Autonomous Obstacle-Aware Motion for Pipetting Laboratory Robots Using Reinforcement Learning

Kamyar Barakati

000690767

Introduction

Automated liquid-handling systems have become the operational backbone of modern biomedical laboratories, supporting high-throughput applications in drug discovery, genomic preparation, clinical diagnostics, and combinatorial chemistry. Commercial platforms such as the Tecan Freedom EVO[1] (Figure 1), Hamilton STAR[2], Beckman Coulter Biomek[3], and Opentrons OT-2[4] enable precise and repeatable pipetting operations while maintaining sterility and processing efficiency. A critical limitation, however, underlies conventional workstation motion control: robotic trajectories assume that the laboratory deck remains geometrically fixed throughout operation. In practice, deck layouts shift frequently due to plate replacement, reagent replenishment, manual intervention, and module insertion. Any displacement of consumables or carriers without recalibration can lead to collision events, resulting in contamination, bent pipette needles, damaged tip cones, reagent loss, and execution delays. This work addresses this limitation by shifting motion planning from static, pre-scripted paths to perceptually informed, collision-aware behavior. A reinforcement-learning policy is trained using simulated LiDAR sensing[5] to infer free-space corridors, avoid unexpected deck obstacles, and navigate dynamically congested environments while reliably reaching pipetting targets (**Figure 1**). This framework replaces reactive halting with proactive spatial reasoning, enabling the robot to adapt to unplanned geometry changes without manual reset or recalibration.

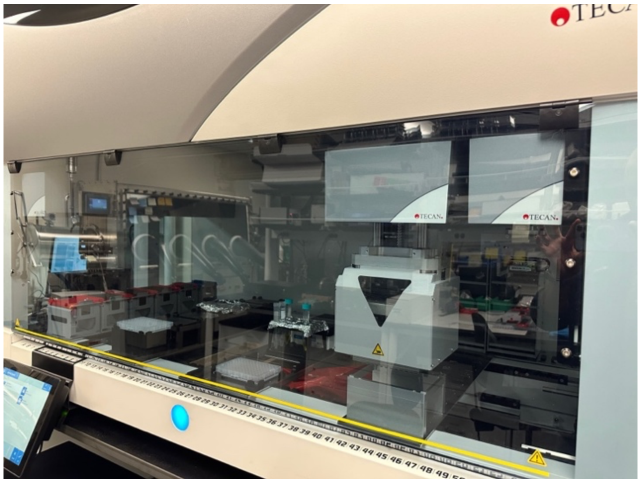

Figure 1*: Automated liquid-handling workstation (Tecan platform). The robot operates within a confined workspace containing microplate racks, reagent containers, and pipette heads. Changes in layout or unexpected obstacles can pose collision risks during robotic motion.*

Related Work

I. Reinforcement Learning for Navigation

Research in RL-based navigation has predominantly emphasized mobile robotic locomotion in expansive, obstacle-rich contexts. Canonical examples include outdoor sensor-guided ground vehicles and the BARN benchmark, which simulates cluttered indoor hallways for collision-avoidance policy development. These studies assume significant maneuvering area, omni-directional LiDAR perception, and relatively generous tolerance to path deviations. By contrast, pipetting instrument decks impose strict geometric confinement, sub-centimeter tolerance, and hardware fragility.[1]

II. Robotic Manipulation and Arm Safety

Recent RL investigations, particularly those from OpenAI and DeepMind, have demonstrated competent object manipulation, grasping, and dexterous control. However, these studies focus on intentional object interaction through grippers and manipulators. In this project, the objective is not to grasp but strictly to avoid physical contact with objects, thus inversely framing obstacle presence as a negative constraint rather than a target for controlled manipulation.[2,3]

III. Laboratory Automation Safety Principles

Safety mechanisms in commercial pipetting systems are traditionally encoded through emergency halts, software occupancy flags for deck slots, and protocol-level verification checks. While these methods reduce contamination risks and procedural errors, they do not provide real-time path adaptation when unexpected objects emerge. Instead, operators must manually intervene, or restart runs. RL navigation potentially removes this bottleneck by enabling platform autonomy rather than shutdown procedures.[3,4]

RL Environment

A custom PyBullet-based[5] simulation environment was constructed to emulate realistic pipetting deck constraints as represented in **Figure 2**. The environment is a bounded two-dimensional navigable square workspace in which walls represent deck boundaries and cubic obstacles emulate microplates, tip racks, reagent reservoirs, chillers, or user-placed containers. The robot is modeled as a circular agent capable of differential drive locomotion. A simulated LiDAR sensor emits seventy-two rays, providing continuous radial distance feedback.

