Mid-term Report: Diffusion Policy for LeRobot SO-101 Manipulation Tasks

Embodied Artificial Intelligence 2025 Course Project
Guanheng Chen, Zuo Gou, Zhengyang Fan

1 Project Topic

Our project aims to implement and extend **Diffusion Policy** for all three official benchmark tasks—Lift, Stack, Sort—and a custom manipulation task, within a unified simulation environment (SAPIEN/Gym). Diffusion Policy is a recent imitation learning framework that models the robot's action distribution as a conditional diffusion process, enabling the capture of multi-modal behaviors and robust control in high-dimensional spaces. Compared to traditional explicit policies, it is better at handling ambiguous or multi-solution tasks. We use expert demonstration data for all tasks, and train the policy via supervised learning (behavior cloning) with a DDPM backbone.

2 Technical Progress

2.1 Methods Implemented

Our pipeline roughly consists of the following steps:

- **Data Processing:** Expert demonstration data including joint angles and gripper actions are normalized to $[0,1]$ using min-max statistics. Sequence chunking generates fixed-length action/observation windows for training.
- **Model Architecture:** The policy is a **Denoising Diffusion Probabilistic Model (DDPM)** with a multi-layer perceptron (MLP) backbone. The model receives as input the normalized observation O_t, a noisy action increment sequence $\tilde{\Delta a}$, and a time embedding τ_t (learned via a small MLP). The output is the predicted noise for each action increment.
- **Training:** The model is trained to minimize the mean squared error (MSE) between the predicted and true noise, plus a smoothness loss to encourage temporally smooth actions:

$$\mathscr{L} = \mathbb{E}_{t,\Delta a,\varepsilon}\left[||\varepsilon - \varepsilon_\theta(\tilde{\Delta a}, t, O_t)||^2\right] + \lambda_{\text{smooth}} \cdot \text{SmoothLoss} \tag{1}$$

  where $\tilde{\Delta a}$ is the noisy action increment, t is the diffusion step, and λ_{smooth} is a tunable weight.
- **EMA:** An exponential moving average (EMA) of model weights for stable evaluation.

2.2 Preliminary Results and Visualizations

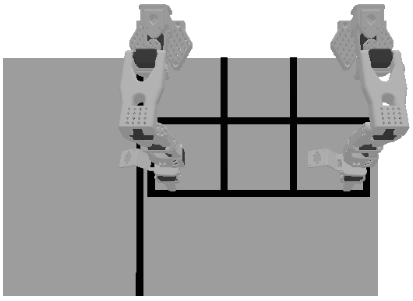

Figure 1: Simulation environment for all tasks. The workspace and two SO-101 robot arms are arranged to closely match the real-world setup, supporting flexible reconfiguration for Lift, Stack, Sort, and custom tasks.

To ensure sim-to-real consistency and enable multi-view perception, we set up three onboard cameras in the simulation: a front camera (with distortion to match the real robot), a left wrist camera, and a right wrist camera. These camera placements are consistent with the real robot hardware and provide diverse visual observations for both training and evaluation.

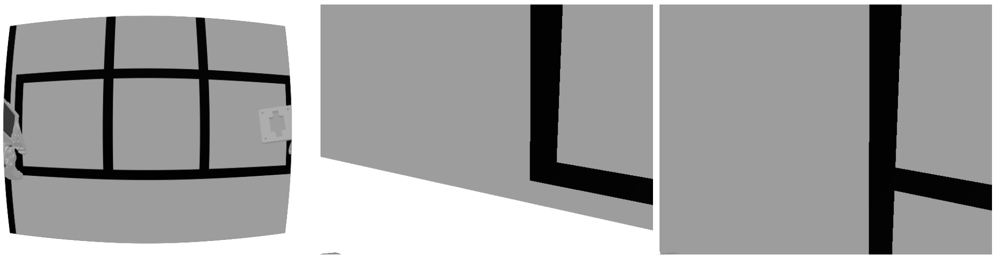

Figure 2: Sample images from the three onboard cameras in simulation.
Left: Front camera (with distortion), providing a global view of the workspace and both robot arms, closely matching the real robot's main camera.
Middle: Left wrist camera. Right: Right wrist camera.

In initial experiments on the Lift task, the agent is able to approach the target and move smoothly, but it cannot yet successfully interact with or manipulate the block. No successful lifts have been observed so far. We suspect this is due to gripper control or action protocol mismatch between the dataset and simulation. We plan to further debug the gripper logic and refine the action mapping. Meanwhile, the pipeline is being extended to Stack, Sort, and custom tasks, where similar evaluation and troubleshooting will be performed.

3 Remaining Work Schedule

We have developed a detailed timeline to ensure systematic progress across all required tasks. The plan is as follows:

Table 1: Project Timeline (Dec 2025 - Jan 2026)

Date Range	**Key Tasks & Milestones**
Dec 1 – Dec 7	**Simulation Environment Setup:** Complete the simulation environment construction for all benchmark tasks (Lift, Stack, Sort) and the self-defined task, including workspace, objects, and camera configuration.
Dec 8 – Dec 14	**Data Collection:** For each benchmark task, collect at least one set of demonstration trajectories with a success rate above 50%. Prepare and preprocess datasets for subsequent training.
Dec 15 – Dec 21	**Diffusion Policy Training:** (1) Train a diffusion policy using the provided datasets; (2) Train a diffusion policy using self-collected data. Conduct ablation studies if time permits.
Dec 22 – Dec 26	**Real Robot Deployment:** Deploy the trained policy to the real robot. Align the real robot and simulation environments, and perform sim-to-real transfer tests.
Dec 27 – Dec 30	**Policy Improvement:** If the policy performance is unsatisfactory, consider introducing reinforcement learning (RL) for fine-tuning. Iterate on data collection and training as needed.
Dec 31 – Jan 1	**Evaluation and Visualization:** Conduct comprehensive evaluation of all tasks (success rate, robustness, etc.), collect and visualize results, and prepare demo videos.
Jan 2	**In-class Presentation:** Present methodology, results, and video demonstration in class.
Jan 3 – Jan 9	**Final Report:** Compile and submit the final project report, including all experimental analysis and visualizations.

Timeline: Sim, demos, train, deploy, RL, eval, present, report.