Using Natural Language to Do Science

How Agentic AI Is Changing Physics

James Nightingale

www.jamesnightingale.net

By the end of this workshop you will

- **Consider** how natural language is becoming the primary interface for doing data science.
- **Know** what agentic AI is.
- **See** how AI can act as a scientific research assistant.
- **Discover** how agentic AI is transforming scientific software development.
- **Explore** vibe coding as a creative endeavour that can enhance scientific insight.



Margaret Hamilton with the Apollo guidance software, 1969.

A health warning

For most of this talk I will **drink the AI kool-aid** and show you what these tools can do.

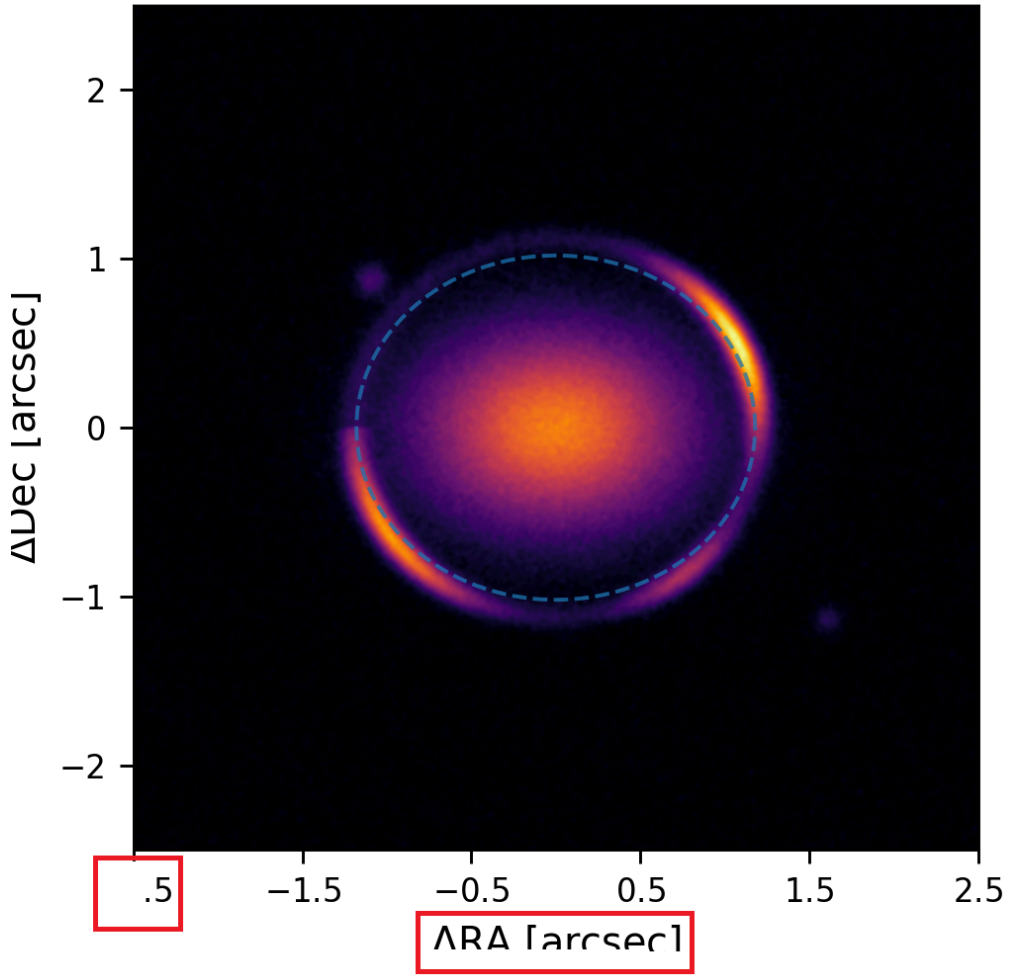
At the end, we will come back down to earth and discuss its **pitfalls and challenges**.



This image may be AI generated.

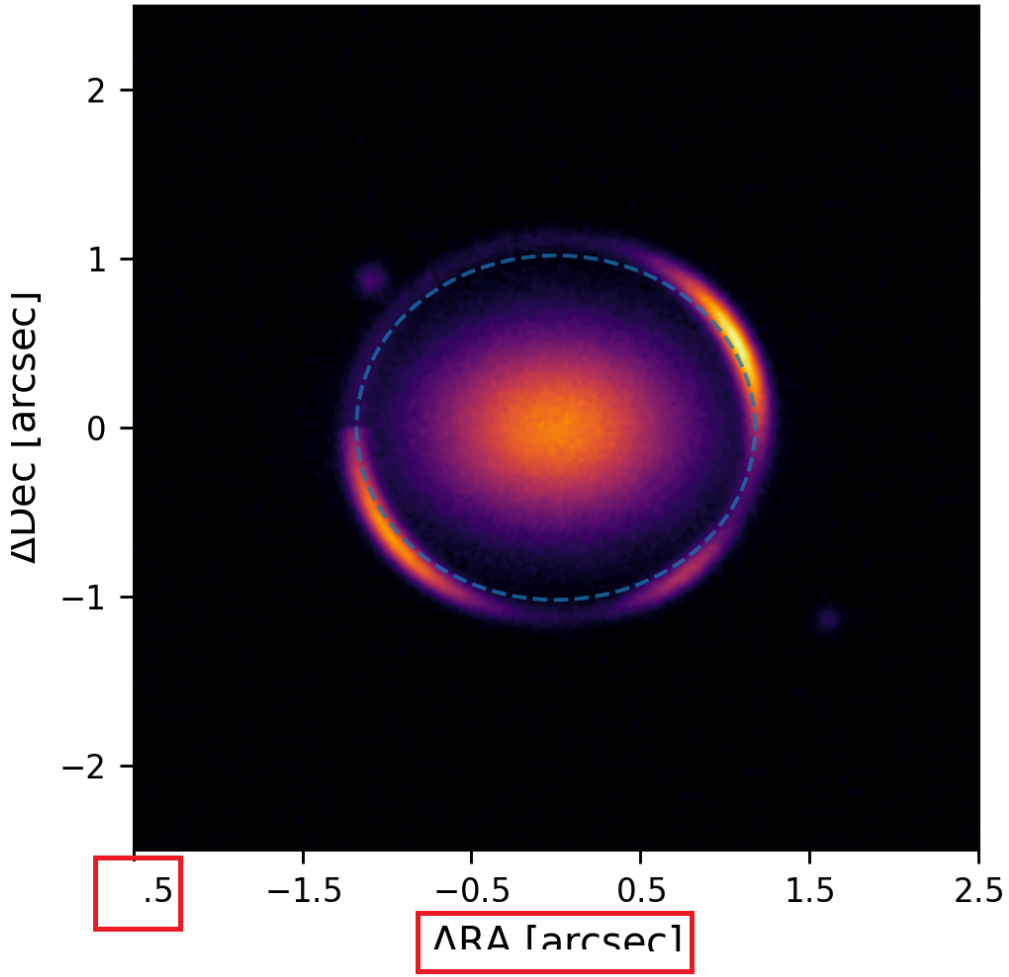
Remember This?

- You spot a tiny issue with a matplotlib figure. 🔍



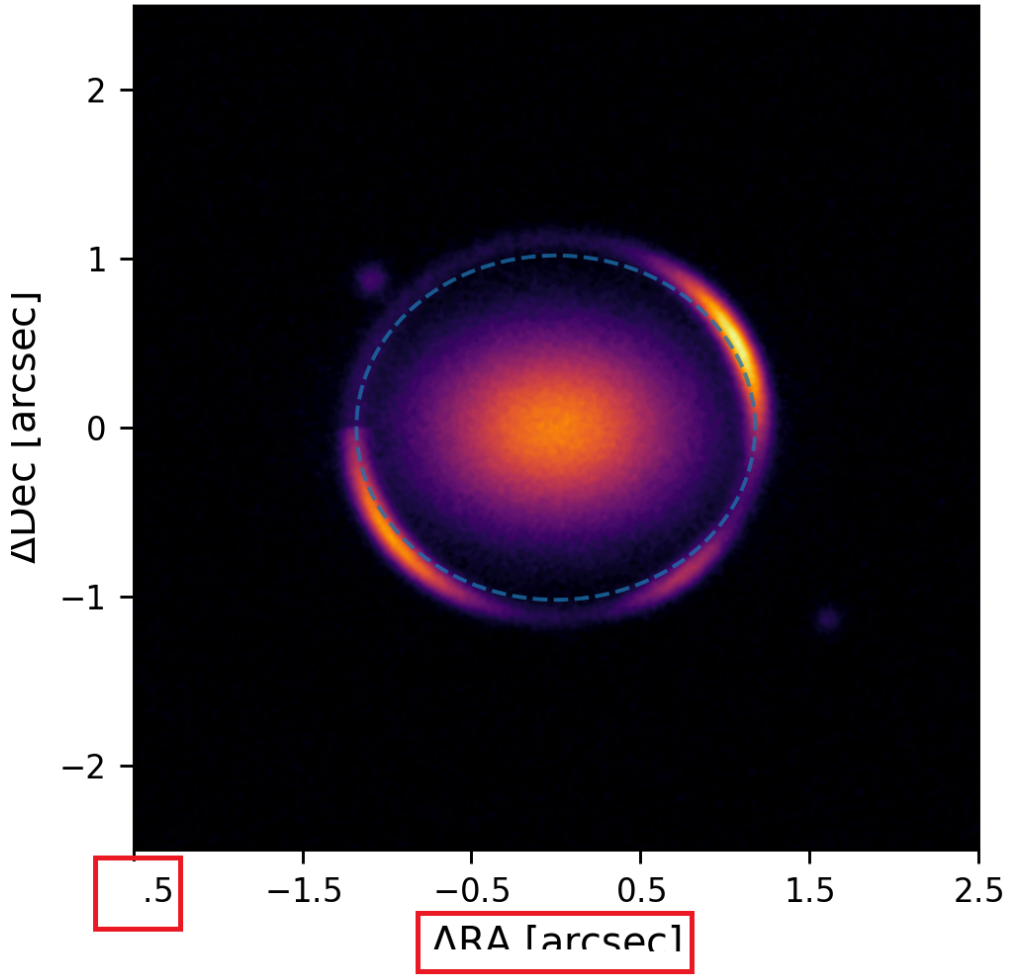
Remember This?

