CLOSING THE LOOP:
DESIGN PRINCIPLES OF MODERN LEARNING SYSTEMS

DUE 17:00, TUESDAY 1 SEPTEMBER 2026

1 Assignment

Optional reading. For broader context you may - but are not required to - look at two papers:

- Ouyang et al. (2022), *Training Language Models to Follow Instructions with Human Feedback* (InstructGPT) - https://arxiv.org/abs/2203.02155
- DeepSeek-AI (2025), *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning* - https://arxiv.org/abs/2501.12948

You do not need to read either paper in full: a short summary is enough (generated by ChatGPT). A detailed account of the methods is not graded - only the general logic matters.

Answer format. A short paragraph is expected for each question. Propose a solution (or several) and briefly explain the reasoning behind it.

1 You have a pretrained text-to-text LLM. Among other things, you can ask it to generate song lyrics. Suppose you want its outputs to consistently resemble a particular style - for example, the high-level characteristics associated with Max Korzh.

What are the different ways you could encourage this behavior? Think broadly: prompting, additional training data, fine-tuning, generating many candidates and selecting among them, using feedback from another model or a classifier, and so on.

Write a short paragraph describing as many approaches as you can think of. For each one, briefly explain *where* the feedback comes from.

2 Suppose you have a model that generates answers to a range of prompts, together with a reliable metric that scores how deceptive each answer is. Your goal is to build the most convincing liar: to encourage the model to produce false statements that are as plausible as possible. You can generate as many model outputs as you like and score them automatically with this metric, obtaining both high-scoring examples (“good” lies) and low-scoring ones (“bad” truths).

One simple approach is to keep only the high-scoring examples and fine-tune the model on them. Can the low-scoring examples also be useful? If so, how? What information do they carry, and what exactly is lost when we simply discard them? Describe any approaches you can think of.

3 Read the code accompanying Andrej Karpathy’s *Pong from Pixels*. You do not need to read the blog post itself - a short summary of it is enough. Identify the main components of the learning loop directly in the implementation: what is the agent, what does it observe, where do these observations come from, what actions can it take, what feedback does it receive, where does this feedback come from, and how is that feedback used to update the agent? For each component, copy and paste the relevant snippet of code and briefly explain what it does.

Then consider the “super-liar” LLM from the previous task. Assume you are given a large dataset of prompts asking factual questions and a function `score_text_to_be_a_lie(text)` that assigns a score to each generated answer. Identify the corresponding components of the learning loop in this setting - agent, observations, actions, environment and feedback - and describe how you would use the score to update the LLM.