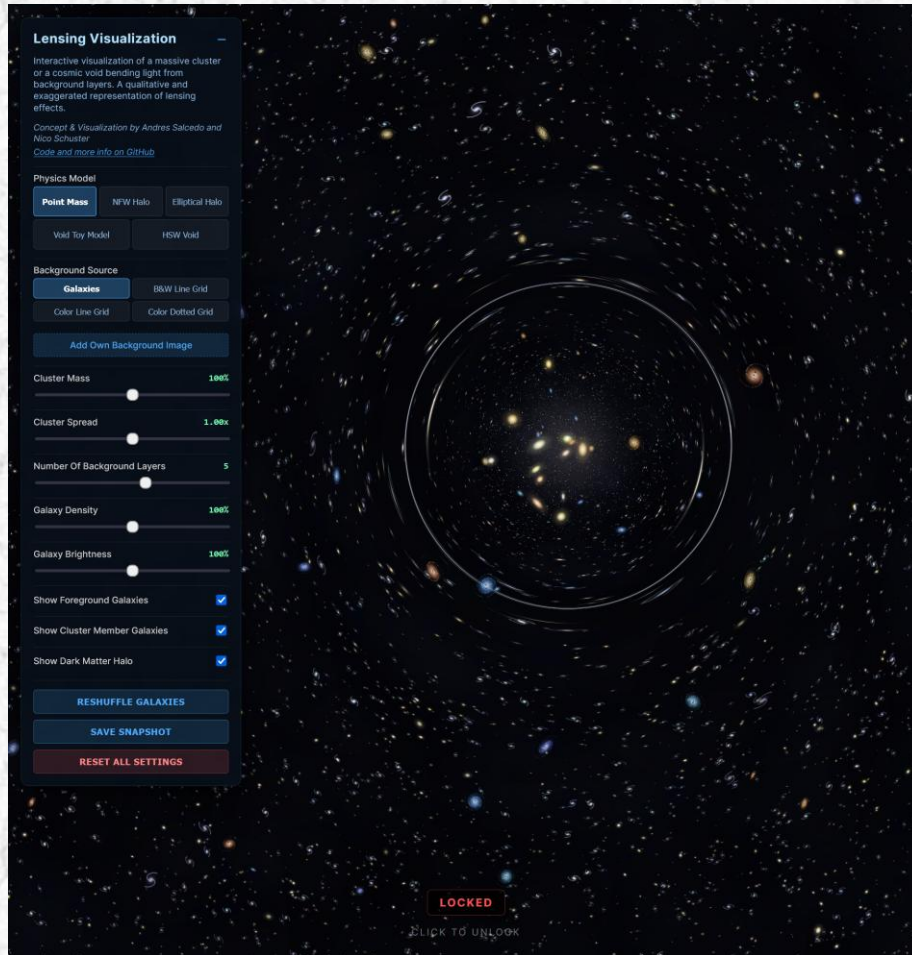
The Why

By efficiently working with AI/LLMs:

