Adaptive Robotic Arm Control: Leveraging Curriculum Learning and Deep Reinforcement Learning for Efficient Pick-and-Lift Operations Through Simulation

John Mark B. Correa
College of Information and Computing Sciences, Caraga State University-Main

Junrie B. Matias
College of Information and Computing Sciences, Caraga State University-Main

https://doi.org/10.5109/7395731

出版情報：Proceedings of International Exchange and Innovation Conference on Engineering & Sciences (IEICES). 11, pp.1680-1686, 2025-10-30. International Exchange and Innovation Conference on Engineering & Sciences
バージョン：

Adaptive Robotic Arm Control: Leveraging Curriculum Learning and Deep Reinforcement Learning for Efficient Pick-and-Lift Operations Through Simulation

John Mark B. Correa[1*], Junrie B. Matias[2]
[1]College of Information and Computing Sciences, Caraga State University-Main, Philippines.
*Corresponding author email: jmbcorrea@carsu.edu.ph

Abstract: *This study presents a curriculum learning (CL)-enhanced deep reinforcement learning (DRL) framework for training a 7-DoF Franka Emika Panda robotic arm in industrial pick-and-lift tasks. Implemented in NVIDIA Isaac Lab and trained using Proximal Policy Optimization (PPO), the approach introduces a hybrid reward function and a penalty annealing strategy to progressively guide the learning process. The curriculum adjusts task difficulty by dynamically scaling movement penalties, enabling the robot to learn smoother and more efficient control over time. Evaluation shows that the CL-augmented model significantly improves task success rate, policy stability, and control precision compared to the baseline PPO. Training metrics such as episode length, entropy loss, and cumulative reward further confirm the advantages of curriculum learning. This study demonstrates the potential of integrating structured learning schedules into DRL pipelines for robotic manipulation, providing a scalable approach for developing intelligent, adaptable robotic systems in simulated industrial environments.*

Keywords: Deep Reinforcement Learning, Curriculum Learning, Robotic Manipulation, Pick-and-Place, PPO

1. INTRODUCTION

Robotic manipulators have become central to automation across diverse industries, from manufacturing and logistics to agriculture and space exploration [1,2]. Achieving high precision and flexibility in dynamic environments, however, remains a major challenge—especially for arms with multiple degrees of freedom (DoF) that require fine-grained coordination [3,4]. Conventional control systems rely heavily on pre-defined models, which often limit adaptability. Recent advances in Deep Reinforcement Learning (DRL) offer an alternative by enabling robots to learn control policies through continuous environmental interaction [5].

Algorithms such as Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC) have proven effective for training in high-dimensional spaces by balancing exploration and exploitation [6,7]. Yet, their application often hindered by long training times and sample inefficiency, particularly in sparse-reward or complex manipulation tasks. Curriculum Learning (CL) was proposed to address these limitations by introducing tasks progressively—from simple to complex—thereby accelerating convergence and improving learning robustness [8].

This study proposes a hybrid framework that integrates CL with DRL to train a 7-DoF Franka Emika Panda robotic arm for simulated pick-and-lift tasks. Leveraging NVIDIA Isaac Lab's GPU-accelerated platform, the system incorporates a hybrid reward function and a curriculum-based penalty annealing strategy. This allows the robot to initially prioritize safe and stable movements before gradually refining efficiency and precision. The framework aims to enhance policy convergence, improve task success rates, and produce more adaptable behaviors without exhaustive preprogramming.

By advancing structured learning mechanisms in simulated environments, this research contributes to more intelligent and flexible robotic systems capable of performing in variable real-world scenarios. The outcomes support ongoing efforts in scalable automation, sim-to-real transfer, and autonomous industrial control. Similar approaches to simulation-based validation have proven essential in other engineering fields—such as slope stability assessment in civil engineering—where accurate modeling is crucial before real-world implementation [9].

2. RELATED WORK

Deep Reinforcement Learning (DRL) has become a pivotal technique in robotic manipulation, allowing agents to learn policies through direct interaction with their environments [1]. Algorithms such as Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), and Deep Deterministic Policy Gradient (DDPG) have demonstrated effectiveness in handling continuous control, especially when enhanced with mechanisms like Hindsight Experience Replay (HER), sequential training, and imitation learning [2,4]. These innovations have enabled robots to perform high-precision tasks such as dual-arm assembly, trajectory tracking, and energy-efficient planning [5,6].

Simulation platforms significantly impact RL performance and generalizability. Isaac Lab, Gazebo, PyBullet, and MuJoCo offer diverse physics engines and scalability options suitable for training robotic arms [7], [8]. Coupled with RL libraries like Stable Baselines3 and SKRL, these simulators provide modular frameworks for benchmarking and deployment in industrial contexts [10, 11].