- You spot a tiny issue with a matplotlib figure. 🔍
- You change the python code, but it doesn't change the figure. 😤



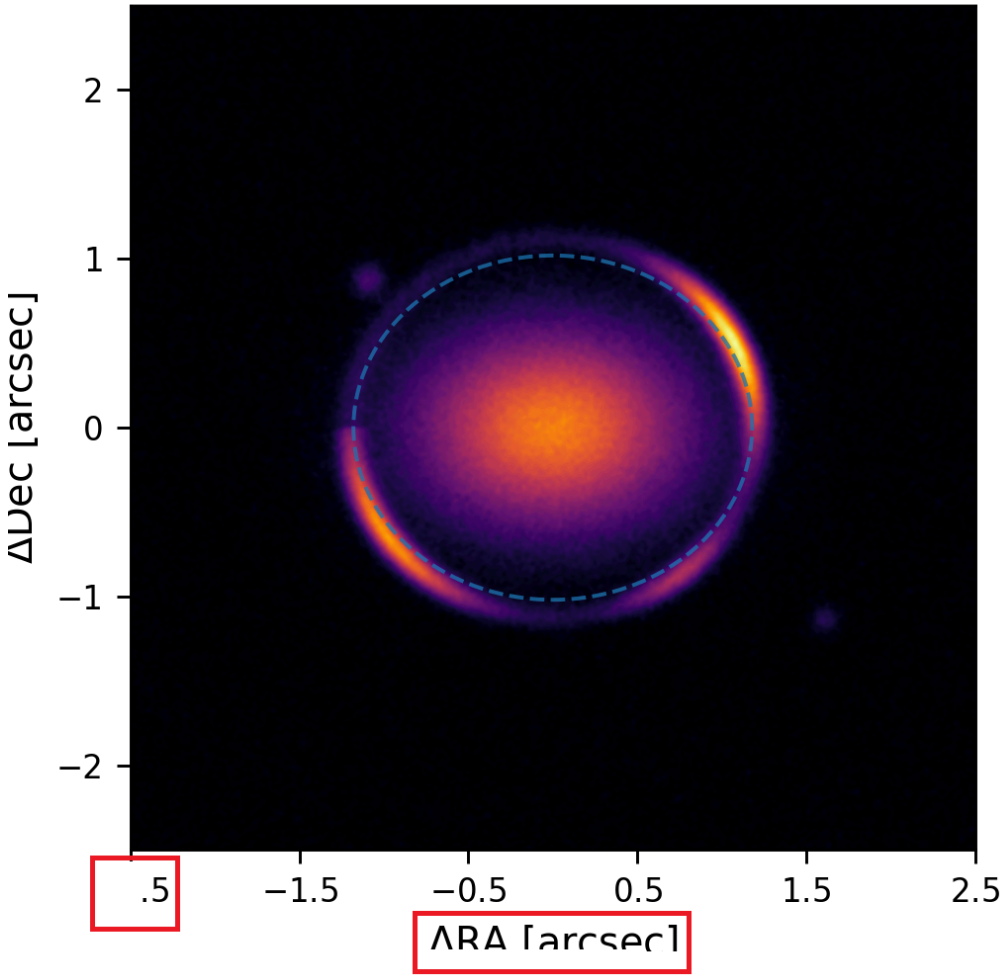
Remember This?

- You spot a tiny issue with a matplotlib figure. 🔍
- You change the python code, but it doesn't change the figure. 😤
- You trial-and-error StackOverflow answers for the **next four hours**. 😩

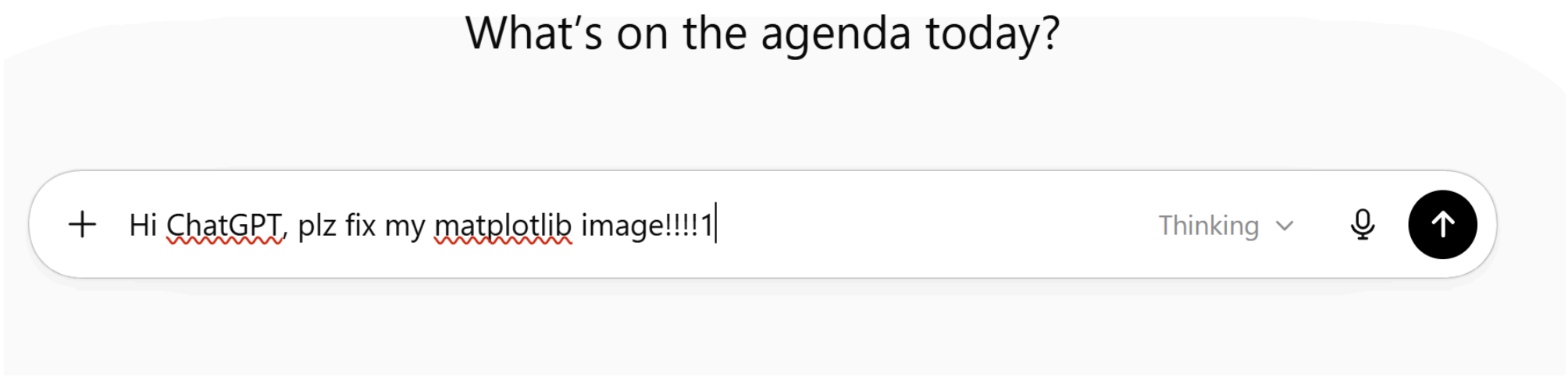


Remember This?

You wish you could just **tell matplotlib what you want the figure to look like**.



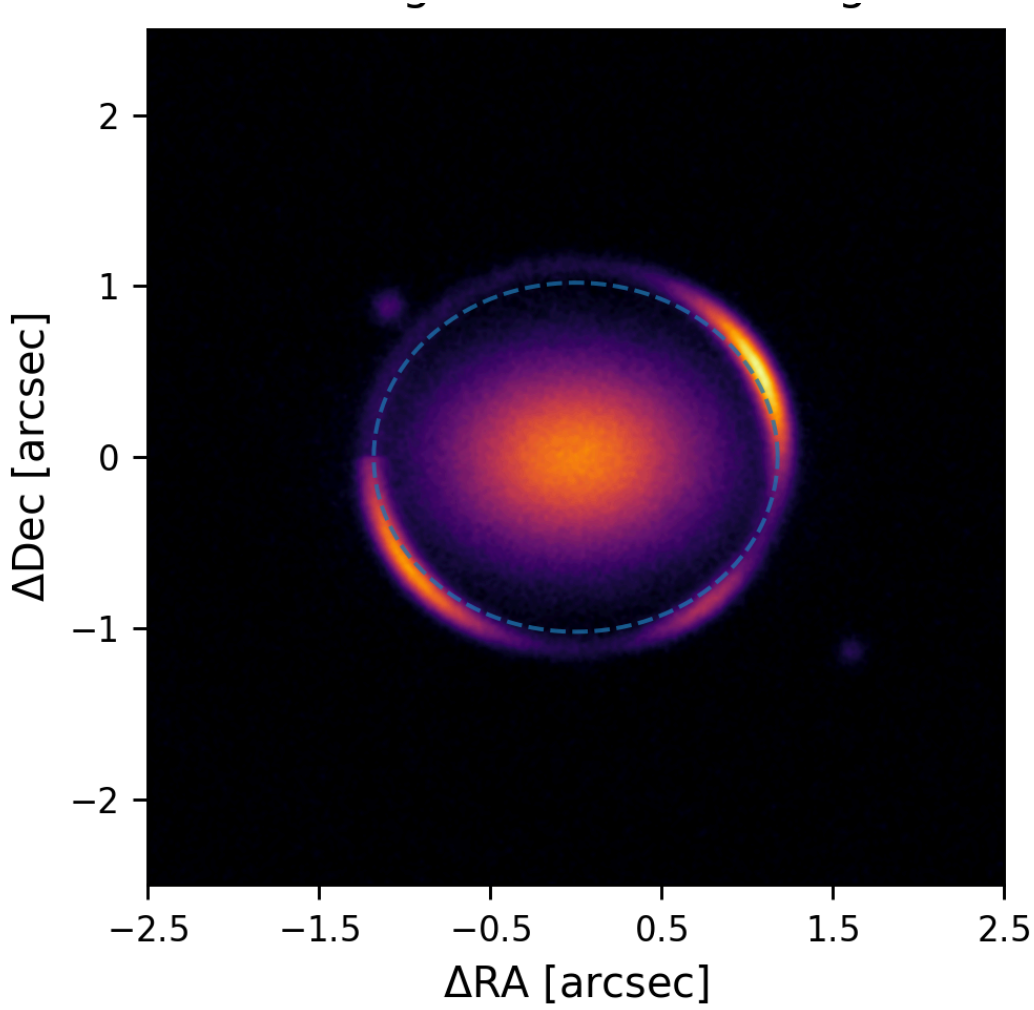
Now You Can!


What's on the agenda today?
Hi ChatGPT, plz fix my matplotlib image!!!!1
Thinking

Using Natural Language

We don't have this problem anymore.

AI lets us use **natural language** to interface with matplotlib.



Who has used ChatGPT or another AI to help make a science figure?

[Poll the room: hands up.]

Who enjoyed writing matplotlib (or other visualisation library) code and *wrestling with its API and syntax?*

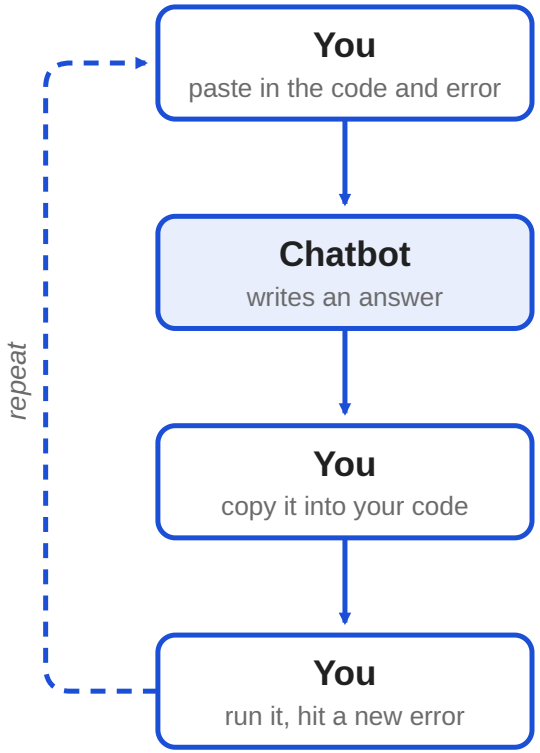
[Poll the room: hands up.]

