

Group Project Guidelines

INF 385T: Post-Training of Large Language Models

Overview The group project is a semester-long investigation of whether and how an open-weight language model should be post-trained for a task your team selects. Teams will define a target behavior, build a credible baseline and evaluation suite, make an evidence-based training decision, and communicate the result.

A successful project does not need to produce a state-of-the-art model or a positive training result. A well-designed negative result is fully successful when the evaluation is credible and the conclusions match the evidence.

Central question Your project should answer:

For this task and evaluation setting, does post-training provide a meaningful benefit beyond a strong baseline? what evidence supports that conclusion?

The answer may be yes, no, or not yet. The quality of the evidence matters more than the direction of the result.

Project expectations Each project must include:

- a clearly defined task and target behavior
- a justification for considering post-training
- a baseline that is strong enough to make the comparison meaningful
- a documented evaluation set and evaluation procedure
- at least one post-training intervention when the preliminary evidence warrants it
- controlled comparison of the baseline and adapted model
- quantitative results where appropriate and qualitative error analysis in all cases
- discussion of limitations and failures
- a reproducible code repository

Teams are not expected to apply every method covered in the course. One well-motivated and well-evaluated intervention is better than several poorly controlled training runs.

Scope guardrails Choose a project that can be evaluated with the available time and compute. The following defaults are strongly recommended:

- Use one open-weight model family unless there is a clear reason to do otherwise.
- Focus on one primary task and one primary behavioral objective
- Establish one serious baseline before training

- Use one main post-training method (additional methods only when they answer a specific question)
- Prefer a smaller, audited dataset to a large dataset that the team cannot inspect.
- Avoid projects whose success depends on obtaining private, restricted, or personally sensitive data
- Do not make deployment claims that the course-scale evaluation cannot support.

Teams should discuss unusually compute-intensive or safety/data-sensitive plans with the instructor before beginning training.

Milestones

Proposal	5%	Sep 23	Establish a feasible task, baseline, and evaluation plan
Project checkpoint	5%	Oct 28	demonstrate concrete progress and identify blockers
Final presentation	10%	Nov 18	Present claims and evidence early enough to receive actionable feedback
Final report	25%	Dec 7	Submit the complete project after incorporating presentation feedback

Proposal The proposal should be concise (1–2 pages excluding references) and address:

1. **Task and motivation:** What behavior will you study and who or what setting motivates it?
2. **Why post-training:** Why might prompting or retrieval be insufficient? What evidence could change your mind?
3. **Baseline:** What model and prompting or retrieval baseline will you establish?
4. **Evaluation:** What is the evaluation unit, primary measure, held-out design, and validation plan?
5. **Data:** What training data may be used and what are its source, scale, quality?
6. **Initial method:** What intervention are you considering and why is it appropriate?
7. **Feasibility and risks:** What are the main compute, data, evaluation, or safety risks?
8. **Team responsibilities:** What will each team member initially own?

Project checkpoint Each team gives a 5-minute progress presentation followed by 2 minutes of public Q&A, supported by one structured checkpoint slide containing:

- the current task definition and any change since the proposal
- at least one baseline result
- the status of the evaluation set and how it has been checked
- the status of training data and the training pipeline
- one concrete artifact from a pilot run
- the largest current blocker or risk
- named owners and target dates for the work remaining before Nov 18.

A checkpoint is complete when the presentation and slide provide concrete evidence of progress. If a training run is not yet appropriate, the pilot artifact may instead demonstrate a completed data, evaluation, or infrastructure test. It must still show that the team has retired from a meaningful project risk.

Final presentation Presentations take place on Nov 18 so that feedback can influence the final submission. Exact presentation and discussion times will be announced after the number of teams is known. The presentation should communicate:

- target behavior and project question
- baseline and evaluation design
- data and post-training intervention
- the most important result
- representative successes and failures
- the strongest supported conclusion
- the most important limitation
- the revision or additional evidence the team now considers highest priority

Feedback and revision Presentation feedback is part of the project process. Teams will use peer, instructor, and TA feedback from final presentation to prepare the revised final submission. Examples of possible revisions include:

- additional evaluation or error analysis
- method or implementation correction
- narrower or better-supported claim
- clearer explanation or visualization
- improved documentation and reproducibility

Teams will prioritize revisions that materially affect the evidence or conclusion. You are not expected to rerun an entire training program during the final week when narrowing a claim or adding a targeted analysis is the more responsible response.

The final report should briefly identify the most important changes made after the presentation. Teams may decline to follow a suggestion when they provide a clear methodological or feasibility reason.

Revised final report and code The revised final report and public code repository are due Dec 7. A suggested report length is 4–8 pages excluding references and appendices; concise reports are welcome when they contain the required evidence.

Report structure

1. Problem and motivation
2. Related work or technical context
3. Data and provenance
4. Baseline and evaluation design
5. Post-training method
6. Results
7. Error analysis and regressions
8. Limitations and ethical considerations
9. Changes made after presentation feedback
10. Conclusion
11. Individual contributions and AI Use Statements

Repository requirements The repository should contain:

- a README with setup and reproduction instructions;
- environment or dependency information
- scripts or notebooks for data preparation, analysis, training, and evaluation
- configuration files or clearly recorded command-line arguments
- small samples or schemas when full data cannot be redistributed
- the evaluation rubric or judge prompts, where applicable
- a statement of known reproducibility limitations

Do not commit model weights (other than adapters) or credentials.

Teamwork and individual contributions All team members are expected to contribute substantively. Teams may divide responsibilities, but every member should understand the task, evaluation, main method, and central findings. The final report must include a short contribution statement for each team member. In rare cases, individual grades may differ when there is clear evidence of *substantially* unequal contribution.

Use of generative AI The course AI policy applies to every project milestone.

- Your team must choose the project task and domain yourselves
- Your team must author the evaluation prompts, rubrics, and gold labels. AI tools may be used to critique a draft you wrote
- Your team is responsible for interpreting experiments and writing the substantive analysis
- AI tools may be used for permitted activities such as (prose and code) editing and debugging when disclosed
- The final report must include an individual AI Use Statement

Please note that methodological use of a model (e.g. synthetic data generation, AI-generated feedback) does not remove the normal disclosure requirement.