AI Tools in Science:
Large Language Models in Scientific Workflows

Ulrich Degenhardt

Draft Talk – 40 minutes

[1]Max-Planck-Institut for Dynamics and Self-Organization
Göttingen

Why This Talk?

- Scientists quietly use LLMs in their workflows.
- But LLMs are known for hallucinations and unreliability.
- Goal: Explore what LLMs can *actually* do in scientific workflows.
- Perspective: personal, experience-based, not systematic research.

Time: 0:30–2:00

Example 1: The “Magic/Miracle” Cartoon

- Vague memory: cartoon with scientists, equations, and “magic/miracle” in the middle.
- Google search fails: no precise keywords.
- LLM reconstructs the reference and finds Sidney Harris’s famous cartoon.
- Demonstrates fuzzy retrieval and context reconstruction.

Time: 2:00–3:00

Example 2: Solving a Math Problem from a Screenshot

- Screenshot of a linear algebra exercise in a PDF.
- LLM reads and interprets the mathematics (including LaTeX notation).
- Produces a full, seemingly correct proof.
- Key point: impressive reasoning behaviour — but correctness must be checked.

Time: 3:00–4:00

Early Lessons

- LLMs can do tasks that go beyond simple search.
- They appear to reason, summarise, and infer.
- At the same time: hallucinations are frequent.
- Takeaway: impressive *capabilities* but fragile *reliability*.

Time: 4:00–7:00

A Physicist's Skepticism About ML

- From a physics viewpoint, ML often looks like extreme curve fitting.
- Neural networks with $\sim 10^{11}$ parameters: "10th-order polynomial through 3 data points".
- Lack of conceptual simplicity and universality.
- ML/AI often feels epistemically unsatisfying compared to physical theories.

Time: 7:00–8:30

But It Works…

- Despite theoretical unease, ML works remarkably well in practice.
- Emergent behaviour at scale that no one expected 20 years ago.
- Physicists must grapple with tools that are powerful but opaque.
- This tension underlies the rest of the talk.

Time: 8:30–10:00

Practical Realities of LLMs

- Stochastic: same question can yield different answers.
- Very verbose: Brandolini's Law — debunking $>$ generating.
- Limited context window: they *forget* earlier parts of the conversation.
- Golden rule: ask questions that can be verified or falsified.

Time: 10:00–11:30

Infrastructure for Scientific Use

- Treat LLM interactions like experiments.
- Use note-taking tools (e.g. Obsidian) as a “second brain”.
- Export chats as markdown and store them as a lab notebook for AI.
- This makes results reproducible and reviewable later.

Time: 11:30–13:00

Behaviour Control

- Prompting can change global behaviour across a session.
- Examples:
  - Numbered dialogue format for structured discussion.
  - “No Yes-Man”: explicitly ask the model to challenge weak ideas.
  - *Advocatus diaboli*: temporarily activate an adversarial role.
- Aim: steer LLMs toward scientifically useful and critical behaviour.

Time: 13:00–16:00

LLMs and Anthropomorphism

- LLMs are conversationally fluent: they effectively pass the Turing test.
- Humans are primed to attribute mind, emotion, and intention.
- But LLMs have no consciousness, goals, or feelings.
- Empathy, politeness, humour: all simulated pattern generation.

Time: 16:00–17:30

System Trust vs. Social Trust

- Appropriate analogy: tools like `sed`, Linux, LAPACK.
- These tools are “battle-tested” in adversarial, high-pressure use.
- Trust is based on documented behaviour and known failure modes.
- LLMs should be evaluated as technical instruments, not colleagues.

Time: 17:30–21:00

Stage 1: “Google on Steroids”

- Paste compiler or interpreter error messages into the LLM.
- Get direct explanations and potential fixes.
- Ask “manual-level” questions (e.g. matplotlib legends, API usage).
- Saves time compared to manual search + StackOverflow.

Time: 21:00–22:30

Stage 2: Instant Prototyping

- Turn ideas into minimal working examples quickly.
- Example: path signatures from CSV time series with `iisignature`.
- LLMs can generate drafts in many languages: Python, Julia, C++, Fortran, Go.
- Barrier from “I have an idea” to “I have running code” becomes tiny.

Time: 22:30–24:00

Stage 3: Development Assistance

- Example project: Cryptomator-like encryption in Go.
- Long initial prompt describing requirements and constraints.
- LLM proposes architecture, cryptographic choices, and commands.
- Then an iterative loop: compile, copy errors, let LLM fix, repeat.

Time: 24:00–27:00

Leverage and Risks

- Feels like having a bright, tireless PhD student as an assistant.
- Greatly increases personal leverage in software development.
- But: easy to become lazy and stop checking changes carefully.
- Security and correctness require disciplined review — AI amplifies both good and bad.

Time: 27:00–29:00

AI for Literature Search

- Information abundance: hundreds of millions of papers.
- Semantic Scholar as a large bibliographic database.
- Tools like scienceOS build on top of it for scientists.
- Much better results for literature search than general-purpose LLMs.

Time: 29:00–30:30

Working with Semantic Scholar Tools

- Example query: “All publications of Ramin Golestanian from 2025”.
- Tool returns a curated list with short descriptions.
- Citation network visualisation helps see connections.
- Limitation: typically up to $\sim$ 100 sources per query.

Time: 30:30–32:00

Document Analysis and Evidence Tracking

- Upload many PDFs (100+) and ask questions across the whole set.
- Answers draw on both Semantic Scholar metadata and full text.
- scienceOS highlights exact locations in PDFs that support each answer.
- This explicit grounding increases scientific trust in the tool's output.

Time: 32:00–35:00

Why LLM Summaries Are Hard to Remember

- Generating summaries is easy; remembering them is not.
- Cognitive psychology: memory needs encoding, storage, and retrieval.
- Passive storage in Obsidian is not enough for long-term learning.
- Need spacing, retrieval practice, and active processing.

Time: 35:00–36:30

A Practical Workflow for Summaries

- Step 1: Ask LLM for a detailed summary of a paper.
- Step 2: Write your own 3-sentence meta-summary:
  1. What is this about?
  2. What is the core idea?
  3. Why is this important or surprising?
- Step 3: Ask LLM to generate a concept map with labelled edges.
- Combine text + visual map in your notes for better retrieval.

Time: 36:30–39:00

Final Reflections

- Hallucinations are not unique to LLMs; humans make mistakes too.
- Safety comes from critical thinking, scientific method, and engineering practice.
- LLM suggestions are starting points, not final answers.
- Open questions:
  - Better evidence for behaviour-control strategies.
  - Using LLMs as intellectual sparring partners.
  - Understanding GitHub repositories with AI.
  - Richer descriptions of AI-assisted development workflows.

Time: 39:00–40:00