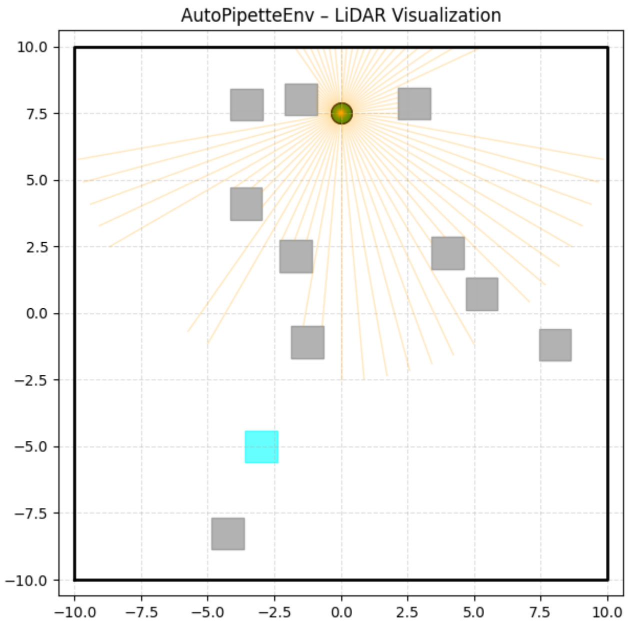

Figure 2: snapshot of the environment

***Table 1**: Environment Characteristics*

Parameter	Value
World size	1 m × 1 m (normalized laboratory workspace)
LiDAR rays	72 directional beams
Sensor field	360° sweep

Obstacle count	10–20 randomized placements
Robot motion type	Differential drive

The LiDAR perceptual system is critical in replicating the confined line-of-sight constraints present within robotic liquid handling decks, where objects cast sharp occlusion shadows and reflective surfaces induce significant spatial ambiguity. A key modification lies in LiDAR filtering logic: rays that contact the goal cube are replaced by maximum distance values to prevent the goal from being interpreted as an obstructive body. This adjustment ensures that the policy does not erroneously treat the desired dispense location as a collision zone. Furthermore, randomized obstacle generation across episodic resets induces deck variability, compelling the policy to generalize rather than memorize a static laboratory map.

Methodology

The navigation agent was trained within the custom AutoPipetteEnv simulator, which emulates a confined liquid-handling deck populated with discretized laboratory obstacles. The objective is to reach a fixed dispensing coordinate while avoiding physical interference with microplates, reservoirs, and tip carriers.

I. **Action Space**: The agent issues two continuous control signals

$$\boldsymbol{a} = [\boldsymbol{v}, \boldsymbol{\omega}] \qquad [1]$$

where $\boldsymbol{v}$ represents linear velocity and $\boldsymbol{\omega}$ rotational angular velocity.

Linear Velocity (v)	–0.7 to 1.0 m/s
Angular Velocity (ω)	–1.5 to 1.5 rad/s

II. Observation space: The observation vector consists of normalized LiDAR distances, normalized linear distance to target, and signed angular deviation from goal heading.

$$\boldsymbol{obs} = [\boldsymbol{LiDAR72}, \boldsymbol{dgoal}, \boldsymbol{\theta goal}] \qquad [2]$$

III. **Reward Function**: The reward structure integrates collision avoidance, approach accuracy, path smoothness, and terminal success.

***Table 2**: Reward Function*

Reward Component	Contribution
Progress toward goal	Positive proportional decrease in distance
Free space navigation	Encourages avoidance of congested areas
Proximity penalty	Large negative weight near obstacles
Smooth turning	Penalizes abrupt steering
Goal completion	High positive terminal reward
Collision	Immediate large negative termination

This dense shaping ensures navigational stability without overfitting to sparse reward spikes.

IV. **RL Algorithm:** Proximal Policy Optimization (PPO)[6] was selected because of its stability under high-dimensional continuous control.

***Table 3**: RL Algorithm values*

Hyperparameter	Value
Learning rate	$\mathbf{1 * 10^{-4}}$
n_steps	4096
Gamma	0.995
Clip range	0.30
Batch size	1024
Entropy coefficient	0.02

Eight vectorized environment instances were employed to accelerate training convergence.

Experiments and Results

Figure 3 illustrates a clear and structured progression in policy competence across training. The steady reward increase in Panel (a) indicates that the agent gradually learned to interpret LiDAR feedback not merely as raw distance samples but as spatial constraints governing safe corridor selection and goal advancement. Early instability gives way to consistent improvements as the policy recognizes when forward motion is viable and when proximity signals necessitate deceleration or detouring. Panel (b) reinforces this interpretation: episode duration decreases as hesitation loops, excessive turning, and premature avoidance behaviors diminish. Rather than reacting uniformly to any obstacle return, the agent learns to differentiate between benign clutter and true obstruction, enabling more decisive motion without compromising safety margins. Together, these trends show that PPO successfully internalized LiDAR-informed navigation logic. Training did not produce brittle, memorized behaviors but instead yielded an adaptive policy capable of modulating velocity, steering, and detour selection based on real-time visibility of free space. This convergence reflects both improved efficiency and stable collision avoidance in increasingly cluttered workspaces.