What Is Agentic AI?

A chatbot answers

ChatGPT, Gemini and similar tools, in a chat window.

- Only knows what you paste into the chat window.
- **You** run the code, copy back the error, and repeat.
- Every step goes through **you**.

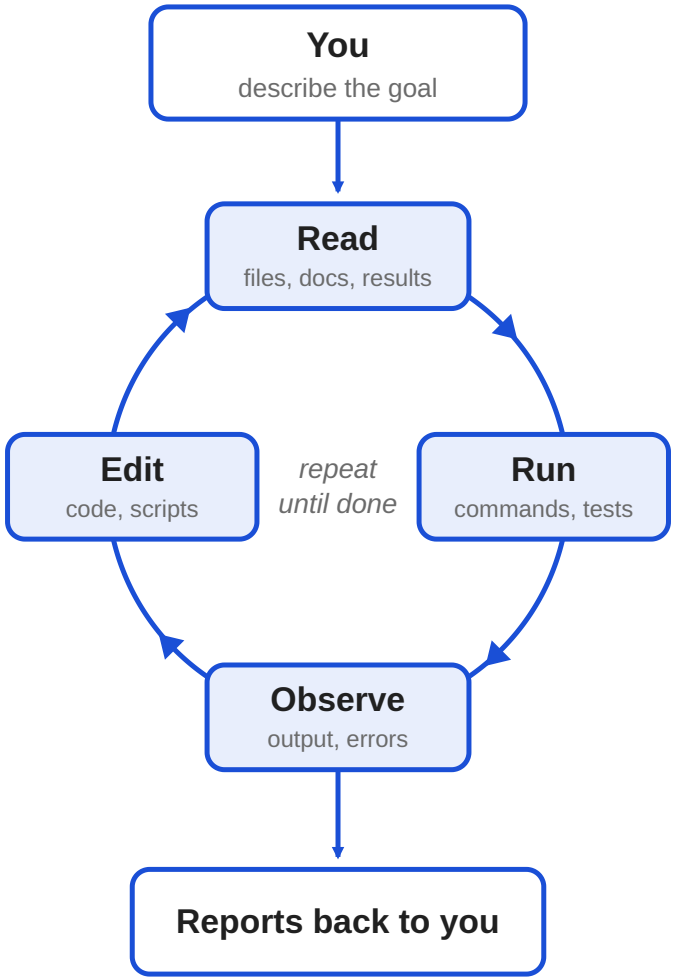


It only knows what you paste.
Every step goes through you.

An agent acts

Claude Code (Anthropic) and **Codex** (OpenAI), in a terminal.

- Runs inside your project folder.
- Reads every file, runs any command, edits code and checks its own work **itself**.
- Asks before doing anything that cannot be undone.



Live Demo

Example 1: One prompt to Claude

> **Read** the source code folders @PyAutoGalaxy and @PyAutoLens and **find** the PyAutoLens power-law mass model parameterization.
>
> **Read** the paper on the COOLEST strong lensing standards (https://arxiv.org/abs/2406.08484).
>
> **Compare** its parameterization to the COOLEST lens model standard by **reading** the COOLEST source code on GitHub (https://github.com/aymgal/COOLEST).
>
> **Go to** the PyAutoLens GitHub repository and **open an issue** comparing the two parameterizations and how they differ.
>
> **Flag** that this issue is for the Natural Language workshop live demo, **give us a winking emoji** and **close the issue**. 😉

Result: github.com/PyAutoLabs/PyAutoLens/issues/820

The part that matters

Parameter-by-parameter comparison

Quantity	PyAutoLens	COOLEST `PEMD`	Direct map?
Centre	`centre = (y, x)`, arcsec, `(y, x)` ordered	`center_x`, `center_y`, arcsec	Yes — same units/origin, only `(y,x)` vs `(x,y)` ordering
Density slope	`slope` = gamma (3D log slope, kappa ∝ R^(1−gamma))	`gamma`, "Mass density slope at Einstein radius"	**Yes** — same definition, gamma = 2 is isothermal, range ~1–3
Ellipticity	`ell_comps = (e1, e2)` Cartesian components	`q` (= b/a) and `phi` directly	No — see below
Orientation	encoded in `ell_comps`; recovered as angle CCW from **+x axis**, range [0, 180)	`phi` CCW from **+y axis** (East-of-North), range (−90, +90]	No — 90° axis offset + different range
Einstein radius	`einstein_radius`, normalized on the **major axis**, amplitude factor `1/(1+q)`	`theta_E`, normalized on the **intermediate axis** `r_inter = sqrt(a*b)`, amplitude factor `1/2`	No — q-dependent rescaling required

Example 2: Building a LaTeX Table

Manually Writing a LaTeX Table

- You copy a LaTeX table template into Overleaf and adapt it to your results. 📋

```
\begin{table}
\caption{...}
\begin{tabular}{cccccc}
\hline
Run & $L$ (m) & $m$ (kg) & $T$ (s) & $\tau$ (s) & $g$ (m\,s$^{-2}$) \\
\hline
 &  &  & $^{+}_{-}$ & $^{+}_{-}$ & $^{+}_{-}$ \\
\hline
\end{tabular}
\end{table}
```

Manually Writing a LaTeX Table

- You open a file with your results. 📂

```
Summary (3.0 sigma limits):

oscillator
    period                                                  1.0038 (1.0000, 1.0072)
    damping_time                                            51.0 (47.7, 54.0)
    phase                                                   1.6397 (1.6157, 1.6647)
```

Manually Writing a LaTeX Table

- You find the period, damping time and g in the 3σ summary. 🔍

```
Summary (3.0 sigma limits):

oscillator
    period                                                  1.0038 (1.0000, 1.0072)
    damping_time                                            51.0 (47.7, 54.0)
    phase                                                   1.6397 (1.6157, 1.6647)
```

Manually Writing a LaTeX Table

- You copy and paste them into LaTeX. ⌨️

```
1 & 0.25 & 0.05
  & $1.0038^{+0.0034}_{-0.0038}$
  & $51.0^{+3.0}_{-3.3}$
  & $9.796^{+0.076}_{-0.070}$ \\
```

Manually Writing a LaTeX Table

- You repeat for **every column of every run**. 😩

```
1 & 0.25 & 0.05 & $1.0038^{+0.0034}_{-0.0038}$
  & $51.0^{+3.0}_{-3.3}$ & $9.796^{+0.076}_{-0.070}$ \\
2 & 0.50 & 0.10 & $1.4159^{+0.0042}_{-0.0042}$
  & $106.2^{+13.4}_{-12.1}$ & $9.846^{+0.058}_{-0.064}$ \\
3 & 0.75 & 0.05 & $1.7404^{+0.0071}_{-0.0064}$
  & $54.1^{+4.4}_{-4.8}$ & $9.775^{+0.082}_{-0.072}$ \\
4 & 1.00 & 0.20 & $2.0092^{+0.0081}_{-0.0084}$
  & $132.2^{+8.8}_{-9.9}$ & $9.779^{+0.085}_{-0.083}$ \\
5 & 1.25 & 0.10 & $2.2431^{+0.0055}_{-0.0054}$
  & $113.1^{+10.5}_{-11.9}$ & $9.808^{+0.046}_{-0.052}$ \\
```

Manually Writing a LaTeX Table

You wish you could just **tell LaTeX what you want the table to look like**.

One prompt to Codex

> The **output** folder has the fit results for five runs of a pendulum experiment, and **paper/main.tex** is our paper.
>
> **Read** the model.results file in each run's folder and **build** a table in the Results section from them: string length, bob mass, period, damping time and g for every run, with 3 sigma errors.
>
> **Write** a caption describing the table's contents.
>
> **Build** the PDF and **show** me the table.

No More LaTeX!

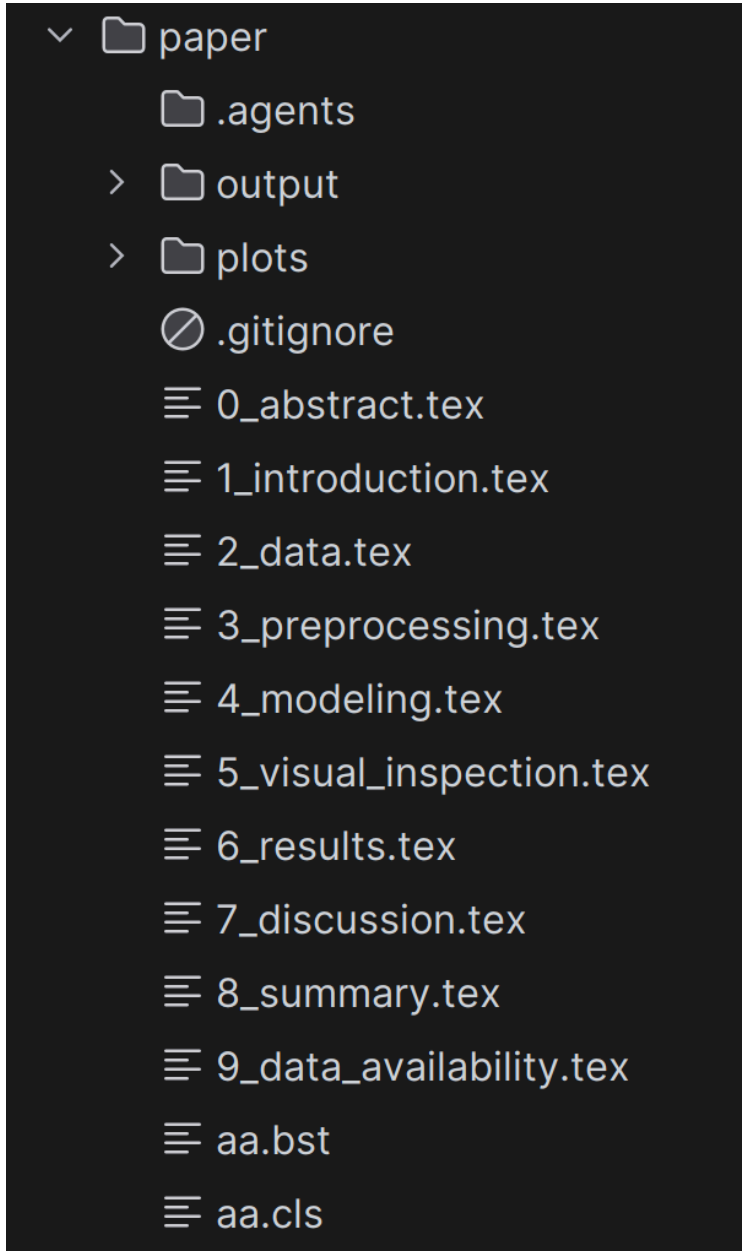
TABLE I. Setup and fitted parameters for the five pendulum runs. For each run we list the string length L and bob mass m, the period T and damping time τ from the damped harmonic oscillator fit, and the gravitational acceleration $g = 4\pi^2 L/T^2$ derived from them. Errors are 3σ limits on the posterior.