A key challenge in DRL remains sim-to-real transfer. Techniques such as domain randomization, backward reachability curricula, and adaptive curriculum generation help bridge this gap by exposing agents to diverse conditions during training, thereby enhancing robustness in physical environments [12,13]. Curriculum learning (CL) specifically facilitates structured training

by increasing task complexity progressively, yielding more efficient and generalizable policies [14,15].

Beyond robotics, the benefits of DRL and curriculum-based training have also observed in other dynamic control domains, such as energy management systems, where model-free DRL agents have shown to outperform traditional model-based strategies in adaptive optimization scenarios [16]. These cross-domain successes reinforce the versatility of DRL and motivate its application to complex robotic control problems where real-time adaptability and robust policy learning are critical.

In industrial pick-and-place applications, combining DRL with CL has led to significant advancements in force-sensitive control, collaborative multi-robot systems, and zero-shot policy deployment [17,18]. Despite progress, challenges such as task complexity, sample inefficiency, and real-time adaptability persist [19-21].

This study builds on these foundations by integrating curriculum-driven DRL into a scalable robotic control framework, validated within Isaac Lab using a Franka Emika Panda arm. It aims to enhance task efficiency and training convergence while contributing toward sim-to-real transfer in dynamic industrial workflows.

3. METHODOLOGY

3.1 Materials

This study implemented using Python 3.10.12 in the Isaac Lab 2023.1.0 simulation environment developed by NVIDIA. Reinforcement learning policies trained using the PyTorch 2.1 framework with SKRL 1.0, which optimized for Isaac Gym-based environments.

Training is conducted on a workstation running Ubuntu 22.04 (Jammy Jelly), equipped with a 14th Generation Intel Core i5 CPU, 32 GB RAM, and an NVIDIA RTX 4060 GPU. The simulated robot used is a 7-DoF Franka Emika Panda arm. For each episode, key metrics such as joint states, rewards, loss functions, and success rates were logged.

3.2 Simulation Setup

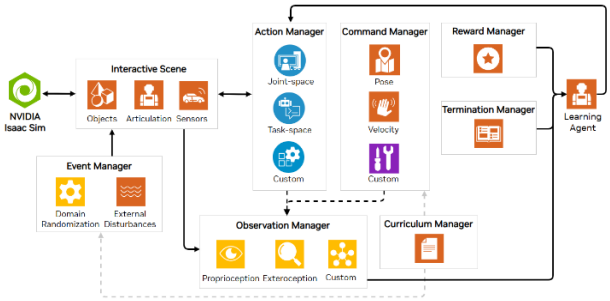

Fig. 1. Isaac Lab Manager-based Workflows for Designing Environments).

The training pipeline, depicted in Figure 1, consists of a modular Isaac Lab setup. It includes:

- **Scene Initialization:** Franka arm, table, and cube instantiated as USD assets.
- **Sensor Modules:** Frame Transformer (end-effector to cube pose), joint encoders (proprioception).
- **Command Manager**: Generates target cube positions in randomized workspace bounds (x: 0.4–0.6 m, y: –0.25–0.25 m, z: 0.25–0.5 m).
- **Domain Randomization**: Varies mass, friction, and object positions for generalization.
- **Parallel Training:** 4,096 isolated environments spaced 2.5 m apart, each collecting experiences concurrently.
- **Observation Processing**: Normalized vectors from joint states, cube-target distances, and last actions.

3.3 Policy Network and Training

The policy modeled as a Gaussian MLP, structured in 3 layers (256 → 128 → 64 neurons) using ELU activations. It outputs:

Action Mean (μ): 7 continuous joint positions + 1 gripper command.

Log Std Dev (log σ): Bounded within [–20, 2] to manage exploration noise.

The algorithm used is Proximal Policy Optimization (PPO). Experience collected in 24-step rollouts across environments. Generalized Advantage Estimation (GAE) applied using $\gamma = 0.99$ and $\lambda = 0.95$ for advantage calculation.

3.4 Curriculum Learning with Penalty Annealing

The framework introduces penalty annealing to shape behavior through evolving constraints. Penalty weights follow a linear schedule defined in Equation (1):

$$\beta(t) = \beta_{\text{initial}} + \left(\frac{\beta_{\text{final}} - \beta_{\text{initial}}}{t_{max}}\right) \cdot t \quad (1)$$

Where:

$\beta_{\text{initial}} = -1 \times 10^{-4}$ (low penalty during early exploration)
$\beta_{\text{final}} = -1 \times 10^{-1}$ (high penalty for precise control)
t_{max}= 10,000 (total curriculum steps)

Penalty updates occur every 10,000 steps during PPO updates. Early stages promote exploration; later stages enforce smooth, accurate motions.