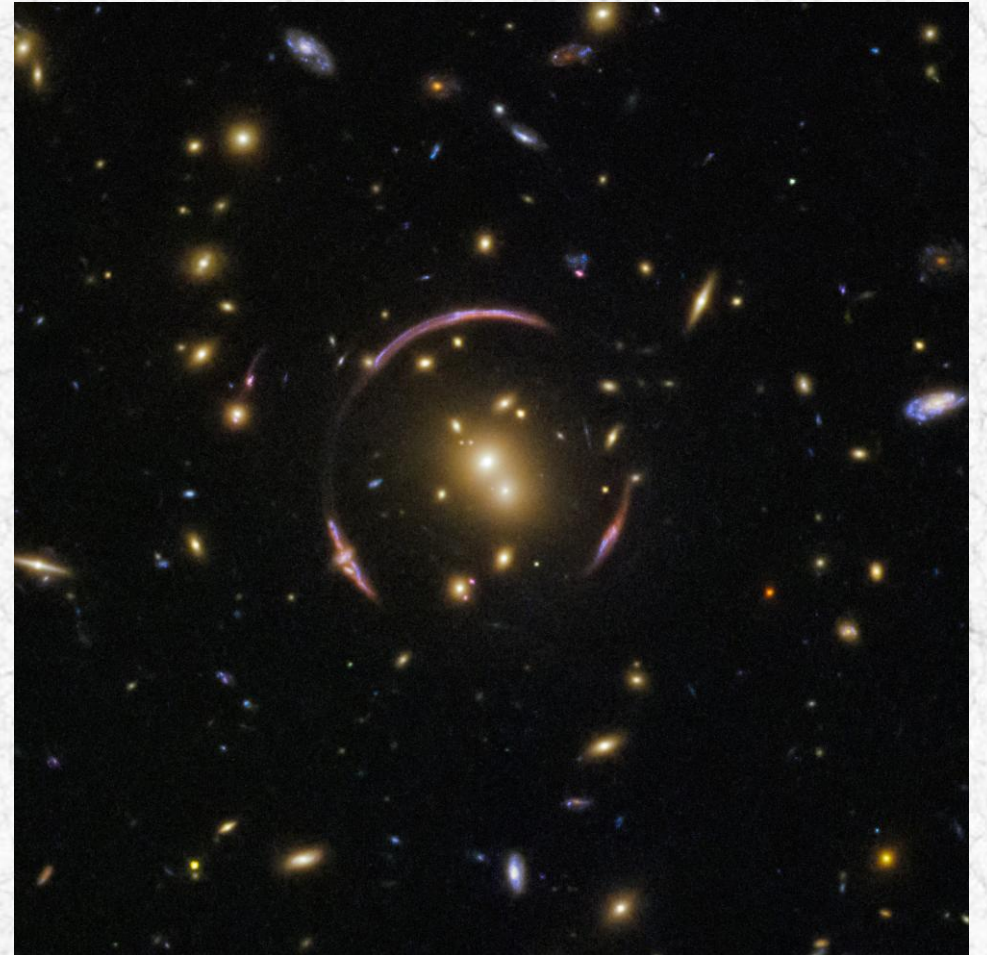
- Huge gain in coding speed
- Validated science independent of the discipline
- Creation of tools we otherwise would not have time/knowledge for
- Immediate accessibility
- Theoretical sandbox / prototyping of ideas

Gravitational Lensing I

Einstein Ring for a Point Mass distribution:

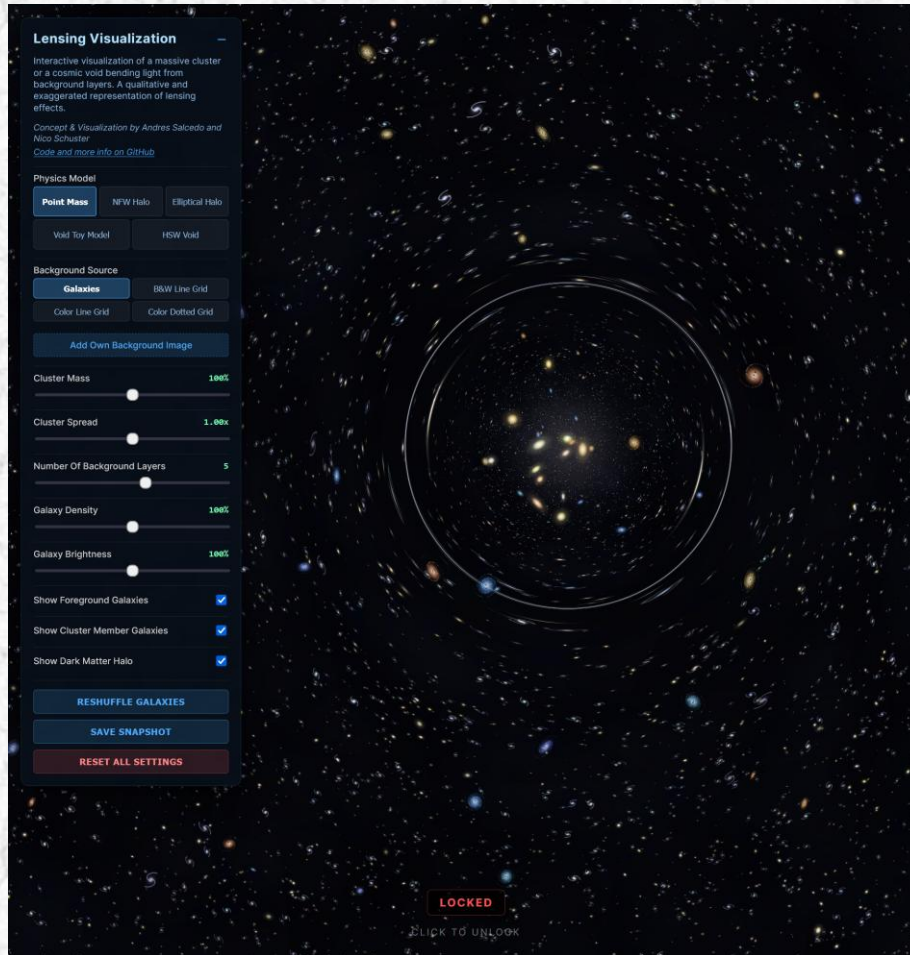


Einstein Ring around [SDSSJ0146-0929](#):