	Setup		Fit		Derived
Run	L (m)	m (kg)	T (s)	τ (s)	g ($\mathrm{m\,s^{-2}}$)
1	0.25	0.05	$1.0038^{+0.0034}_{-0.0038}$	$51.0^{+3.0}_{-3.3}$	$9.796^{+0.076}_{-0.070}$
2	0.50	0.10	$1.4159^{+0.0042}_{-0.0042}$	$106.2^{+13.4}_{-12.1}$	$9.846^{+0.058}_{-0.064}$
3	0.75	0.05	$1.7404^{+0.0071}_{-0.0064}$	$54.1^{+4.4}_{-4.8}$	$9.775^{+0.082}_{-0.072}$
4	1.00	0.20	$2.0092^{+0.0081}_{-0.0084}$	$132.2^{+8.8}_{-9.9}$	$9.779^{+0.085}_{-0.083}$
5	1.25	0.10	$2.2431^{+0.0055}_{-0.0054}$	$113.1^{+10.5}_{-11.9}$	$9.808^{+0.046}_{-0.052}$

Now I tell **Codex** which results go in the table and what it should look like. It pretty much does it.

Using Natural Language to Format a Paper

- One `.tex` file per section, the plots, the outputs, and a `.agents` folder that tells the agent how the paper is written.
- The `plots` folder contains the Python scripts that make the plots, and the figures themselves.
- The agent handles paper formatting, referencing and building the PDF. I read, argue and decide.



Three prompts to your paper

ACRONYMS

"Convolutional neural network" appears all through the paper. **Introduce** the acronym CNN at its first use, then **replace** every later occurrence with CNN across all sections, captions and tables.

RESTRUCTURE

Move Methods before Data and **merge** Sections 4 and 5 into one Results section. **Renumber** the sections, **update** every cross-reference, and **rewrite** each moved section's opening so the paper still reads in order.

FIGURES

Read the journal's figure style guide. **Regenerate** every plot from its script to meet it: column width, font sizes, line weights and colour-blind-safe colours. **Update** every figure in the paper and **check** that each caption still matches its plot.

Who has used agentic AI to help format a LaTeX paper?

[Poll the room: hands up.]

Who enjoyed writing LaTeX code and *wrestling with its tables, formatting and syntax?*

[Poll the room: hands up.]

Using Natural Language to...

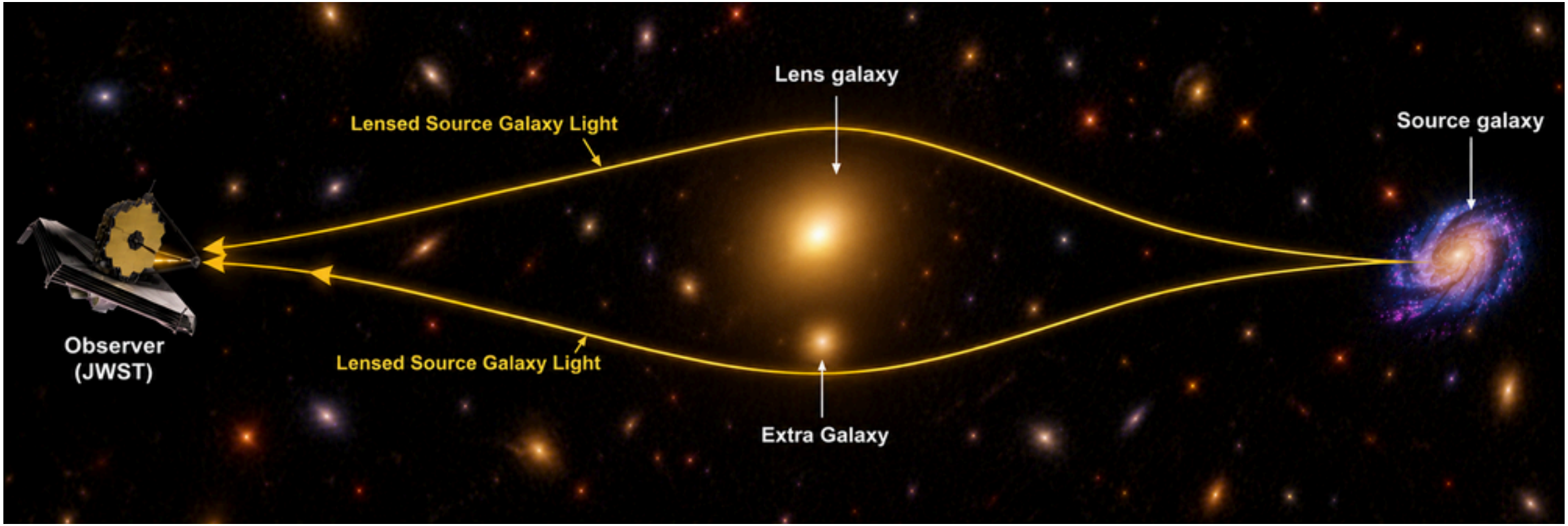
- **Install software:** no fighting conda environments, pip conflicts or compiler errors.
- **Manage jobs on a supercomputer:** no Slurm syntax, batch scripts or `ssh` / `rsync` juggling.
- **Manage code with Git:** no memorising commands to branch, merge or untangle a conflict.



Using Natural Language to Do Science

Strong Gravitational Lensing

I study galaxies whose gravity bends the light of a more distant galaxy into arcs and rings, and use that distortion to weigh them, **dark matter included**.



The COSMOS-Web Ring

Scientists can often look at a dataset and know exactly the analysis they want to perform.

For example, when shown the multi-wavelength **James Webb Space Telescope (JWST)** observations of the COSMOS-Web Ring, I would think:



The COSMOS-Web Ring

“I want to model the F150W and F444W JWST imaging of the COSMOS-Web Ring independently, which are in my local folder dataset/cosmos_web_ring.

Model the lens light with a multi-Gaussian expansion, its mass with a singular isothermal ellipsoid plus external shear, and model the source using a Delaunay mesh.

For speed, run the analysis on the laptop GPU using a JAX optimizer that estimates only the maximum-likelihood solution.

Plot the observed image at each wavelength in the left column, its lensed source model in the second column, its residuals in the third column, and its source reconstruction in the right column.”



Doing It By Hand

- **Download** the JWST images and store them in a folder on my laptop. 📥
- **Find** the PyAutoLens example scripts and adapt them to this lens. 📝
- **Upload** everything to the HPC and set off the job. 🖥️

Even as the developer of PyAutoLens, this would take me **at least 30 minutes**.



PyAutoLens AI Assistant

Starting an AI Assistant with Knowledge using a Wiki

Everything the agent learns, from papers, results and conversations, is written into a **wiki in natural language**.

Each new task **builds on that knowledge**, so the agent knows more about your science every time you use it.

RAG is Dead. Karpathy's LLM Wiki is the future | Project Explained



Simranjeet Singh Follow 9 min read · May 2, 2026

309 11 1

> *A deep-dive into Andrej Karpathy's viral LLM Wiki concept, and how to apply it to build a real, production WhatsApp AI Bot powered by Gemini 3.0 Flash.*

You upload the PDF. Again.

Same document. Same question. ChatGPT reads it, thinks for a second, and gives you the same answer it gave you yesterday, like it's meeting this file for the very first time.

Because it is.

That's the quiet frustration nobody talks about. Every AI session starts with amnesia. You feed it context, it helps you, the conversation ends — and everything evaporates. Next time, you start from zero. Again.

Now multiply that by your research papers, your meeting notes, your bookmarks, and your half-finished Notion pages scattered across three workspaces. You have *knowledge*. Your AI just can't hold onto it.

Then, in April 2026, **Andrej Karpathy, co-founder of OpenAI,** the man who taught the internet how transformers work, quietly dropped a GitHub Gist.

The PyAutoLens Assistant Wikis

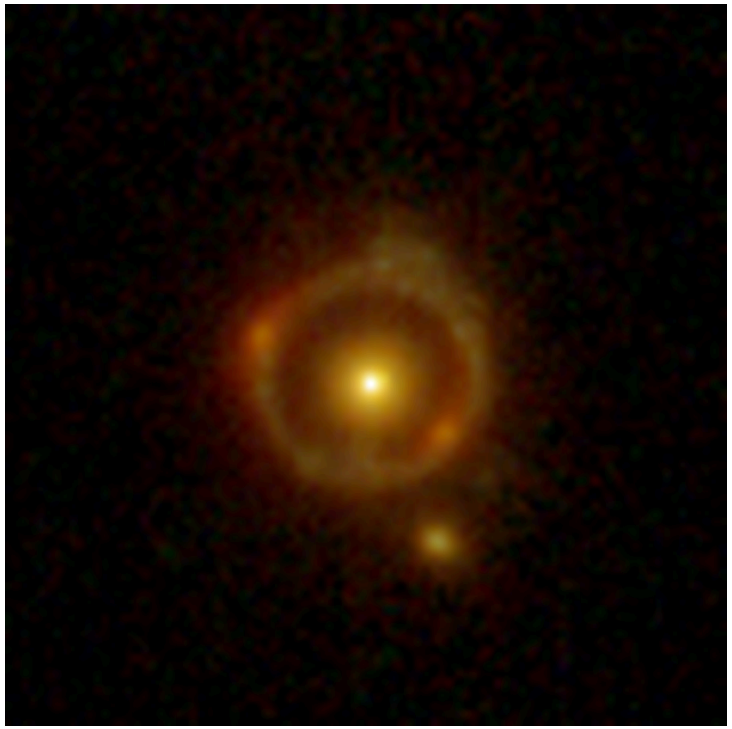
The assistant starts with three wikis of knowledge:

- **Literature:** my 400 favourite strong lensing papers.
- **Documentation:** the PyAutoLens ReadTheDocs and API.
- **Tutorials:** end-to-end science examples.

