Final Project

CS 395T: Foundations of Modern Generative AI
Fall 2026 Prof. Noah Golowich

1 Project overview

The purpose of the final project is to obtain hands-on research experience on some aspect of the course. The scope is broad: your project can address any topic covered in the course and can lean theoretical or empirical.

Projects may be done in groups of 1–3 students.

For a **theoretical project**, you might develop a theoretical model that helps explain the results of an empirical paper. For an **empirical project**, you might investigate whether an algorithm with provable guarantees suggests an improvement to a procedure used to train LLMs. A useful blueprint for empirical projects is to choose an existing paper, replicate its results at a smaller scale, and then explore improvements or modifications.

1.1 GPU resources

The following resources can help with small-scale experiments. Availability, usage limits, and prices can change; check the linked pages before planning your experiments.

- **Google Colab:** Offers free hosted notebooks with access to GPUs, subject to availability and usage limits. GPU type and session length are not guaranteed; paid options are also available.
  `https://research.google.com/colaboratory/faq.html`
- **Kaggle Notebooks:** Provides free GPU notebooks with usage quotas. Check your account's available accelerator quota when planning runs.
  `https://www.kaggle.com/docs/notebooks`
- **Lightning AI Studios:** Offers a free tier with GPU credits and paid usage beyond those credits. Check the current offer and credit conditions when signing up.
  `https://lightning.ai/pricing/`
- **Runpod:** Offers paid GPU rentals by the hour, which can be economical for short experiments. Check the selected machine's rate and any storage charges before starting.
  `https://www.runpod.io/pricing`
- **Modern laptop GPUs:** NVIDIA-equipped or Apple-silicon laptops can train small models, run quantized LLMs, and, with sufficient memory, perform LoRA fine-tuning.

Start with small models and short runs to estimate the resources you need. Save checkpoints and results regularly, especially when using free or interruptible sessions.

1.2 AI use

AI tools are permitted, but you should not have AI "one-shot" your project or its results. You should actively propose ideas and hypotheses, design ways to test them for empirical projects, and formulate problems and conjectures for theoretical projects. AI can assist with this process, but you are responsible for evaluating and verifying its output.

You must understand all aspects of your final project, report, and presentation, including any code, experiments, arguments, or proofs developed with AI assistance.

2 Project pre-proposal (due October 14, AoE)

Submit **1–2** paragraphs about each of **1–3** project ideas you have. The paragraphs should contain the same content as the proposal (below), but compressed; please include some relevant references in this stage. The point of this phase is for the course staff to provide some initial feedback to try to help point you in a promising direction.

If you choose to work in a group, **only one person needs to submit the pre-proposal** on behalf of the group. At the top of each submission, clearly list **your team's Canvas Project Group name** and **all group members' names and email addresses**, even if you are working alone.

Before submitting, each group (including groups of one) should sign up for one of the provided **"Project Groups" on Canvas**:

1. Open the course in Canvas, click **People**, and select the **Groups** tab.
2. Choose an unused group from the provided Project Groups and click **Join**. Share the chosen group's name with your teammates; each teammate should sign in and join that same group.
3. Check that the group's membership matches the names on your submission. If you are working alone, join an unused group as its sole member.

The deadline is October 14, 2026, at 11:59 p.m. Anywhere on Earth (AoE, UTC−12).

3 Project proposal (due October 28, AoE)

Submit a **1–2 page** description of your proposed project, containing:

(a) **Background:** The relevant prior work and the context for your project.

(b) **Problem:** The problem(s) you are trying to solve or the question(s) you want to investigate.

(c) **Approach:** How you will go about doing the project, including your planned experiments or theoretical approach.

Similarly to the pre-proposal, the point of the proposal is for the course staff to help point you in a promising direction.

The deadline is October 28, 2026, at 11:59 p.m. Anywhere on Earth (AoE, UTC−12).

4 In-class presentation

Give a roughly **10-minute presentation** on your project. Explain the motivation and research question, your approach, your main results, and what you learned.

5 Final report

Write a **5–10 page report** describing your project, structured like a NeurIPS research paper. A suggested organization is:

- **Abstract:** A brief summary of the question, approach, and main findings.
- **Introduction:** Motivation, the problem you study, and your contributions.
- **Background and related work:** The relevant prior work and how your project builds on or differs from it.

- **Methods or theoretical setup:** Your approach, with the definitions and assumptions needed to understand it.
- **Results:** For empirical work, describe the experimental setup, baselines, evaluation metrics, and findings. For theoretical work, state your results precisely and provide proofs of all theoretical results.
- **Discussion and conclusion:** Interpret your findings, explain limitations, and describe open questions or next steps.
- **References:** Cite the papers, methods, and other resources you used.