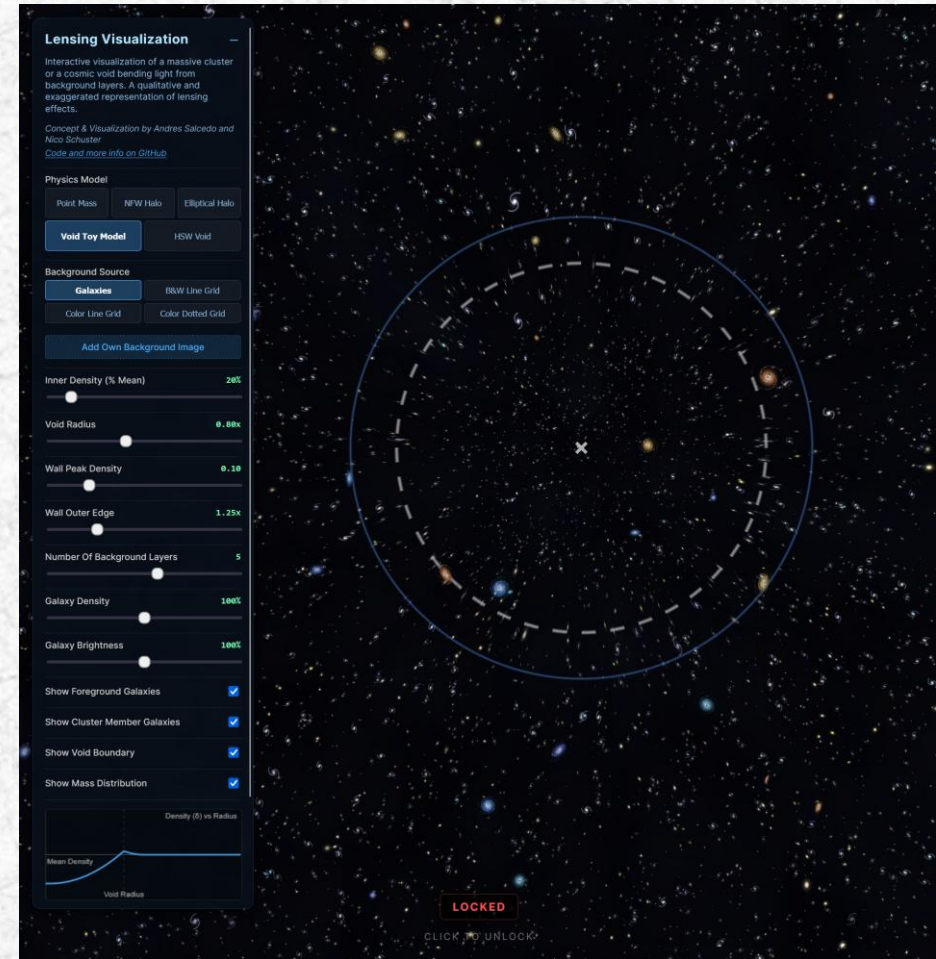


Gravitational Lensing II

Einstein Ring for a Point Mass distribution:



Lensing around cosmic voids:



Gravitational Lensing III

Live Demo:

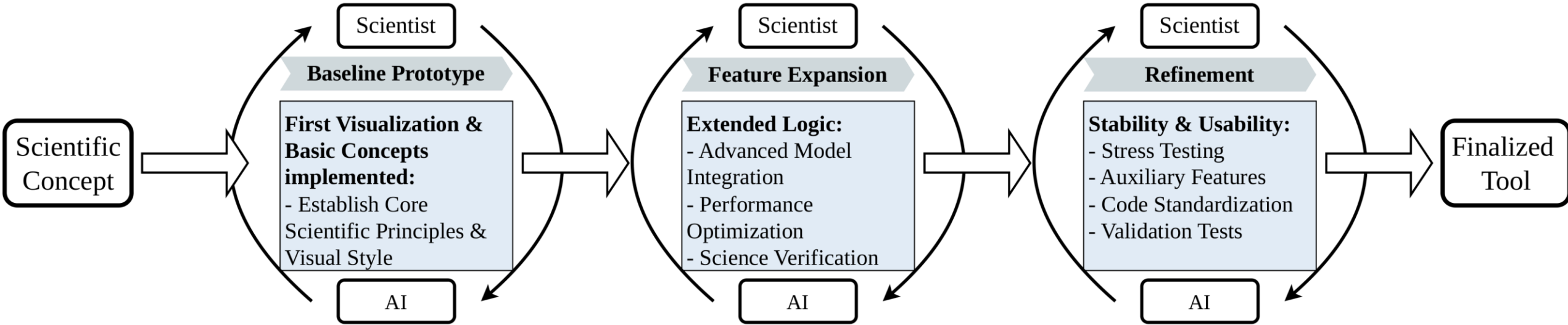


tinyurl.com/lensing-visualization

or

https://nicosmo.github.io/lensing_visualization

The SAIL Framework



The Scientist-AI-Loop:

- A variation of the „Human-In-The-Loop“ model
- Decouples science and syntax
- Each phase relies on continuous & iterative feedback between the Scientist and the AI model
- Model agnostic and applicable to any (scientific / expert driven) coding

The SAIL Framework I

Phase 1: Baseline Prototype

- **Establish foundation:** Start with core scientific principle (e.g., gravitational lensing) and constraints (e.g., governing equations). **Be precise and try multiple times!**
- **Define use:** Purpose (e.g., teaching or prototyping) and limitations (runtime, in-browser, etc.)
- **Rapid results:** Goal is to quickly generate first visuals
- **Test functionality:** Based on early prototype, iterate the overall look / basic functionality
- **Prototype first:** Creating a functional prototype that covers the basic concept

AI generates prototype typically after several iterations in a prompt window

The SAIL Framework II

Phase 2: Feature Expansion

- **Extended logic:** Scientific principles can be expanded (e.g., new physics models)
- **Improved accuracy:** Initial simplifications can be replaced with advanced approximations after verification. *Tip: Fix random seeds to isolate coding errors from physics validation*
- **New features:** Functionality beyond the basic use (e.g., grid visualization, image upload)
- **Analysis tools:** Inclusion directly within the prototype
- **Performance optimization:** AI can suggest shortcuts / coding approaches. Requires constant oversight of retaining the science. *Tip: beware of silent logic overwrites!*

Development feature-by-feature in isolation and testing each addition before continuing

The SAIL Framework III

Phase 3: Refinement

- **Usability & stress testing:** Testing for edge-cases and unexpected inputs (e.g., „overdense“ void)
- **Additional features:** Refined interface, instructions, data export, presets
- **Code standardization:** refactoring, modularization and documentation

Ultimately refining and reviewing code to ensure accuracy and usability of the final tool

Forgot some feature or want to expand more?

-> Go back to Phase 2! SAIL is also iterative between phases

AI Coding Pitfalls

- **Misalignment:** LLM loses track of main objective, needs constant intervention to reset context
- **Knowledge Deficit:** AI tries to override science if it's beyond the standard textbook approach (e.g., voids) or code excels in one part while failing in other areas (e.g., correct lensing but lacking realistic visuals)
- **Contextual limitations:** Theory is applied correctly, but limitations/breakdowns are not recognized

-> Typically in earlier development

- **Logic overwrite:** Code refactored while quietly deleting previously validated code
- **Architectural instability:** Code too long for context window results in breakdown when new features are added

-> Often due to token limits and can be solved by better coding practices

SAILing from Prototype to Public I

Single-File Prototyping (Phases 1-2):

- **Low Barrier:** Interaction with single model in the context window
- **Rapid Testing:** Fast and iterative validation of implemented concepts
- **Self-Contained:** Code within single file. Depending on LLM, can be run directly in context window
- **Risks:** Limitations of context window might necessitate advanced approaches

Careful: Don't forget version control or save files & prompts locally after significant changes