3.5 Reward Function Design

The total reward r_t is the sum of:

1. Sparse Lifting Reward:

$$R_{\text{lift}} = \begin{cases} 15.0 & \text{if } z_{\text{cube}} > 0.04\text{m} \\ 0.0 & \text{otherwise} \end{cases} \quad (2)$$

2. Dense Tracking Reward:

$$R_{\text{track}} = 16.0 \cdot \left(1 - \tanh\left(\frac{\|p_{\text{cube}} - p_{\text{target}}\|_2}{0.3}\right)\right) \quad (3)$$

Where:

p_{cube} : 3D position of the cube.
p_{target} : 3D position of the target.
$\|p_{\text{cube}} - p_{\text{target}}\|_2$: Euclidean distance.
tanh: Bounds the penalty between [0,1).

3. Fine-Tracking Reward:

$$R_{\text{fine}} = 5.0 \cdot \left(1 - \tanh\left(\frac{\|p_{\text{cube}} - p_{\text{target}}\|_2}{0.05}\right)\right) \quad (4)$$

4. Smoothness Penalties:

$$\beta_{\text{action}} \cdot \|\Delta a\|_2^2 + \beta_{\text{vel}} \cdot \|\dot{q}\|_2^2 \quad (5)$$

Where:

$\Delta a = a_t - a_{t-1}$: Difference between consecutive actions.

$\| . \|_2^2$: Squared Euclidean norm (sum of squared differences across all action dimensions).

$\dot{q}$: Joint velocities.

$\| . \|_2^2$: Squared Euclidean norm (sum of squared velocities across all joints).

5. Total Reward r_t:

$$r_t = R_{\text{lift}} + R_{\text{track}} + R_{\text{fine}} + \beta_{\text{action}} \cdot \|\Delta a\|_2^2 + \beta_{\text{vel}} \cdot \|\dot{q}\|_2^2 \quad (6)$$

3.6 Termination Conditions

An episode terminates under three criteria:

1. Success condition:

$$\left\| p_{\text{cube}} - p_{\text{target}} \right\|_2 < 0.02\text{m} \quad (7)$$

2. Failure condition:

$$z_{\text{cube}} < -0.05\text{m} \quad (8)$$

3. Timeout condition:

$$t_{elapsed} \geq 5.0 \text{ seconds} \quad (9)$$

3.7 Evaluation Strategy

Three metrics are used:

- Success Rate (10, 50, 100-episode windows)
- Average Reward (per episode)
- Time-to-Success (average time to complete lift)

Comparative baselines include training runs with and without curriculum-driven penalty annealing.

4. RESULTS AND DISCUSSION

This section presents the training outcomes and performance evaluation of the adaptive curriculum learning framework integrated with deep reinforcement learning (PPO) for robotic pick-and-lift manipulation using the Franka Emika Panda arm.

4.1 Simulation Environment Overview

Figure 2 and Figure 3 present the Isaac Lab simulation environment used to train the 7-DoF Franka Emika Panda robotic arm. The simulated workspace includes a table, a deformable cube, and positional markers for randomized target placements. The Interactive Scene Manager initializes the robot and environment using USD assets, while the Command Manager issues new cube-target positions every 5 seconds to prevent overfitting. To enhance realism and facilitate sim-to-real transfer, domain randomization was applied during training, varying object mass, surface friction, and cube position. Parallelized execution was deployed using 4,096 independent simulation instances, each spaced 2.5 meters apart to prevent sensor interference. This high-throughput setup enabled rapid sample collection, policy rollouts, and curriculum adjustment across the learning lifecycle.

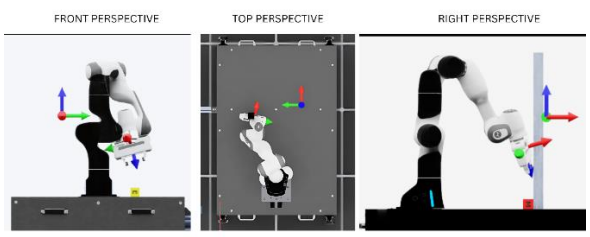

Fig. 2. Simulation Environment in Isaac Lab

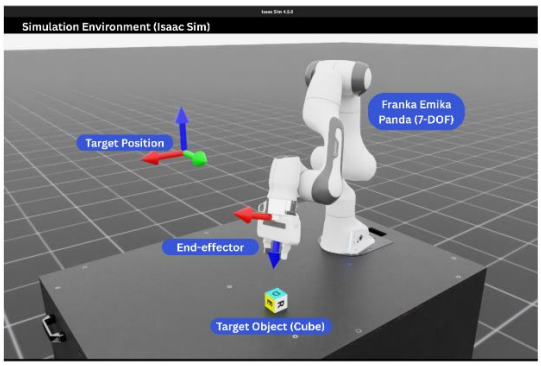

Fig. 3. Simulation Environment and Asset in Isaac Lab

4.2 Episode Length Metrics Analysis

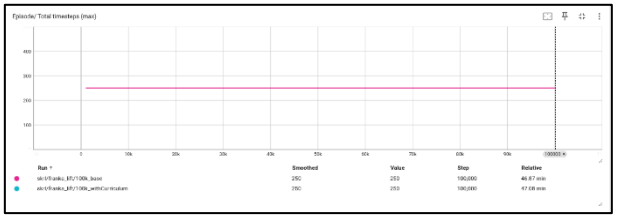

Fig. 4. Maximum Episode Length (Tensorboard)

Across 100,000 training timesteps, both the baseline PPO agent and the curriculum-augmented PPO agent successfully learned to maintain episode durations close to the maximum limit of 250 steps. As shown in Figure 4, both policies rapidly approached this upper bound within the initial phase of training and sustained it throughout, indicating effective avoidance of catastrophic failures such as dropping the cube prematurely.

