Project Announcement

This project will be assessed based on the following components:

- **Mid-term written report** (Due: Nov 30 [Track 1]/**Nov 23 [Track 2]**, up to 5 pts)

- **In-class presentation** (Jan 2, up to 10 pts)

- **Final written report** (Due: Jan 9, up to 20 pts)

A team consists of **2–4 students**. All components are team-based. The total project score is 35+bonus points.

1 Mid-term Written Report

Each team should submit a brief report summarizing the current progress of your project. We recommend using the official *NeurIPS* template. The report should be about one page in length and must not exceed two pages. It should include the following:

- The selected project topic and the list of team members.

- A detailed timeline or schedule for the remaining work.

- The technical progress achieved so far, including methods implemented and any preliminary visual results.

2 In-class Presentation

Each group will present its project during the final session. The presentation should:

- Introduce the project's related works, motivation, setup, and methodology.

- Provide key experimental results, supported by visuals or demos.

- Include a **video demonstration** of your robot or simulation results.

- Highlight the strengths, limitations, and potential improvements of your approach.

The presentation will be graded based on clarity, technical depth, and demonstration quality.
Additionally, if a team successfully deploys **a real-robot policy** during an in-class presentation, it will earn **up to 3 bonus points**.

3 Final Written Report

Each team is required to complete a comprehensive experiment report that clearly presents the entire project workflow. The report should include:

- A clear description of the overall framework and methodology.

- Details of the training process, implementation, and evaluation metrics.

- Experimental results organized and presented using charts, tables, and figures.

- Discussion and analysis of results, including challenges and lessons learned.

Open-source code contributions are highly encouraged. We recommend using the official *NeurIPS* template for your report, though you are free to use your own formatting style.

4 Track 1: LeRobot SO-101

4.1 Overview

The final project is based on the **LeRobot SO-101** platform — a lightweight robotic arm with abundant open-source resources. Each team is required to design a project using SO-101 as the primary development platform. The course will provide opportunities for **real-robot deployment and testing**. We will prepare **3 benchmark tasks**, corresponding real-robot datasets, and provide real-world scene configurations so that you can replicate the setups in simulation.

Your goal is to use these resources to train, test, and deploy a policy that can achieve a reasonable success rate on the real robot.

Teams are also required to **define their own problems and methods** based on SO-101. If you offer a corresponding real-scene configuration, you can contact TAs to attempt **sim-to-real** deployment.

- **Simulation Reproduction:** Reproduce the same task and environment setup in simulation as on the real robot. Apply suitable methods to solve the task.

- **Deployable Policy:** Use the provided real-robot data and your simulation environment to train or fine-tune a policy that can be deployed on the real SO-101 robot.

- **Self-Defined Task and Method:** Design your own task and method in the simulation. After achieving stable results, you may contact TAs to test it in the real environment.

Figure 1: SO-101

4.2 Grading Reference

- **Simulation Reproduction:** Options include:

  - Build a gym-style simulation environment of 3 benchmark tasks (3×1 pts)
  - Use a suitable method to collect trajectories of 3 benchmark tasks with a success rate higher than 50%. (3×1 pts)

- **Depolyable Policy:** *A deployable policy* only uses RGB images, a prompt, and the robot's proprioception state. Options include:

  - Use suitable methods to get a deployable policy in the simulation. ($c_1 + c_2 + c_3$ pts) c_i depends on the success rate: 0 if $< 20\%$, 0.5 if $[20\%, 50\%]$, 1 if $> 50\%$.

– Use suitable methods to get a deployable policy in the real world. ($c_1 + c_2 + c_3$ pts) c_i depends on the success rate: 0 if $< 10\%$, 0.5 if $[10\%, 30\%]$, 1 if $> 30\%$.

- **Self-Defined Problem and Method:** Options include:

  – Provide a comprehensive review of the literature on this field or the related work of your own solution. (2 pts)

  – Define a new task and propose your own method. The score is calculated by $c_{\text{task}} \times c_{\text{method}} \times (c_{\text{success}} + c_{\text{analysis}})$ (6 pts max).

  c_{task} (3pts max) depends on task complexity:

  * Long horizon: > 60 seconds. (in simulation time) contains substeps to solve (1 pt)
  * Dual arm: design a task that needs two-arm coordination. (1 pt)
  * Fancy object: interact with soft or articulated objects. (1 pt)
  * Instruction-related: requires understanding or execution of instructions. **Without instruction, the robot cannot finish the task.** (1pt)
  * Obstacle complexity: the task contains significant obstacles that cannot be ignored and must be navigated or manipulated. Removing these obstacles would make the task substantially easier. (1 pt)
  * Other dimensions: you may propose additional aspects that increase task difficulty or novelty. You should write them in the **mid-term report**. These will be assessed by the TAs and the instructor. (up to 1 pt)

  c_{method} (2pts max) depends on

  * Difference from the existing codebase: your method must meaningfully differ from the existing code. If no difference is present, the entire c_{method} score is 0. If a valid difference exists, this earns 1 point.
  * Efficiency: compared to the baseline, your method achieves the task using fewer expert demonstrations or fewer interactions with the environment. (1 pt)
  * Generalization: your method demonstrates the ability to perform the task outside the training range, as verified by appropriate experiments. (1 pt)
  * Other dimensions: you may propose additional aspects that increase task difficulty or novelty. You should write them in the **mid-term report**. These will be assessed by the TAs and the instructor. (up to 1 pt)

  c_{success} depends on the success rate: 0 if $< 20\%$, 0.25 if $[20\%, 50\%]$, 0.5 if $> 50\%$.

  c_{analysis} (0.5 pt max) evaluates the depth of result interpretation, including discussion of failure cases, limitations, and insights gained from experiments.