[Live Demo]

github.com/PyAutoLabs/autolens_workspace/blob/main/notebooks/imaging/modeling.ipynb
github.com/PyAutoLabs/autolens_assistant/blob/main/skills/al_build_imaging_model.md

PyAutoLens AI Assistant



Your starting prompt Copy

```
I want to model the F150W and F444W JWST imaging of the COSMOS-Web
Ring independently, which are in my local folder
dataset/cosmos_web_ring.

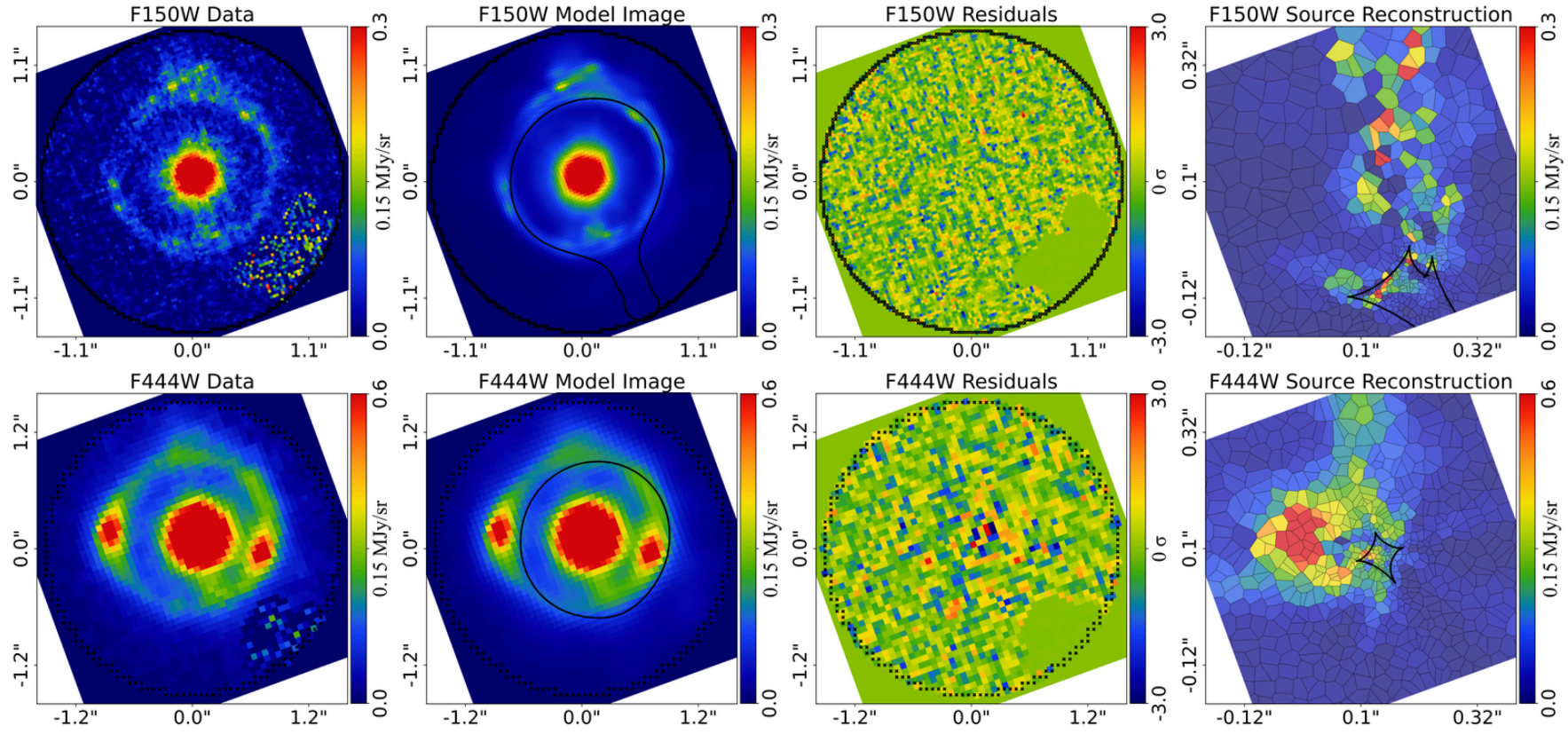
Model the lens light with a multi-Gaussian expansion, its mass with
a singular isothermal ellipsoid plus external shear, and model the
source using a Delaunay mesh.

For speed, run the analysis on the laptop GPU using a JAX optimizer
that estimates only the maximum-likelihood solution.

Plot the observed image at each wavelength in the left column, its
lensed source model in the second column, its residuals in the third
column, and its source reconstruction in the right column.
```

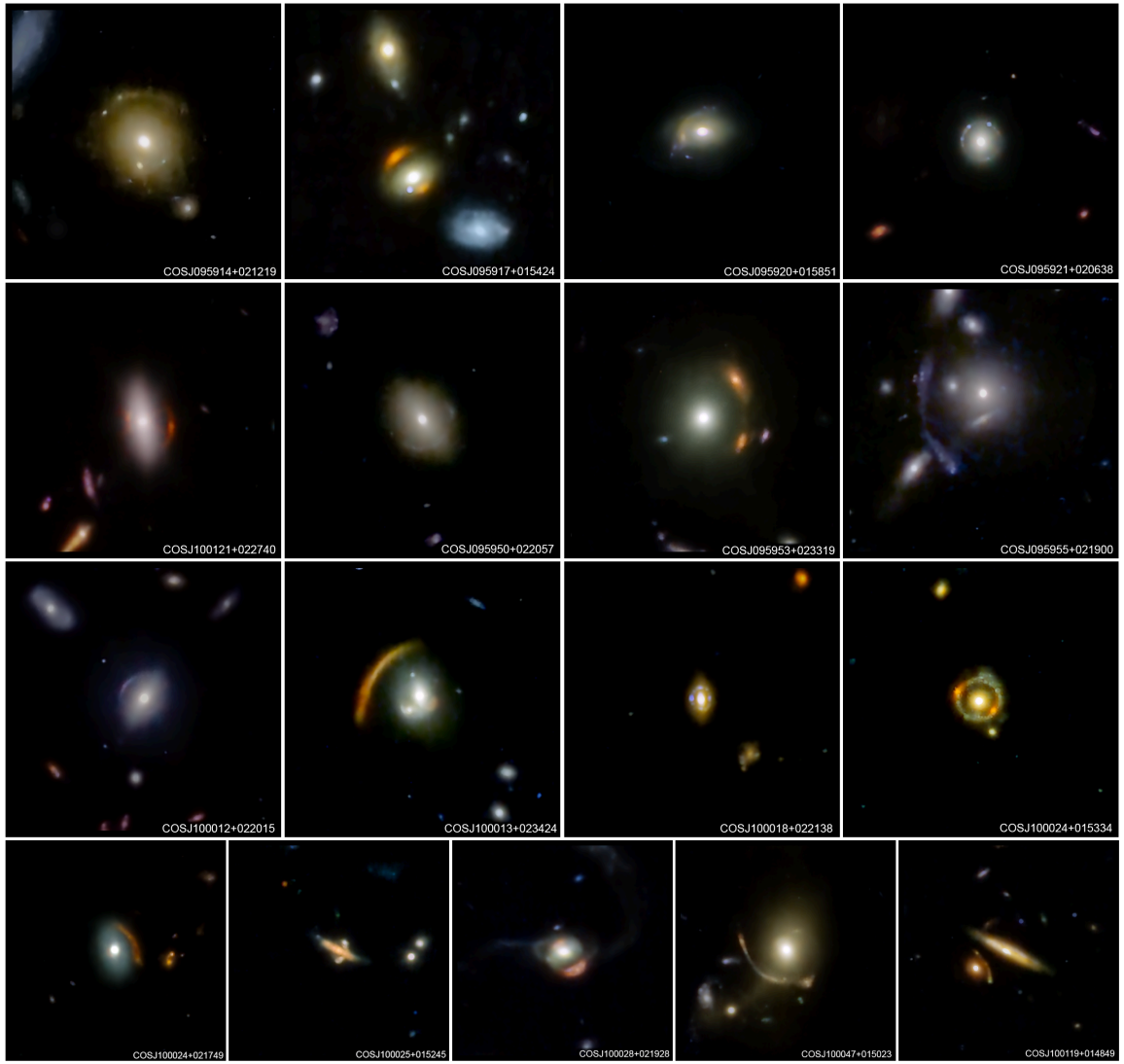
The Result

The assistant fits both wavelengths and plots the data, lensed source model, residuals and source reconstruction: F150W on top, F444W below.