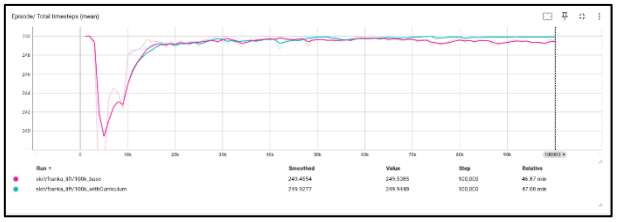

Fig. 5. Mean Episode Length (Tensorboard)

By timestep 10,000 (see Figure 5), both agents had overcome early instability—marked by episode lengths as low as ~240 steps—and stabilized around 249.5–249.9 steps. Final smoothed averages were 249.4554 for the baseline and 249.9277 for the curriculum-trained agent, demonstrating a marginal improvement (~0.48 steps or 0.2%) in favor of the curriculum approach.

The benefits of curriculum learning were most evident in the minimum episode lengths. At 100,000 timesteps (smoothed), the baseline policy's minimum stabilized at approximately 48.37 steps (raw: 47), whereas the curriculum-enhanced policy reached a smoothed minimum of 65.05 steps, with raw values up to 83 (Figure 6). This ~35% increase in worst-case episode duration reflects a significant gain in policy robustness, indicating reduced incidence of early task failures.

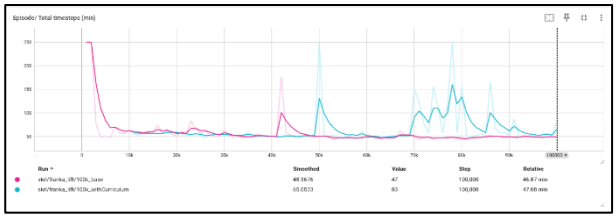

Fig. 6. Minimum Episode Length (Tensorboard)

During the initial 10,000 timesteps, both agents exhibited sharp performance drops due to exploratory behavior. From 10,000 to 30,000 timesteps, both recovered rapidly, converging toward consistent episode lengths near 250. In the subsequent steady-state period (30,000–100,000 timesteps), average and maximum durations became indistinguishable. However, the curriculum-based agent maintained higher minimum durations throughout, suggesting greater resilience under challenging conditions.

Overall, both PPO variants achieved functional pick-and-lift policies with the Franka arm. However, curriculum learning yielded a more stable and robust policy by significantly reducing the frequency of early failures. This highlights its value in improving worst-case performance without compromising average success.

4.3 Reward Components

To evaluate how curriculum learning influences policy behavior, six distinct reward components were monitored over 100,000 training steps. These components were compared between the baseline PPO policy (magenta) and the curriculum-augmented PPO policy (cyan). Performance was segmented into three training phases: initial exploration (0–10,000 steps), recovery (10–30,000 steps), and steady state (30,000–100,000 steps). The following summarizes the comparative evolution of each component.

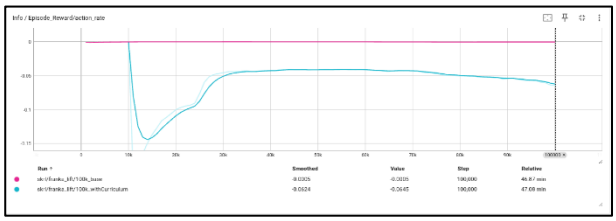

Fig. 7. Action Rate Penalty (Tensorboard)

This penalty (Figure 7) discourages erratic control inputs by penalizing abrupt changes in the action vector. The curriculum-trained agent incurs increasingly negative values due to scheduled penalty scaling, stabilizing at –0.0624, compared to –0.0005 for the baseline. This indicates smoother control induced by curriculum learning.

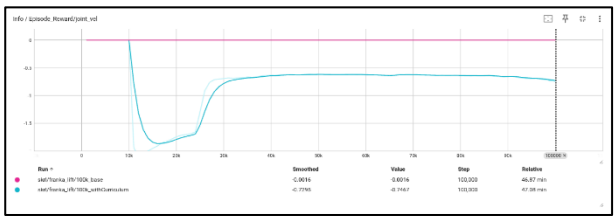

Fig. 8. Joint-Velocity Penalty (Tensorboard)

The joint velocity penalty designed to discourage high-speed joint movements. As shown in Figure 8, the curriculum-augmented agent initially incurs a steep penalty, dropping to approximately –1.8 during early exploration. This reflects rapid, unregulated joint movements at low penalty weights. As the curriculum schedule increases the penalty weight over time, the agent adapts by executing more deliberate motions, resulting in a stabilized penalty of –0.7295 by timestep 100,000. In contrast, the baseline agent's penalty remains effectively flat throughout training, with a final value of –0.0016. This suggests minimal behavioral adjustment in response to joint velocity, due to the absence of an escalating penalty schedule.