  – Deploy your own task and method on the real robot. (2 bonus pts max)

4.3   About Real Robots

- **Real-world Settings:** The real-world settings will be released before November 16. These include the physical scene configuration, robot parameters, host machine specifications, and robot interface, allowing students to replicate similar results in simulation. The interface will follow a *Gym-style* environment, enabling the use of policy wrappers and environment wrappers for consistent code between simulation and real deployment.

- **Real-world Datasets:** The real-world datasets will be released before November 16. The datasets will be collected via teleoperation and organized in the **LeRobot Dataset** format. Each dataset entry will include the task name, RGB images from the wrist and third-person viewpoints, and the robot's proprioceptive observations. It is recommended that data collected in simulation be similarly organized in the LeRobot Dataset format to facilitate code reuse and seamless transition to real-robot deployment.

- **Rules for Usage:** Each team will have **one opportunity** to use the real robot **on-site** before submitting the final report, with a session duration of approximately **30 minutes**. Teams must contact a TA to schedule the session and demonstrate that their test scenario and method work successfully in simulation. During the course presentation, teams can optionally sign up for on-site real-robot demonstrations. In the final written report, providing detailed code instructions, model checkpoints, and test objects will allow the TAs to attempt real-robot deployment independently.

  Note: If your test task is not included in the provided benchmark scenarios, the team is responsible for preparing all corresponding physical assets or objects required for real-robot testing.

- **Real-robot Evaluation and Scoring:** The success rate for real-robot testing will be calculated from 10 deployment trajectories. For each team, the highest success rate among the on-site test, the course presentation test, and the TA's independent test will be used for grading.

4.4 Guidelines

Since the real-robot configurations and the project announcement are not released simultaneously, it is highly recommended that teams start familiarizing themselves with the development environment and the potential technology stack **as early as possible**. These preparatory activities do not require access to the real robot and can be done independently. Early preparation will help teams maximize their time when the real-robot resources become available.

The following are suggested resources and areas to study:

- **LeRobot SO-101 Information:** Detailed information can be found on the official website: `https://huggingface.co/docs/lerobot/so101`. Introduction videos are also available on platforms such as YouTube or Bilibili, which can help you become familiar with the hardware and capabilities.

- **LeRobot GitHub Codebase:** The official GitHub repository lerobot provides reference implementations and simulation environments. You can explore existing implementations of DP, ACT, π_0, SmolVLA, TD-MPC2, and HIL-SERL as examples for your project.

- **Simulation Environment:** ManiSkill includes official implementations of the SO-100 simple pick-and-place scenario, which can serve as a reference. Note that the course uses SO-101, so some adjustments may be necessary.

- **Policy Collection in Simulation:** You can use reinforcement learning, motion planning, or teleoperation to collect policies. For RL, popular codebases such as `stable-baselines3` or `CleanRL` are recommended. For motion planning, you may use libraries like `cuRobo`, `MPlib`, or the IK algorithms provided in modern simulators. For teleoperation, you can start with keyboard control, and if available, consider VR/AR devices or 3D navigation mice.

- **Importing Realistic Assets:** To introduce more realistic elements into your simulation, you may refer to existing digital twin projects such as RoboTwin, which guide importing physical assets and creating more realistic simulation environments.

5 Track 2: Propose Your Own Topic

You have the option to choose your own topic. For this project, you must follow these tips:

- Your proposal project should be related to embodied AI.

- You must discuss your proposal with TAs and get their approval.

- You **cannot** use a paper that you have already posted on arXiv or have published.

- **Mid-term Report Deadline:** Each team must submit a mid-term report summarizing their progress by **November 23**. This report should include your selected topic, team members, detailed project schedule, and technical progress achieved so far.

5.1   Requirements

The score for each requirement serves as a reference. We will adjust it based on your actual effort.

- **Literature review:** Provide a comprehensive review of the literature on this field or the related work of your own solution. (2 pts)

- **Run baselines:** Run the baselines in your setting. (4 pts)

- **Algorithm Design:** Design your own algorithm. (14 pts)

6   About Submission

Please pack your code and written report together in a `.zip` file and upload it to Web Learning: `https://learn.tsinghua.edu.cn/`. We highly recommend you start early! If you have any questions, please contact Chenhao Lu or Gu Zhang via WeChat or email (luch21@mails.tsinghua.edu.cn or zg24@mails.tsinghua.edu.cn).