Mercier et al. 2024, A&A 687, A61 (arXiv:2309.15986), Fig. 5, F150W and F444W rows — CC BY-SA 4.0

There are over 100 JWST Strong Lens Images



Credits: Ghassem Gozaliasl & Kaija Virolainen

Your starting prompt Copy

```
There are over 100 JWST strong lenses in my
local folder dataset/cowls, one subfolder per
lens.

Fit every lens with the same model I used for
the COSMOS-Web Ring: F150W and F444W modelled
independently, the lens light with a multi-
Gaussian expansion, its mass with a singular
isothermal ellipsoid plus external shear, and
the source using a Delaunay mesh.

Run them on the HPC, one lens per job, and tell
me when they have all finished.
```

Making Science Accessible (Outreach & EDI)



Your starting prompt Copy

```
I want to use the PyAutoLens Assistant:
https://github.com/PyAutoLabs/autolens_assistant First clone that
repository, cd into it and follow its AGENTS.md.

I'd like to understand how gravitational lensing works using the
JWST image of the COSMOS-Web Ring that ships with the assistant.
Show me the picture, explain what we are looking at, and walk me
through fitting a lens model so we can measure the mass inside the
ring and see how well the model reproduces the observations. Pitch
it at my level: ask me what my background is first. Explain what we
are doing as we go, and let me ask questions or change the analysis
along the way.
```

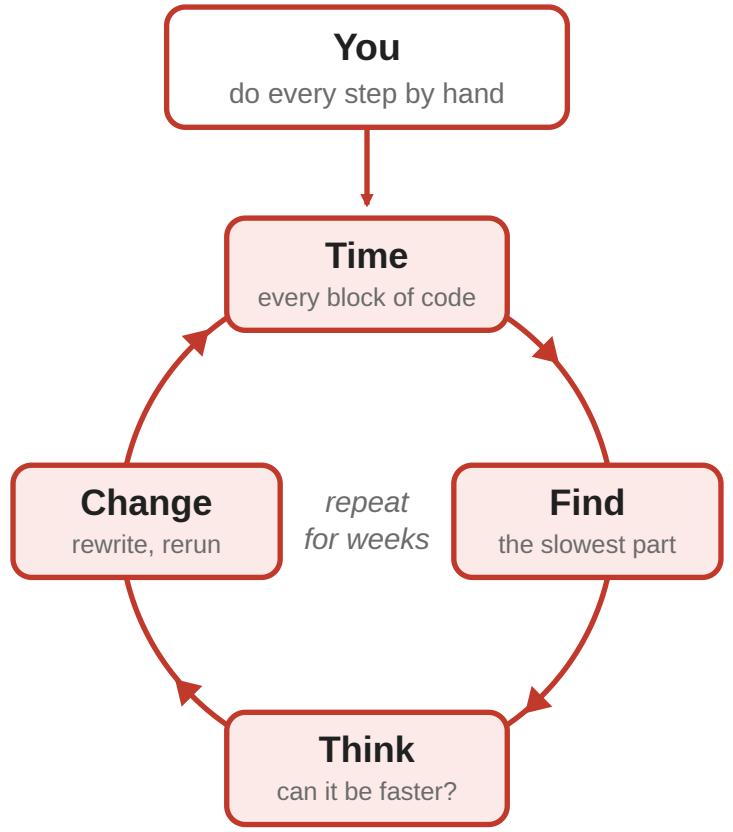
Try it on my website: www.jamesnightingale.net

Loop Engineering & Code Optimization

How I used to make PyAutoLens fast

- ⏱️ **Time** every block of code in the part I wanted to speed up.
- 🐢 **Find** the slowest part.
- 🤔 **Think** whether I could make it faster.
- ✏️ **Change** the code, rerun it, and time it again.
- 🔁 **Repeat**, one idea at a time, for days or weeks. 😩

But honestly? **I really enjoyed this process.** 😄

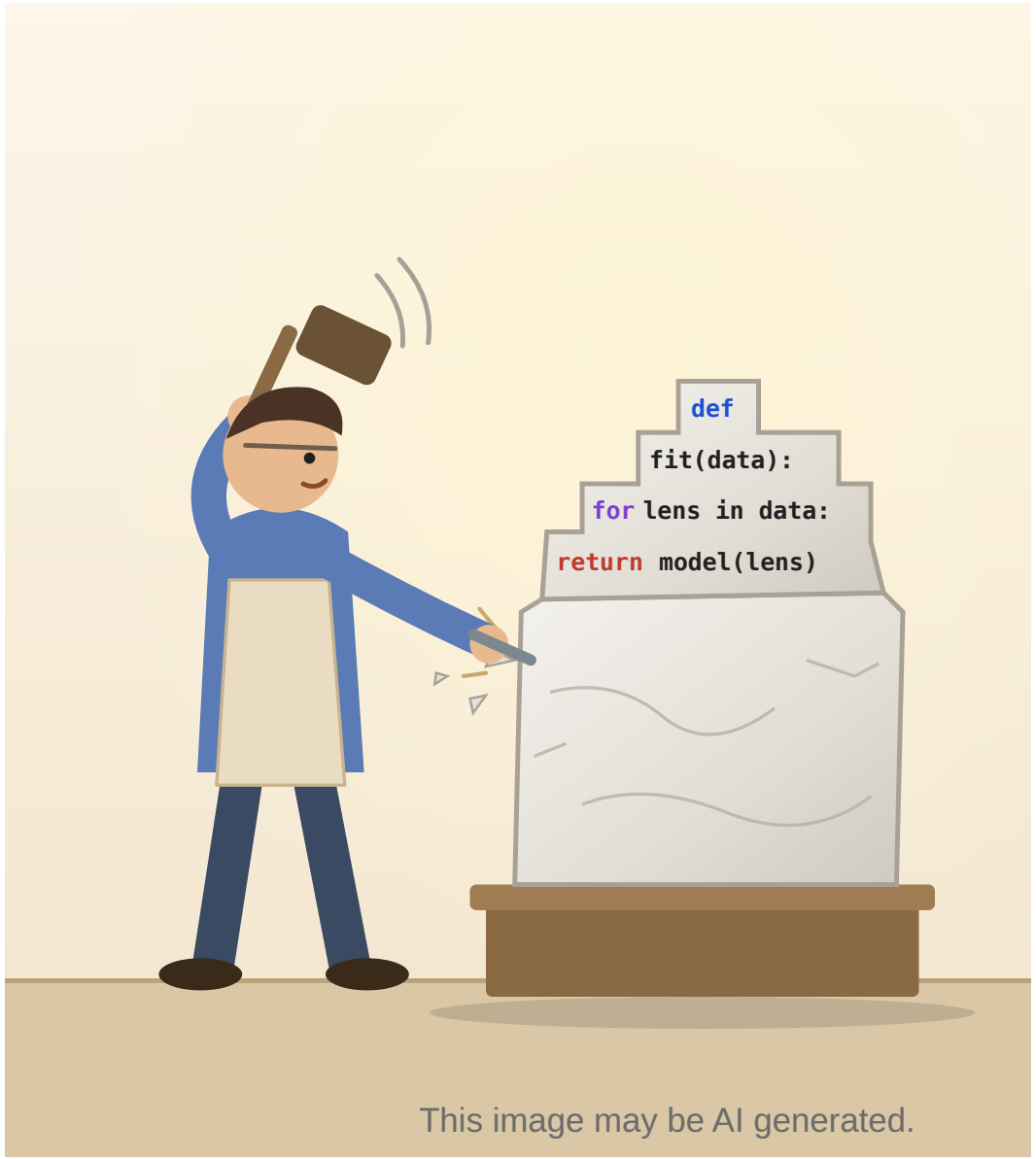


The old way

one hypothesis at a time,
limited by my own time

Code as sculpture 🎨

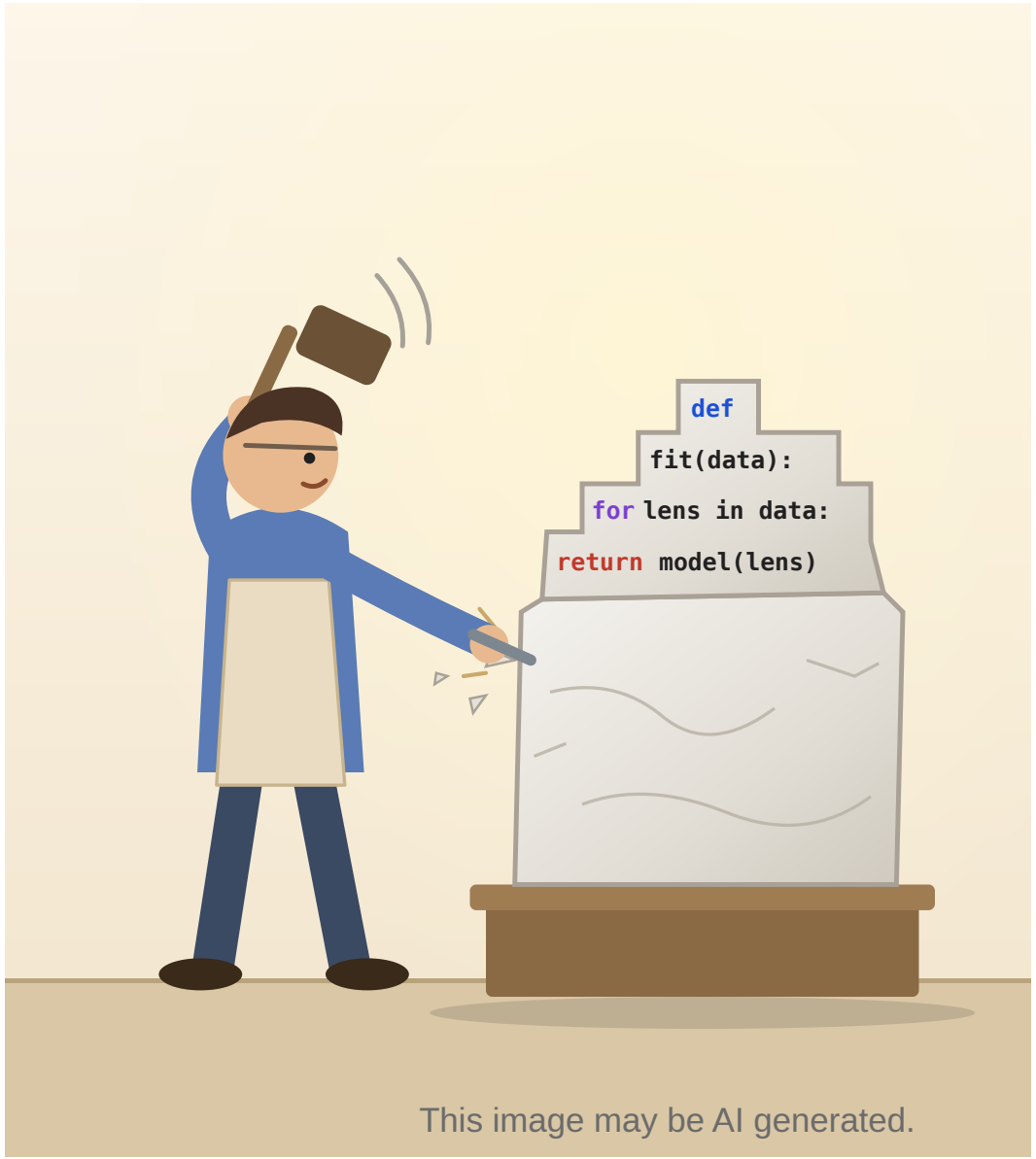
Like a sculptor, I worked the code by hand, one chip at a time, until the fast version emerged from the block. 🔨