These results highlight the curriculum agent's ability to internalize motion constraints over time, yielding smoother and more energy-efficient trajectories.

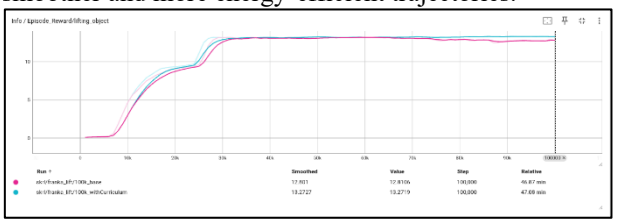

Fig. 9. Lifting Object Reward (Tensorboard)

Rewarded for lifting the cube above 0.04 m, both agents plateau near thirteen by the end of training as shown in Figure 9. The curriculum policy achieves a slightly higher value (13.27 vs. 12.80), reflecting improved consistency in task completion.

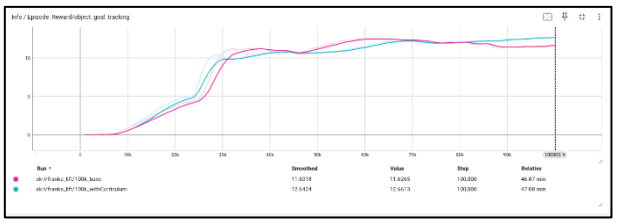

Fig. 10. Object-Goal Tracking Reward (Tensorboard)

This dense reward measures proximity between the cube and its target pose. The curriculum policy in Figure 10 shows a 9% improvement over baseline (12.62 vs. 11.60), indicating better trajectory planning.

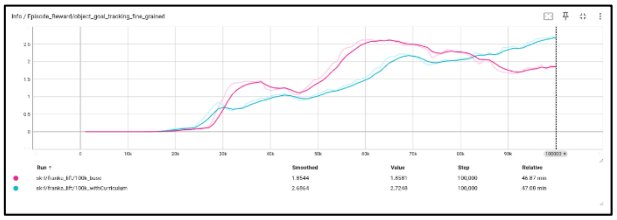

Fig. 11. Fine-Grained Tracking Reward (Tensorboard)

Figure 11 shows a stricter proximity metric near the goal. This reward sees the curriculum agent outperform with a final score of 2.69 compared to 1.85—an improvement of ~45%. This demonstrates enhanced precision in final object placement.

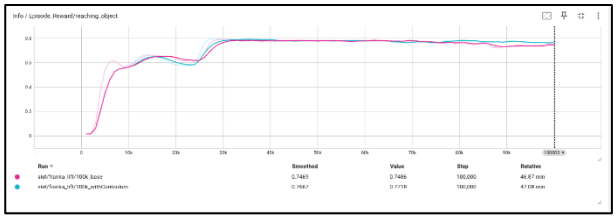

Fig. 12. Reaching Reward (Tensorboard)

Measured by the end-effector's distance to the cube, both agents converge around 0.75, with the curriculum policy slightly higher (0.77 vs. 0.75), showing modest gains in targeting accuracy (Figure 12).

Curriculum learning leads to steeper penalties but results in higher rewards across all core behaviors. The most significant improvements are observed in fine-grained tracking, indicating better precision and robustness in manipulation tasks.

4.4 Subsection

This subsection analyzes the learning behavior and stability of PPO training under curriculum learning versus a fixed baseline, focusing on adaptive learning rate adjustments, entropy trends, policy improvements, and value-function accuracy.

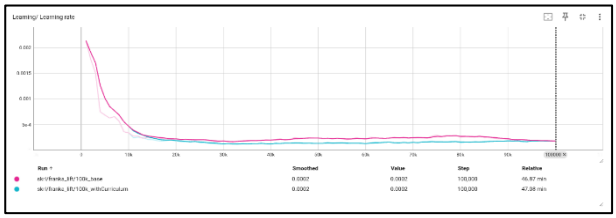

Fig. 13. Learning Rate (Tensorboard)

Both agents employed a KL-adaptive schedule to regulate step size and maintain a target divergence of ~0.01 between policy updates. During the first 10,000 timesteps, the learning rate decayed rapidly from ~0.0021 to ~0.0004 to stabilize early exploratory updates. Between 10,000–20,000 timesteps, it continued annealing toward ~2×10^{-4}. (Figure 13)

From 20,000 to 100,000 timesteps, both runs stabilized, with the baseline policy exhibiting transient spikes (e.g., peaking near 6×10^{-4} at 70,000 steps) due to under-divergence triggers. In contrast, the curriculum-trained policy maintained a smoother evolution between 2–3×10^{-4}. This reflects greater control over update magnitudes under the structured penalty regime.