SAIling from Prototype to Public II

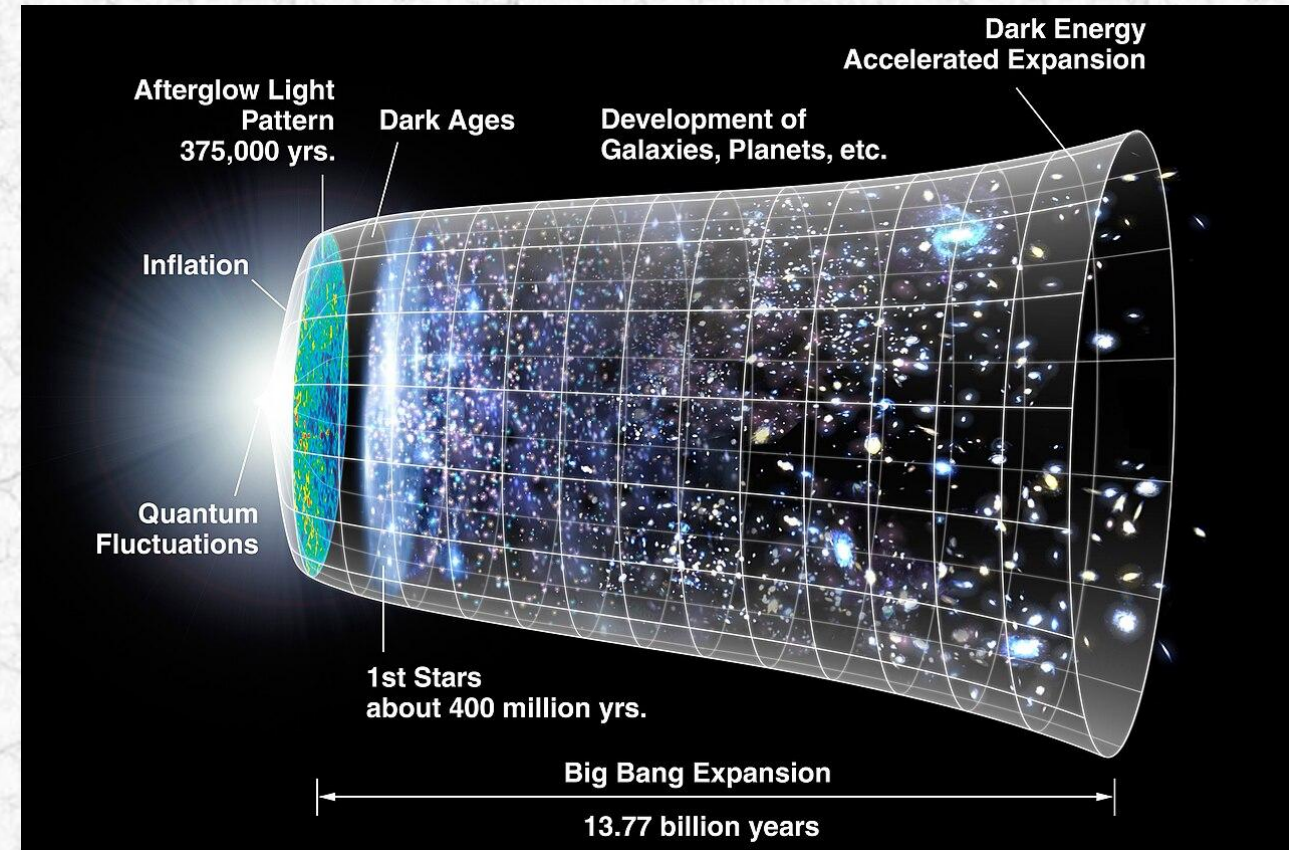
Agentic IDE Coding (Phases 2-3):

- **Higher entry barrier:** Work in Agentic IDE (VSCode, Cursor, etc.)
- **Multi-File Architecture:** AI can create multi-file repository to separate different logic and features (e.g., physics engine, visual style)
- **Refactoring & Standardization:** Clean up code, create documentation
- **Breaking Context Limits:** Code kept in context, AI only modifies certain sections
- **Feature-by-Feature:** Developed in isolation and iteratively

Best approach: Rapidly validate concepts in **Single-File**, then graduate to **IDE** for expansion, optimization and finalization

Exploring the Cosmic Web

- **The goal:** A browser-based „quasi-N-body“ simulation for cosmology outreach/teaching
- Demonstrates the formation of structures in our Universe
- Highlights baryonic acoustic oscillations and cosmic voids
- **Challenges:** Simulations typically run in millions of CPU-hours. LLM don't recognize theory breakdowns



Source: [Wikipedia](#)

Exploring the Cosmic Web

Live Demo:



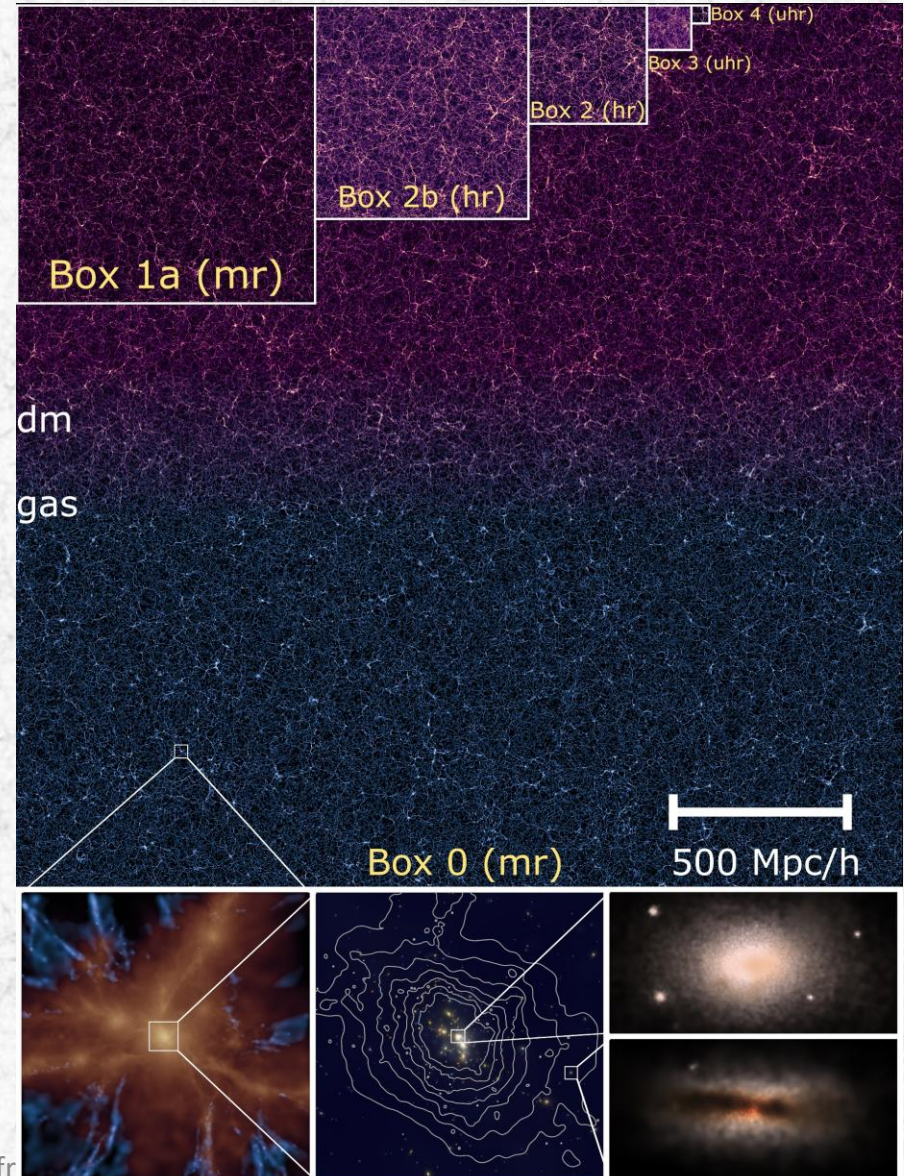
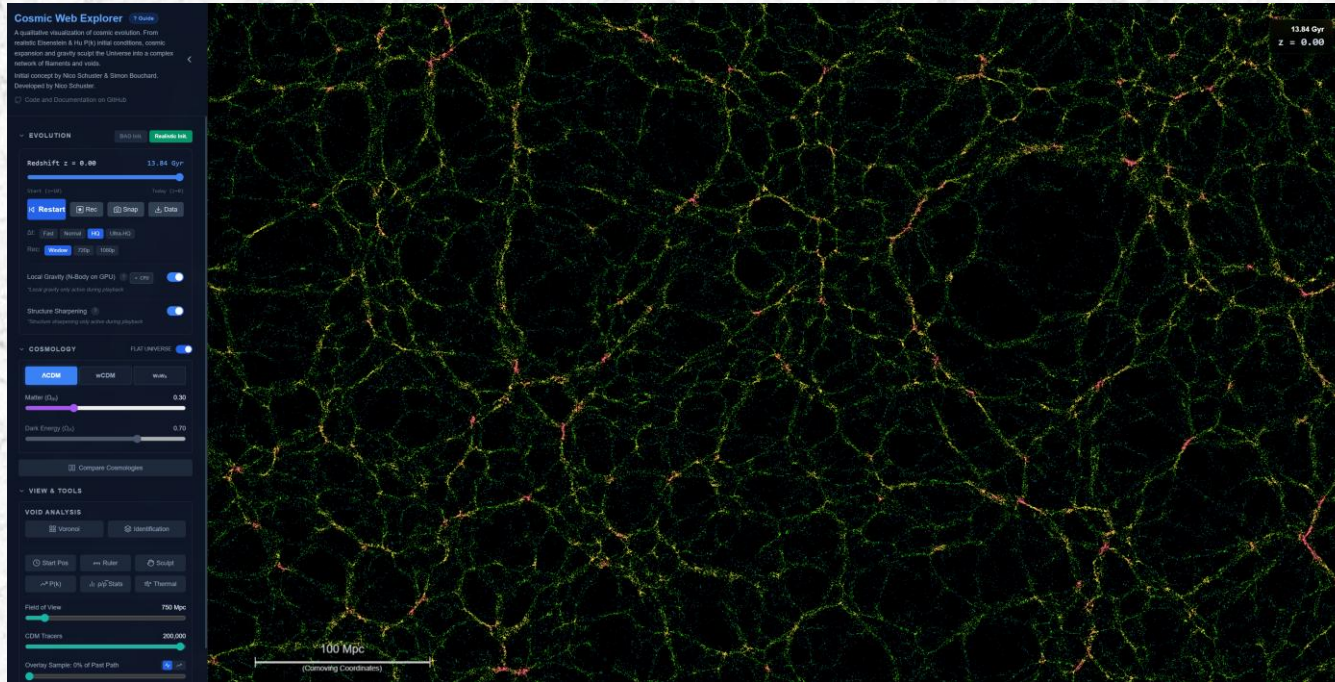
tinyurl.com/cosmic-web-explorer