Code as sculpture 🎨

Like a sculptor, I worked the code by hand, one chip at a time, until the fast version emerged from the block. 🔨

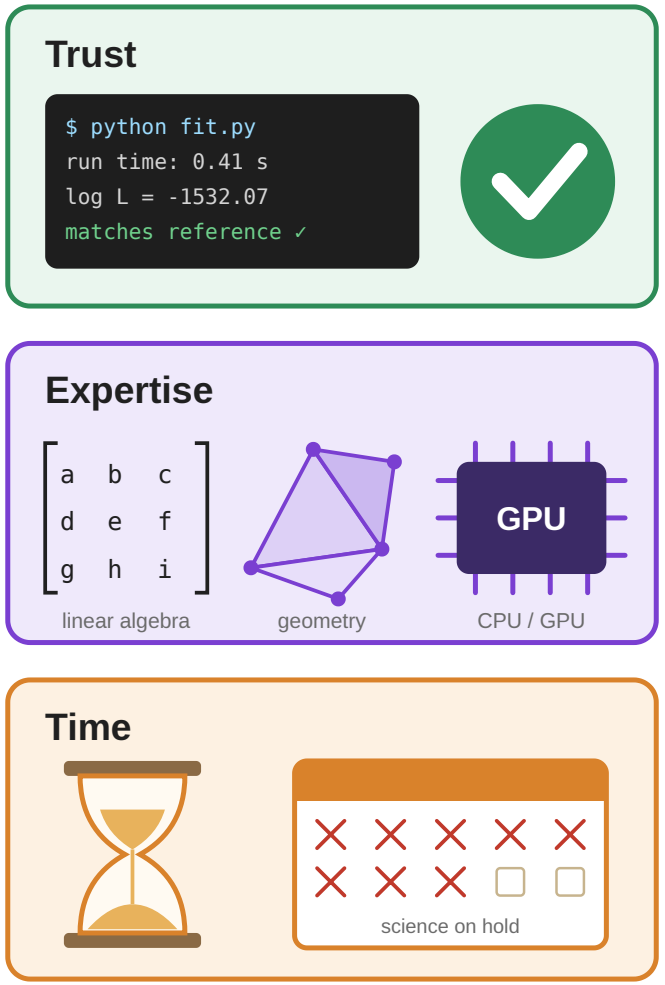
After a few months of using agentic AI, I realised this process would soon be **obsolete**: the sculptor would have to **hang up their tools**. 🪦



Three key aspects of this task

1. ✅ **Trust:** the code runs end-to-end, so you can instantly confirm the answer is correct.
2. 🧠 **Expertise:** optimizing code often draws on domain expertise beyond our own knowledge base. For me, this was linear algebra, spatial geometry and CPU/GPU programming.
3. ⏳ **Time:** the process is time intensive and trial and error, and sooner or later you need to get back to doing science.

This makes it the **perfect problem for agentic AI and loop engineering**. 🤖



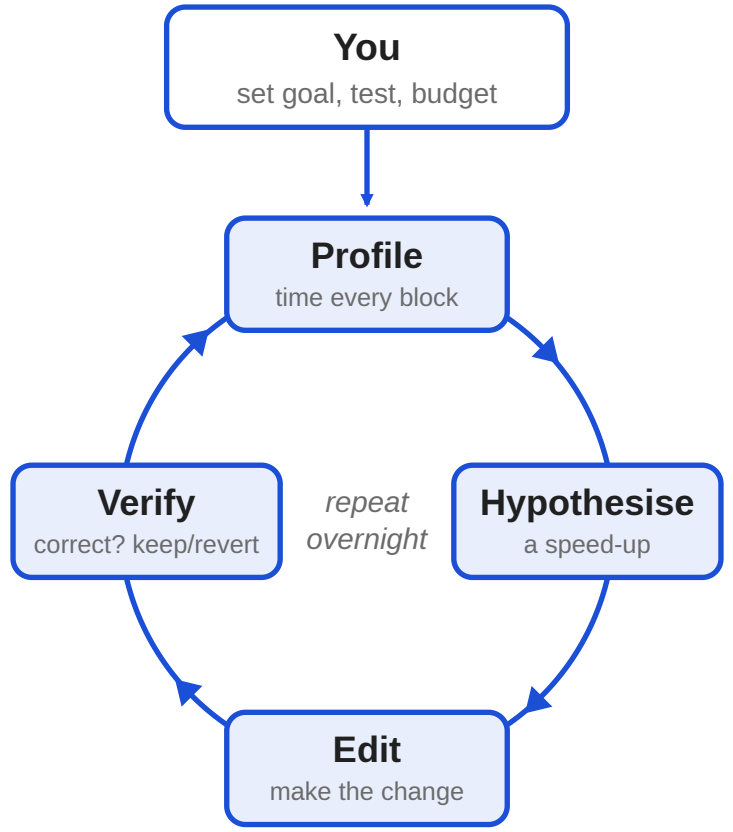
What is loop engineering?

Instead of turning the loop yourself, you **design the loop** and let an agent turn it. 🔁

- 📦 **GitHub repository for profiling**
- 📖 **Natural-language guide to the code I'm optimizing**
- ⏱️ **The timing script used by the AI agent**

I can have an agent run this loop: it makes the code faster and I'm happy. 😊 But when it finishes, **the next agent starts again from scratch**. 😕

The loop is not complete. What do I need to do to complete it? 🙋



Loop engineering
the agent turns the loop, you design it

Closing the loop with a wiki 📚

Remember how we gave the PyAutoLens assistant its knowledge? **Wikis.**

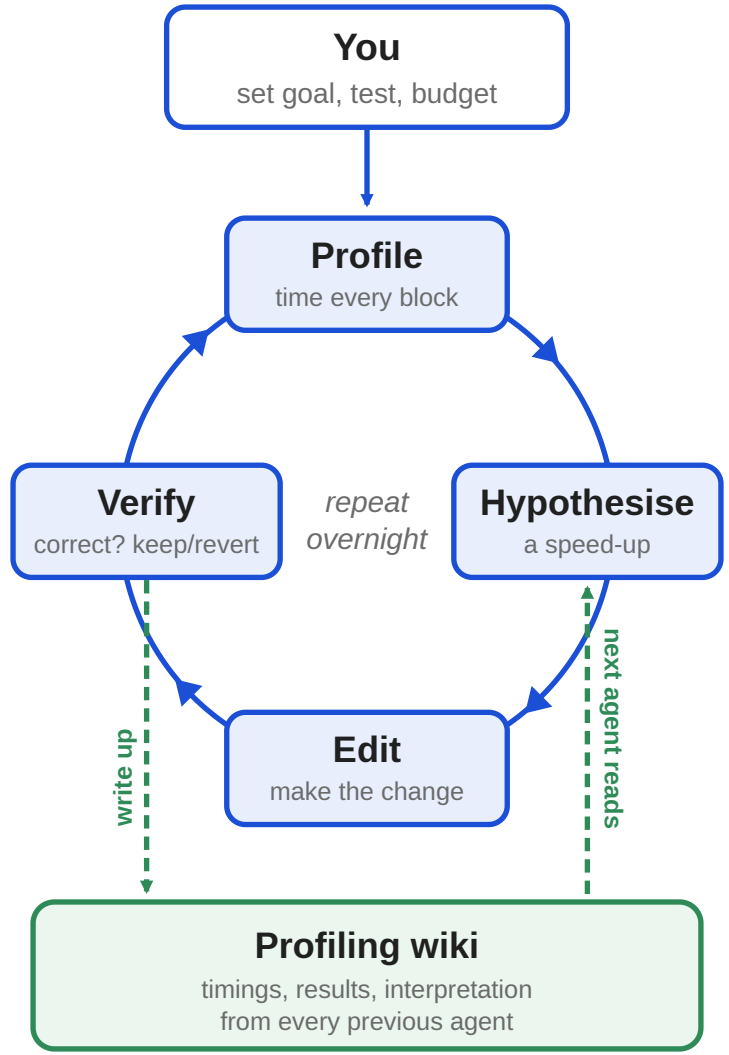
Now the agent **writes up** its work in a profiling wiki:

- ⏱️ its **timings**
- 📊 its **results**
- 💡 its **interpretation** of the findings

When the next agent starts the loop again, it draws on the knowledge of **every previous agent**. 🔁

[Live Demo]

github.com/PyAutoLabs/autolens_profiling/tree/main/wiki



Two realisations

1. 🔁 **Run the loop 24/7.** PyAutoLens just keeps getting faster: **over 10× so far!** 🚀
2. 💬 **Stay in the loop.** Ask the repository, in natural language, what research it found and how the code sped up.

> I want to show in a workshop how I can use natural language to ask the autolens_profiling repository questions about speeding up PyAutoLens.
>
> Can you give a high level summary of how we have sped up the interferometry analysis recently?