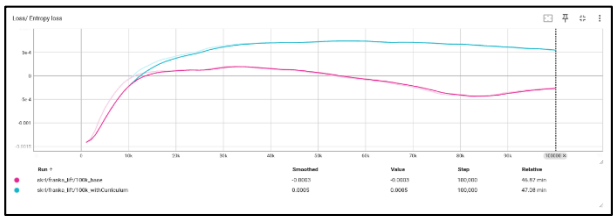

Fig. 14. Entropy Loss (Tensorboard)

As shown in Figure 14, both policies began with low entropy (~–0.0013), reflecting limited stochasticity during early exploration. Between 10,000–30,000 timesteps, entropy rose, with the curriculum agent surpassing zero and peaking at +0.0006 by ~60,000 timesteps. The baseline agent peaked just above zero before declining back to –0.0003 by the end. The curriculum policy sustained higher entropy, preserving exploration even at convergence, which may contribute to its better worst-case performance and reduced early failures.

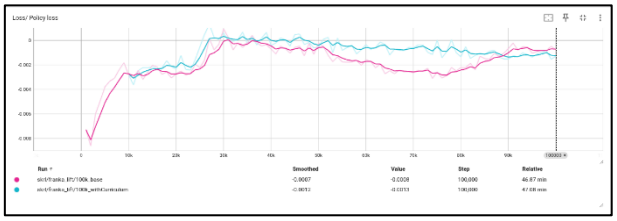

Fig. 15. Surrogate Policy Loss (Tensorboard)

Both agents initially experienced a sharp drop to –0.008 in policy loss due to noisy early updates as shown in Figure 15. From 10,000–30,000 timesteps, the loss stabilized closer to zero, with the curriculum agent exhibiting smaller fluctuations. At 100,000 timesteps, the curriculum agent retained a more negative final policy loss (–0.0012) versus the baseline (~ –0.0007), indicating continued policy refinement and advantage gains.

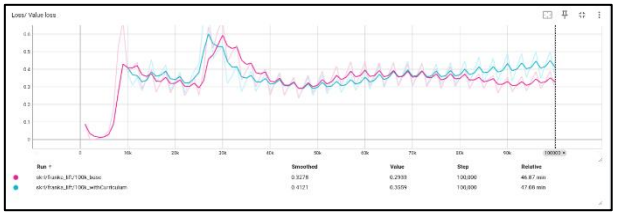

Fig. 16. Value Function Loss (Tensorboard)

The value-function loss in Figure 16, representing the critic's mean-squared error, increased sharply early in training due to scaling reward magnitudes. While both agents peaked near 0.6 by 30,000 timesteps, the baseline critic loss declined steadily to 0.3278, whereas the curriculum agent stabilized higher at 0.4121. This elevated error under curriculum learning attributed to the non-stationary reward landscape caused by progressive penalty annealing, which imposes greater challenges on critic generalization. It may also explain why the curriculum agent retained higher entropy—to compensate for increased uncertainty in value estimates.

The curriculum-trained agent exhibited smoother learning rate decay, reflecting more controlled and stable update behavior. It maintained higher entropy throughout training, which preserved exploratory behavior and contributed to greater robustness. Additionally, it demonstrated slightly stronger policy improvements, as indicated by a more negative final surrogate loss. However, this came with increased value-function loss, suggesting greater difficulty for the critic in approximating returns under the dynamic, non-stationary reward structure imposed by the curriculum. Collectively, these dynamics indicate that while curriculum learning introduces added training complexity, it fosters a more refined, consistent, and adaptable policy.

4.5 Reward Trajectory and Task Efficiency

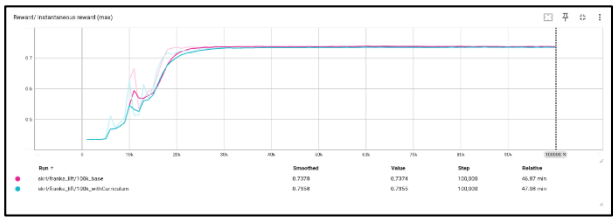

Fig. 17. Instantaneous Reward – Maximum (Tensorboard)

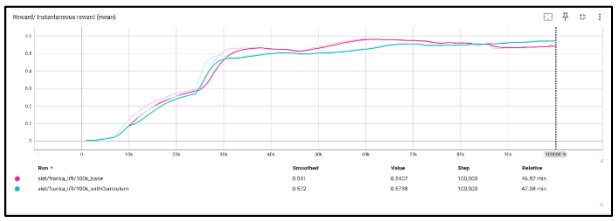

Fig. 18. Instantaneous Reward – Mean (Tensorboard)