or

https://nicosmo.github.io/cosmic_web_explorer

Exploring the Cosmic Web: Simulations

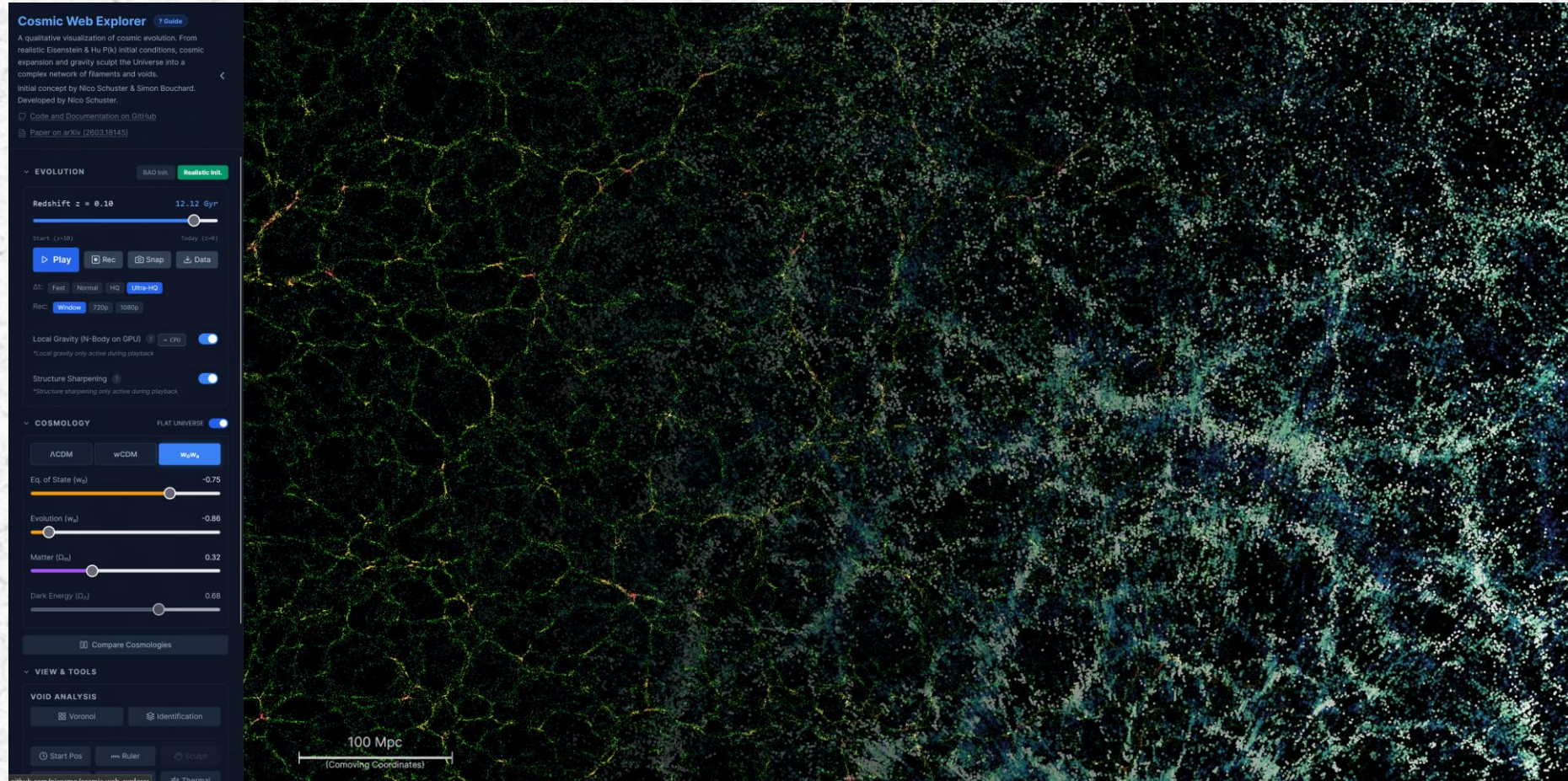
Comparison with cosmological simulations:



Credit:
Benjamin
Seidel (see:
[Dolag et al. 2025](#))

Exploring the Cosmic Web: Data

Comparison with the year 5 DESI data:



Produced for DESI/CNRS press release: tinyurl.com/DESI-CNRS-Y5-DATA

Accelerated Development & Scalability

- **Condensed timelines:** Time shifted from coding to conceptual oversight & use of intuition
Gravitational Lensing tool (30-40 hours) & Cosmic Web Explorer (<80 hours incl. testing)
- **Reduces technical barrier:** We can actually spend time to develop such tools
- **Universal blueprint:** Applicable to many scientific fields and concepts, *e.g. visualizations of collider events, interactive Feynman diagrams, detector designs, gravitational wave events, supernova light curves, and more concepts beyond physics*
- **Broader application:** SAIL principles apply beyond visualization to any general scientific computing and data analysis, although **more rigorous verification required**

Impact: Outreach & Teaching

- **„Gamified“ Learning:** Interactions and high visual fidelity improve engagement
- **Outreach:** Abstract concepts can be translated into observable phenomena for the public
- **University Teaching:** Students can explore scientific mechanics, e.g. observe how changing parameters affect models to build intuition
- **Adaptable level:** Complexity can be matched to the expertise of the audience, from introductory courses to advanced master courses

Impact: Presentations & Prototyping

- **Niche Visualization:** Highly targeted tools to explain specific concepts, e.g., new theoretical developments to peers
- **Theoretical Prototyping:** Visualize and test new frameworks or phenomenological models before committing to more computationally expensive analyses
- **Collaborations:** Theorists and experimentalists can enhance dialogue about models and measurements.
- **Instant Access:** Lightweight tools for browsers allow complex models to be shared easily

SAIL Conclusions

- **Syntax vs. Science:** LLMs optimize for code, not true science
- **Scientist-AI-Loop:** Scientist as the Architect (Logic) and AI as the Syntax Engine (Implementation)
- **SAIL framework:** Baseline Prototype -> Feature Expansion -> Refinement
- **Expert Oversight:** Constant verification of the Science and Logic, Identification of Failures
- **Condensed Timeline:** Coding speed up significantly from sometimes months to days
- **Universal Outreach & Sandboxing:** Quick and more effective communication of science

The Paper:



[arxiv:2603.18145](https://arxiv.org/abs/2603.18145)

Lensing Tool:



Build tools, experiment with them and share them!

Cosmic Web:



A last tip: Github Copilot Pro by applying for GitHub Education (**free!**)

13.84 Gyr
 $z = 0.00$

Thank you

150 Mpc
(Comoving Coordinates)

nicosmo.github.io/cosmic_web_explorer

schuster@cppm.in2p3.fr