Adapt this suggested outline to your project.

6 Project ideas and topics

The following ideas are in no particular order. This list will be updated over the next few weeks. An idea/structure that may be effective for many of the below questions is as follows: while the papers listed below (and others you will come across on your literature search) typically experiment at large scales, you will (likely) have (much) less compute. But perhaps there is a way to replicate their findings in much smaller (perhaps synthetic) settings. Doing so will require thorough understanding of the findings & techniques in prior work together with careful design of experiments at the scale you *can* experiment with.

1. What are some ways to increase diversity in LLM outputs?
   Starting point: Liwei Jiang et al., *Artificial Hivemind: The Open-Ended Homogeneity of Language Models (and Beyond).* `https://arxiv.org/abs/2510.22954`

2. How do reinforcement learning algorithms for LLMs respond to incentives to compress chain-of-thought (CoT) reasoning?
   Starting point: Pranjal Aggarwal and Sean Welleck, *L1: Controlling How Long A Reasoning Model Thinks With Reinforcement Learning.* `https://arxiv.org/abs/2503.04697`

3. How does the compute-optimal allocation between model size and training tokens change when data are repeated or contain different amounts of noise? Can you come up with simple (eg synthetic) distributions over sequences where you can experiment with this question at small scale?
   Empirical starting point: Hoffmann et al., *Training Compute-Optimal Large Language Models.* `https://arxiv.org/abs/2203.15556` .
   Theory starting point: *Scaling Laws in Linear Regression: Compute, Parameters, and Data*, Lin et al., `https://arxiv.org/pdf/2406.08466`

4. Under a fixed inference budget, when should a reasoning model revise an existing answer versus generate another independent answer, especially when the verifier is imperfect?
   Starting point: Charlie Snell et al., *Scaling LLM Test-Time Compute Optimally Can Be More Effective than Scaling Model Parameters.*
   `https://arxiv.org/abs/2408.03314`

5. Can you replicate the benefits of on-policy distillation at smaller scales? Assuming you can, what is the optimal balance of off-policy and on-policy data? Can it be beneficial to include other types of data (i.e., off-policy) which is *not* related to the target task?
   Starting point: Rishabh Agarwal et al., *On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes.*
   `https://arxiv.org/abs/2306.13649`

6. How do the order and number of tokens unmasked at each step affect the quality–speed tradeoff in masked diffusion language models?
   Starting point: Chen et al., *Optimal inference schedules for masked diffusion models.*
   `https://arxiv.org/pdf/2511.04647`

7. How does the distribution of tasks seen during training affect a transformer's ability to learn new tasks in context? For simple function classes (e.g., linear functions), can you identify distribution shifts where in-context learning fails even though a standard learning algorithm still succeeds?
   Starting point: Shivam Garg et al., *What Can Transformers Learn In-Context? A Case Study of Simple Function Classes.*
   `https://arxiv.org/abs/2208.01066`

8. When does reinforcement learning improve the probability of producing a correct answer while reducing the set of problems a model can solve with many attempts? Can you reproduce this

tradeoff on synthetic reasoning tasks, and test whether encouraging exploration during training improves both measures?
Starting point: Yang Yue et al., *Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?* (2025).
https://arxiv.org/abs/2504.13837

9. How reliably can experiments with small models predict which training-data mixtures work best for larger models? Under a fixed compute budget for selecting a mixture, is it better to test more mixtures with smaller models or fewer mixtures with larger models? Can synthetic data sources help explain when the ranking of mixtures changes with scale?
Starting point: Qian Liu et al., *RegMix: Data Mixture as Regression for Language Model Pre-training* (2024).
https://arxiv.org/abs/2407.01492

10. When can reinforcement learning with rewards unrelated to correctness nevertheless improve reasoning accuracy? Can you construct a small-scale setting where you control which reasoning strategies a model sees during pretraining, then test how this affects the benefits of random rewards versus correctness rewards?
Starting point: Rulin Shao et al., *Spurious Rewards: Rethinking Training Signals in RLVR* (2025).
https://arxiv.org/abs/2506.10947

11. When can a model improve its reasoning by repeatedly applying the same transformer block to its hidden states, without generating additional CoT tokens? Can you reproduce this effect on small synthetic tasks, and determine when using more iterations at test time than during training helps or hurts? Can an adaptive stopping rule achieve similar accuracy with fewer iterations?
Starting point: Jonas Geiping et al., *Scaling up Test-Time Compute with Latent Reasoning: A Recurrent Depth Approach* (2025).
https://arxiv.org/abs/2502.05171