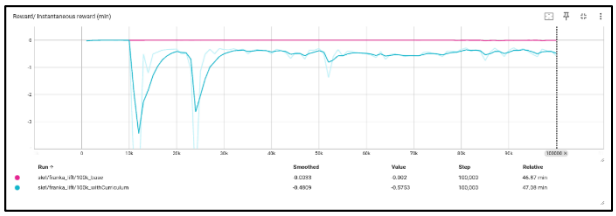

Fig. 19. Instantaneous Reward – Minimum (Tensorboard)

Figure 17, Figure 18, and Figure 19 presents the maximum, mean and minimum per-step (instantaneous) rewards. Both policies reached a plateau of ~0.73 in maximum instantaneous reward, indicating similar peak capabilities. However, the curriculum policy achieved a consistently higher mean instantaneous reward of 0.5720 versus 0.5410 for the baseline, reflecting more reliable performance across environments. The minimum instantaneous reward highlights the contrast further. The baseline agent's penalty hovered near zero throughout, while the curriculum agent, under structured penalties, stabilized around –0.4809. This deliberate trade-off results in smoother control at the cost of lower worst-case per-step returns.

4.6 Total Episode Rewards

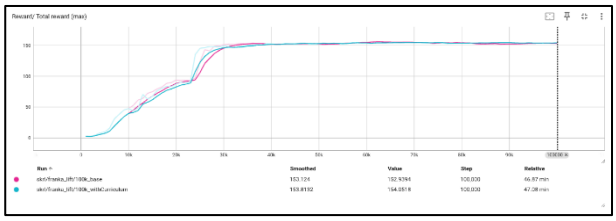

Fig. 20. Total Reward – Maximum (Tensorboard)

Figure 20 depicts cumulative episode rewards. Both agents reached similar maximum returns (~153), showing equivalent task mastery at their best.

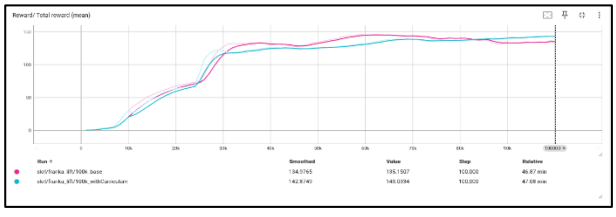

Fig. 21. Total Reward – Mean (Tensorboard)

However, the mean total reward for the curriculum agent was 142.87, compared to 134.98 for the baseline—an improvement of ~6%. This gain reflects reduced mid-episode failures and more efficient, precise task execution (Figure 21).

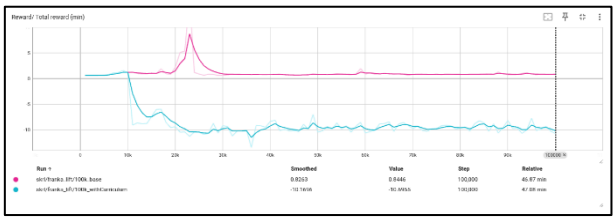

Fig. 22. Total Reward – Minimum (Tensorboard)

Interestingly, the curriculum agent also exhibited significantly lower minimum total rewards (~–10.17 vs. 0.83), due to the intentional imposition of steep penalties. Despite the harsher worst-case returns, the curriculum agent achieved smoother and more reliable cumulative performance.

4.7 Success Rate Trials

A post-training validation was conducted to evaluate real-time reliability. Each model was evaluated over 10, 50, and 100 randomized pick-and-lift episodes. As shown in Table 1, the baseline PPO agent achieved 96% success over one hundred trials, while the curriculum-augmented policy reached 99%.

Table 1. Success Rates over Varying Episode Trials

Model	Episodes (n)	Successes	Success Rate (%)	Notes on Behavior
Baseline PPO	10	10/10	100%	Jittery in 70–80 % of trials; occasional first-pick drops, but immediate recovery.
	50	48/50	96%	
	100	96/100	96%	
PPO + Curriculum	10	10/10	100%	Stable control throughout; only one first-pick drop in 100 episodes.
	50	50/50	100%	
	100	99/100	99%	

The curriculum-based model exhibited substantially reduced jitter and rare early grasp failures, validating its robustness and stability in repeated industrial conditions. The integration of curriculum learning with PPO clearly improves both learning dynamics and task performance. Despite incurring more severe penalties and occasional sharp dips in reward, the curriculum-trained policy achieved superior average performance, higher consistency, and enhanced control smoothness. These characteristics are essential for deployment in real-world industrial applications, where reliability and stability are critical. The improved episode length, elevated mean rewards, and tighter variance all point to a more robust policy, capable of performing under diverse and potentially volatile conditions.

