Final Project: Evaluating and Improving LLM Scientific Capabilities

CDS DS 595 · Spring 2026 · Boston University · v0.2

35% of final grade

1 Timeline

Date	Milestone
Mon Mar 30	**Project released.** Start (optional) team formation.
Fri Apr 3	**Teams finalized.** Students not in a team will be assigned.
Mon Apr 13	**Proposal due**. Instructor feedback within 3 days.
Apr 13–28	Build eval, collect data, fine-tune, iterate.
Wed Apr 29	**In-class presentations**.
Fri May 1	**Writeup due.**

2 Overview

In teams of 3, identify a **scientific capability** that current LLMs struggle with, build a **rigorous evaluation** for it, and **improve an open-weights model** on that capability. Improvement must include fine-tuning (SFT and/or RL via Tinker), but may also include scaffolding improvements such as better tool use, retrieval, or prompting strategies. Scaffolding alone is not sufficient — the project must involve fine-tuning.

The project has three phases:

1. **Propose** a capability and demonstrate that frontier models struggle with it
2. **Build** an evaluation benchmark and collect training data
3. **Improve** a 4–8B parameter model and measure the gains

Learning Objectives.

- Understand how scientific capabilities of LLMs are measured
- Design trustworthy evaluations with appropriate baselines
- Gain hands-on experience with LLM fine-tuning (SFT/RL)
- Get an understanding of the fine-tuning algorithms involved
- Develop intuition for token economics and compute budgets
- Learn to work effectively with LLMs as both tool and subject of study

3 What counts as a "scientific capability"?

Broadly interpreted! Almost any tasks which require domain knowledge and reasoning ability count. Some examples:

- Equivalence of complicated mathematical expressions – "is expression *(a)* the same as expression *(b)*"?
- Aspects of reading a scientific plot/figure correctly (requires a multimodal model)
- Estimating the order of magnitude of physical quantities (hard Fermi problems)
- Computing scattering amplitudes from Feynman diagrams
- "Forecasting", i.e. predicting future events given a knowledge of the past
- Saying "I don't know how to do this" rather than giving a wrong answer. (Behaviour-grounded capabilities count too!)

You are encouraged to draw from your own scientific background and interests. **The main requirement is that the task should be something that frontier LLMs can't already solve perfectly.** If you're unsure whether your idea qualifies, ask!

4 Team Formation

Teams of 3 (teams of 2 considered on request). Use Ed Discussion to look for and find teammates. Teams must be finalized by **Friday, April 3**. Students not in a team by then will be assigned.

When teams are set, accept the GitHub Classroom assignment link (to be shared on Ed). The first team member creates the team name; others join.

5 Tinker

Each team receives **$100 in API credits** for Tinker, a distributed fine-tuning platform. Some recommended small models (feel free to pick your own; the model lineup is documented here):

- `Qwen/Qwen3.5-4B`
- `meta-llama/Llama-3.1-8B`

Think about how to allocate your compute credits: which model, how much training data, how many training runs. A 500-example SFT run on a 4B model costs roughly $0.25, so you have plenty of room to iterate. Report your total spend and how it broke down in the writeup.

Credits: [TBD.]

Getting started: Sign up at `https://auth.thinkingmachines.ai/sign-up`. See the Tinker Cookbook for examples of supervised learning and RL workflows. If you use Claude Code, you can install the Tinker skill to get help navigating the API directly from your terminal.

6 Deliverables

6.1 Proposal (due Mon Apr 13)

A short document (`PROPOSAL.md` in your repo) containing:

1. **Capability:** One sentence describing the scientific task, what distinguishes it as an interesting and challenging task.
2. **Evidence of frontier failure:** ~ 5 concrete examples where a frontier model (Gemini, Claude, ...) gets the task wrong or performs poorly. Include the prompts and example outputs.
3. **Eval plan:** How will you measure success? What's the input format, output format, and grading method?

4. **Data plan:** Where will your training data come from? Rough estimate of size. Do you plan to do SFT, RL, or both? Any scaffolding improvements (tool use, retrieval, etc.)?

6.2 Presentation (Wed Apr 29)

$\sim$ 8 minutes per team, followed by questions. Cover:

- What capability you targeted and why
- Your evaluation design
- Training approach and what you tried
- Results: frontier model vs. initial open-weights model vs. your fine-tuned model
- Conclusions, what worked and didn't

6.3 Writeup (due Fri May 1)

The project GitHub repository containing all code, data, and a `README.md` writeup with the following sections:

1. **Capability.** What scientific task are you targeting? Why do LLMs struggle at it?
2. **Evaluation.** Description of your benchmark: how many examples, input/output format, how correctness is judged (exact match, rubric, automated checker). The eval script should be in the repo and runnable.
3. **Training.** What data did you use, how much, where did it come from? What model and method (SFT, RL)? How many tokens/steps? What did it cost?
4. **Results.** Three-way comparison on your eval:
   - A frontier model
   - The open-weights model before fine-tuning
   - Your fine-tuned model

   Did fine-tuning help? On what types of problems? Any failure modes or regressions?
5. **Token economics.** Total Tinker spend and how it broke down (model choice, number of runs, tokens trained).
6. **Conclusions.** What worked, what didn't, what you would do differently if you had more compute/resources (10x? 1000x?).
7. **AI usage.** Similar to previous assignments in the course, reflections on AI use.

7 Grading

Total: 35 points (35% of final grade). Eval design and fine-tuning scores will draw from both the writeup and the presentation.

Component (points)	Full marks	Needs improvement
Proposal (5 pts)	Concrete capability with clear evidence that models fail; feasible data and eval plan	Capability is underspecified; few or no failure examples; data/eval plan needs more detail
Eval design (10 pts)	Well-designed benchmark (50+ examples, held out from training data); automated or clearly defined scoring; clear input/output format; appropriate baselines	Eval is small or narrowly scoped; scoring criteria unclear; eval data not separated from training data
Fine-tuning (10 pts)	Sound training methodology; meaningful iteration (tried multiple approaches, hyperparameters, data strategies); token spend reported; analysis of what worked and didn't	Limited iteration; few training runs attempted; token spend or analysis incomplete
Presentation (5 pts)	Clear and well-structured; covers eval, training, and results; honest about what didn't work; handles questions	Some components unclear or missing; could better address questions about methodology
Writeup (2.5 pts)	All sections present and clearly written; repo is organized and reproducible; thoughtful reflection	Some sections incomplete; repo could be better organized; reflection is thin
Peer feedback (2.5 pts)	–	Teammates flag unequal contribution

8 Peer Feedback

Each team member will have the opportunity to fill out a short form flagging significantly unequal contribution from teammates. If a form is not submitted or no issues are raised, everyone receives full marks. Scores may be additionally adjusted individually if there is a consistent pattern of unequal contribution.

9 AI Tools

You may use AI assistants freely throughout the project for code, data curation, writing, and analysis. In the README, briefly note how AI was used.