
Assignment 3: Multi-task robot manipulation solutions

EN.601.498 / EN.601.698 · Hands-on Robot Learning · Fall 2026

Weight: 35% · Final in-class demonstration: December 10, 2026

Objective

The objective of this assignment is to produce a unified system (solution) that demonstrates *multi-task* capabilities. The keyword is *system* (or *solution*), meaning it does not need to be a single model (it of-course can, if that is the solution your team chooses to build). We allow teams to flexibly interpret the notion of a multi-task policy, and to allow them design freedom to explore aspects of the course they find most interesting. In general, we expect students to select one (or more) approaches starting from the multi-task policy learning module (Lecture 7 and later), or other closely related research.

The central requirement is for teams to demonstrate, explain, and analyze how their system works – i.e., how it encodes/represents, determines, and executes multiple tasks.

Some example approaches that are eligible: task-conditioned policy learning (e.g., large behavior models, vision-language-action models), finetuned / steered VLA policies, world models, world action models (WAMs), coding agents as policies, and harnesses/other agentic paradigms.

One exception, is that merely loading a set of pre-defined single-task policy checkpoints and switching through them does not, by itself, establish a multi-task solution. We care (and will quiz you about) what is learned-vs-adapted-vs-shared across the multiple tasks (and the answer can't be 'nothing'). Please consult the course staff before investing too much time into your team's solution to see if it is eligible.

Task suite and system design

Each time is required to specify:

- What their task definitions are (initial-conditions, object/distractors involved, success criteria, and other pertinent information needed to instantiate a new task)
- What their task specification interface is (i.e., how do you command the robot to do a particular task? Language? Something else – if so what, and how is that interface versatile enough to support multiple tasks?)
- What their training data (possibly) is (this may not always be known, especially if using a proprietary base model; but we require teams to state so if that's the case. Where finetuning a pretrained model, teams must specify what the finetuning data is and how it was curated)
- What measures they're taking to manage distribution shift
- What kind of compute the system is running on (to the extent known and possible)
- A short (1-3 sentences per team member) contribution statement indicating *who-did-what*.

Any modification to the robots require prior approval from the course staff. We will try and accommodate all *reasonable* modifications. For instance, if you'd like to design, print, and use a finray gripper opposed to the stock SO-101 gripper, we will ask you to show the proposed design/modification, and will ask you questions to understand whether you've thought about the implications of this design change on motor torques, policy learning, etc.

Evaluation and final demonstration

For each task, we require teams to come up with an evaluation protocol **before** running the final demonstrations and/or evaluations. We encourage teams to look at the Large Behavior Models technical report for an example of fine-grained task progress / success metrics using quantitative (predicates) and qualitative measures. Some ideas / measures: task success, subtask success, number of trials, initial-condition distribution, time-to-completion, number of recoveries, number of irrecoverable failures, etc.

During the final demonstration, which will be on the day of the final lecture for this class (December 10), we will have a small faculty panel comprising robotics faculty across the Whiting School of Engineering. The panel will visit each team's demo and will ask teams for a short technical description of their approach. Teams will be allowed to use a computer / laptop screen to showcase a small illustrative figure or diagram explaining the system, if needed. The panel will then step through the demo, and will ask what tasks the system supports, and may request that the team demonstrates tasks of their choosing. Teams will be evaluated along multiple axes, including, but not limited to demo versatility, originality, performance, system design, and generalization.

The faculty panel will then provide feedback to the course staff. The course staff will independently assess each team member's understanding of the *entire system*, regardless of the particular submodule they contributed to.

Submission and assessment

Submit one package per team containing:

1. A report of at most two pages (one page is better still), describing the requirements outlined in the "Task suite and system design" section above.
2. Any demonstrations, if collected for this task.
3. Any rollouts that were tried out before the in-class demo. Preferred in LeRobotDataset v3 format. A short text file describing what we're expected to see, and a short pointer to particularly relevant / interesting videos / episodes that the course staff should pay extra attention to (and why that is interesting). **This is an important component, as it will allow the course staff to fairly evaluate demos, particularly if performance on demo-day is far inferior to pre-demo-day performance.**

Awards

There will be awards for most performant policy, strongest generalization, most innovative demo, and a judge's choice award. These awards are a separate way of recognizing compelling demos, and are independent of grading (although there tends to be a healthy amount of correlation!)