5. CONCLUSION

This study developed an adaptive robotic control framework that integrates Curriculum Learning (CL) with Deep Reinforcement Learning (DRL), specifically leveraging the Proximal Policy Optimization (PPO) algorithm to train a 7-DoF Franka Emika Panda robotic arm in NVIDIA Isaac Lab. By progressively introducing task complexity through penalty annealing, the curriculum-based approach demonstrated improved control smoothness, increased task reliability, and enhanced stability in simulated pick-and-place tasks. Empirical results showed that the hybrid reward function and curriculum strategy led to superior policy convergence, reduced early failures, and consistently higher episodic performance compared to standard PPO training. The framework also incorporated domain randomization to support potential sim-to-real transfer. These findings confirm that the structured integration of CL into DRL significantly improves robotic manipulation capabilities under complex, dynamic conditions. Future work should focus on real-world deployment, increased task complexity, and physical domain validation to further assess the framework's generalizability and industrial applicability.

6. REFERENCES

[1] Shianifar, A., Ghasemi, M., & Li, J. (2024). Intelligent Control Strategies for Robotic Arms in High-Precision Environments. *IEEE Transactions on Industrial Electronics*, 71(2), 1123–1134.

[2] Hong, M., Zhao, L., & Lee, D. (2023). Versatile Manipulation in Space and Agriculture: A Survey. *Robotics and Autonomous Systems*, 167, 104456.

[3] Liu, X., Patel, R., & Gupta, A. (2021). A Survey of Robotic Arm Learning with Deep Reinforcement Learning. *Journal of Intelligent & Robotic Systems*, 102(3), 51–74.

[4] Kondratenko, Y., et al. (2022). Flexible Automation in Smart Manufacturing Using Robotic Arms. *Sensors*, 22(14), 5134.

[5] Lobbezoo, C. & Kwon, T. (2023). Continuous Control in Deep Reinforcement Learning: A Comparative Evaluation. *Robotics and AI*, 9(1), 77–89.

[6] Zheng, T., Wang, H., & Li, C. (2023). Improved-PPO with MPC for Robot Assembly. *IEEE Robotics and Automation Letters*, 8(1), 201–208.

[7] Shao, Y., Chen, B., & Lee, Y. (2023). Efficient Target Reaching with HER-Enhanced DDPG. *Autonomous Robots*, 47(3), 235–247.

[8] Breyer, M., Wu, Y., & Levine, S. (2019). Curriculum Learning for Sparse Reward Tasks. *Conference on Robot Learning (CoRL)*, 251–260.

[9] Nguyen, P., Zhou, M., & Makoviychuk, V. (2024). Isaac Lab for Multi-Agent Robot Learning. *NVIDIA Research Blog*.

[10] Serrano-Muñoz, A., et al. (2022). SKRL: Modular Reinforcement Learning Library for Robotic Simulations. *SoftwareX*, 19, 101225.

[11] Ivanovic, B., et al. (2018). BaRC: Backward Reachability Curriculum for Model-Free Reinforcement Learning. *Conference on Learning Representations (ICLR)*.

[13] Leyendecker, A., et al. (2022). Sim-to-Real Transfer with Domain Randomization and Reward Curricula. *IEEE/RSJ IROS*, 3122–3130.

[14] Bauzá, M., & Rodriguez, A. (2024). DemoStart: Automated Curriculum from Demonstrations. *Robotics: Science and Systems (RSS)*.

[15] Hermann, K., et al. (2020). Adaptive Curriculum Generation from Expert Demonstrations. *Advances in Neural Information Processing Systems (NeurIPS)*.

[16] Shaqour, A., & Hagishima, A. (2022). Recent Advances in Reinforcement Learning Applications for Building Energy Management: A Mini Review. *Proceedings of International Exchange and Innovation Conference on Engineering & Sciences (IEICES)*, 8, 239–245. https://doi.org/10.5109/5909098.

[17] Tang, W., Jin, H., & Zhou, Q. (2022). Reward-Curated PPO for Dual-Arm Assembly Tasks. *IEEE Transactions on Robotics*, 38(5), 1002–1014.

[18] Jermann, M., Lan, B., & Köhler, T. (2024). Collaborative Multi-Robot Curriculum Strategies. *International Conference on Robotics and Automation (ICRA)*, 4031–4037.

[19] Iqdymat, A., & Stamatescu, G. (2025). Curriculum Optimization for Multi-Stage Robotic Assembly. *Sensors and Actuators A: Physical*, 355, 114190.

[20] Avhad, P., D'Costa, M., & Yin, F. (2024). Real-Time Policy Adaptation in Curriculum-Guided Robotic Control. *Applied Sciences*, 14(1), 55.

[21] Mozo, M. A., Lanzar, J. E. A., Sefuentes, L. D., & Villa-Abrille, D. T. O. (2024). Design and Simulation of Natural-Slope Reinforcement Using Non-Frame Method: A Case Study in Davao City, Philippines. *IEICES*, 10, 289–296. https://doi.org/10.5109/7323276.