The biggest gains came from doing less repeated work, then making the remaining work efficient on CPUs and GPUs. The September–October campaigns in `autolens_profiling` tell five main stories:

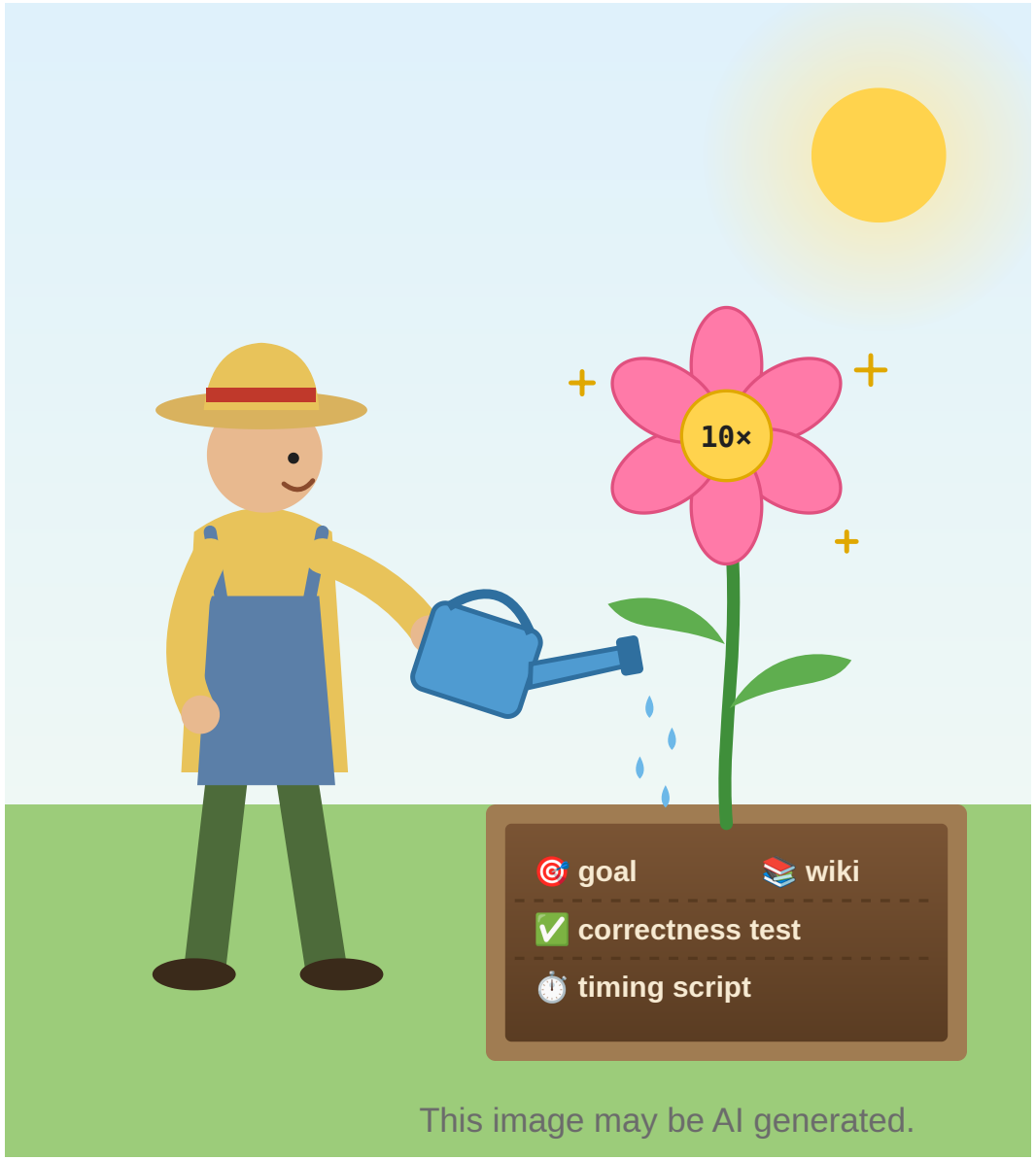
- **Avoid repeatedly processing millions of visibilities.** The *w-tilde* approach precomputes the data-dependent information, so subsequent model evaluations work with smaller image/source calculations. Extending this to multi-Gaussian source models produced dramatic A100 benchmarks: **ALMA ~940 ms → 2.7 ms**, and **JVLA ~24 s → 17 ms**. The older large-data comparisons used a chunked implementation because the dense route ran out of memory. Campaign record
- **Make the initial setup much quicker.** Building the reusable operator with a fast Fourier method reduced one recorded ALMA setup from **~35 minutes to ~7 seconds**. This improves the wait before fitting starts, separately from speeding up each model evaluation. Recorded implementation and measurements
- **Remove accidental repeated calculations and GPU bottlenecks.** Caching the curvature matrix and data vector cut measured NumPy evaluation times by **12–49%**. Changing an expensive GPU operation to handle real numbers before converting to complex numbers reduced an SMA dense-path benchmark from **~869 ms to ~3.4 ms**, with identical output. These are separate benchmarks, so their gains should not be multiplied together. Campaign results

From sculptor to gardener

The **sculptor** 🔨 carves every detail by hand.

The **gardener** 🌸 doesn't sculpt the flower. They create the environment and conditions for it to blossom.

Loop engineering is gardening.
Setting up the loop is preparing the soil and the environment. 🪴



Vibe Coding

Have you noticed visualisations getting better, or people using websites and new interfaces to interact with their results?

[Poll the room: hands up.]

[Live Demo]

COWLS public data: github.com/Jammy2211/COWLS_COSMOS_Web_Lens_Survey

PyAutoScientist: pyautolabs.github.io/cockpit/#scientist

Using Natural Language to Do Science

Natural language is becoming the **primary interface** for doing science.

- 💬 Natural language provides the **interface**.
- 🤖 AI provides the **capability**.
- 👩‍🔬 The scientist provides the **understanding, direction and judgement**.

Today: figures and papers 📊, an AI research assistant 🔭, loop engineering 🌱 and new ways to share results 🌐. **The syntax was never the science.**



AI Pitfalls

⚠️ Progress without understanding

With AI it is **very easy to make rapid progress without understanding what you are doing**. 🏎️

And with the **pressure to publish and perform**, we may take the fast option. 📈

This impacts **early-career researchers (ECRs)** the most. 🎓 I had 10+ years of matplotlib, LaTeX and HPC, and wrote PyAutoLens, before AI: **senior scientists could stand to benefit the most**. 🧠

- 🚀 **Senior scientists** can finally accelerate all the projects they could never wrap up.
- 😟 **ECRs** are left questioning their purpose, when any science analysis can be done in a week.



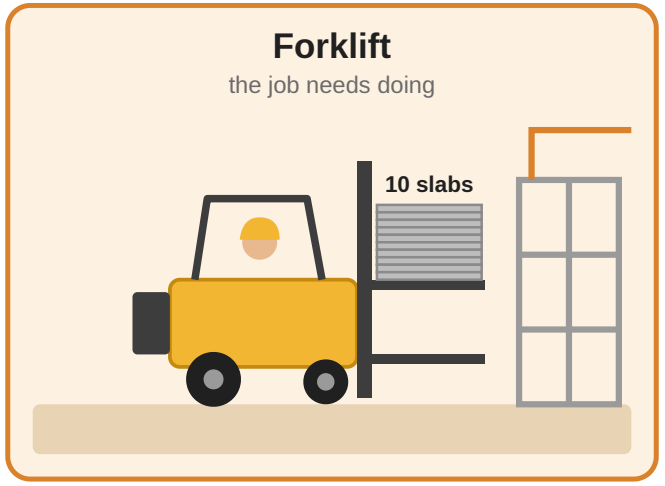
Weightlifting or forklifting? 🏋️🏗️

- 🏋️ **Weightlifting:** you go to the gym to work out. You *could* use a forklift to lift the weights, but what's the point?
- 🏗️ **Forklifting:** if you need to move 10 slabs of concrete to a building site, use a forklift.

With AI, **decide beforehand** whether you are weightlifting or forklifting.

As supervisors, we need to **actively encourage our ECRs to weightlift**. 💪





Analogy after science-fiction writer Ted Chiang, writing in The New Yorker: "Why A.I. Isn't Going to Make Art" (2024).

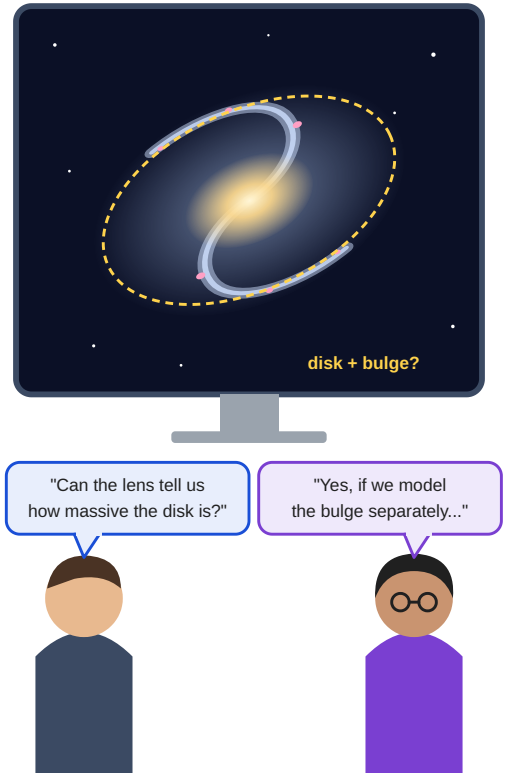
Will our jobs be automated? 🤖👨‍🔬

The thought has crossed my mind: *"Do I want a PhD student, or do I want a bazillion Claude tokens?"* 🤔

But the more I use AI, the more I want **more domain expertise, not less**. 🧠

To study **disk galaxies** with strong lenses (something I know little about), I don't want more Claude tokens. I want **an expert on disk galaxies** to work on the project with me, with natural language as the interface. 🤝

As of **November 2026**, I don't think AI agents are replacing the role of the human. They are **changing it**. 🔄 (Though I'd like the models to stop getting better quite so fast! 😅)



Unresolved challenges

- 🔒 **Data privacy.** Your data, code and ideas go to a giant tech company. We **must have an open-source AI ecosystem emerge**.
- 🌍 **Environmental impact.** GPUs use a lot of energy. But AI made my code faster, **reducing its energy use and emissions**. Hopefully AI can lead to efficiencies and technological breakthroughs. ⚡

Claude Team for Scientists

We have **25 free Claude subscriptions** for the year.

I am keen to lead the rollout, but we should **agree on how we approach it** first.

The syntax was never the science

Turning an idea into an analysis has always meant translating it through matplotlib, LaTeX, Slurm and Python. That syntax was only ever the **interface** between your thinking and the computer.

Now the interface is **natural language**. You say what you want in the words you already think in, the computer does it, and you keep the **understanding, direction and judgement**.