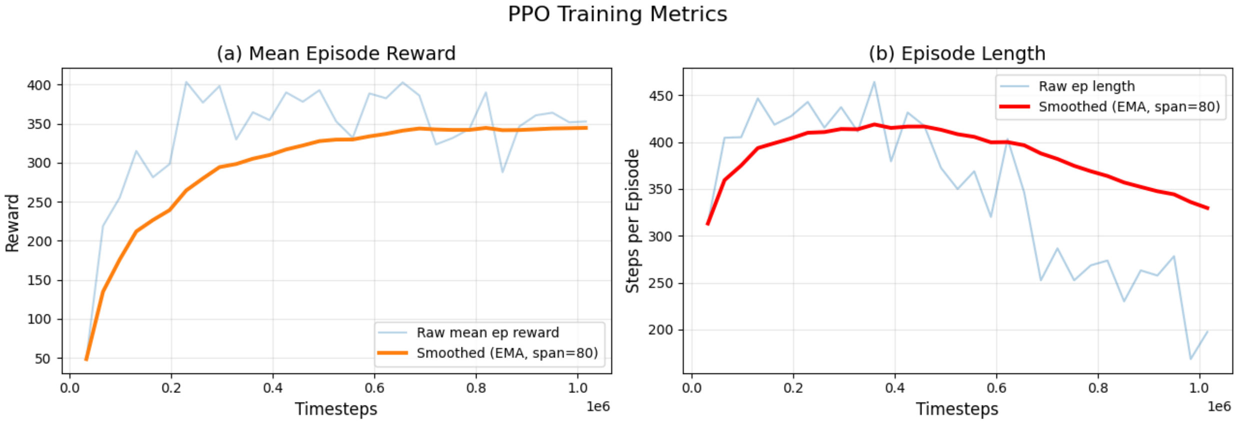

***Figure 3**: PPO Training metrics, (a) Mean episode reward increases steadily and stabilizes, indicating successful policy convergence and improved navigational decision-making. (b) Mean episode length gradually declines, reflecting more efficient goal-directed motion and reduced time spent in avoidance maneuvers.*

I. Simple Configuration (≈ 5 Obstacles)

In the first scenario, the pipetting robot navigated within a sparsely populated deck where only a limited number of obstructions were placed between the start path and the dispensing target. The broader free-space distribution resulted in minimal LiDAR saturation and fewer collision-range proximity reflections. Because the navigation corridors were wide and direct, the agent learned a straightforward visual-geometric association: as long as LiDAR rays returned extended distance values, forward motion remained feasible. Trajectory behavior in this phase demonstrated smooth velocity control with minimal angular oscillation. The robot consistently approached the target without exhibiting hesitation patterns or emergent avoidance loops. This confirms that the underlying policy architecture correctly learned to interpret low-entropy spatial feedback and did not overfit to unnecessarily cautious maneuvering when the environment allowed direct traversal. In **Figure 4**, the top-down LiDAR visualization shows clean beam propagation without dense ray occlusions, indicating unobstructed free space conducive to near-optimal navigation.

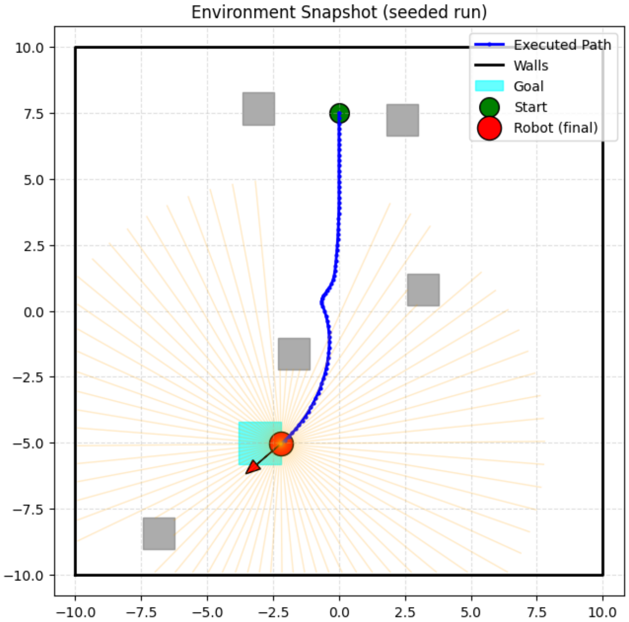

Figure 4: *trained agent trajectory with 5 obstacles*

II. Moderate Configuration (≈ 10 Obstacles)

The second phase introduced an intermediate degree of obstacle clustering. While a complete direct path remained theoretically available, the placement of racks, sample carriers, and reservoir blocks created localized choke points that forced the robot to compose brief avoidance maneuvers. Under this arrangement, LiDAR feedback exhibited comparatively noisier structures, with certain angular sectors returning near-collision values. Behaviorally, the policy shifted from pure goal-seeking to cascaded mapping between free-space selection and route shaping. The pivoting rate increased due to narrower operational corridors, yet the robot avoided degenerating into dead-end oscillation. This is visible **Figure 5**, where LiDAR rays begin to form cluster shadows corresponding to fully or partially occluded quadrants. The moderate configuration revealed the robustness of the reward structure: proximity penalties activated more frequently, preventing acceleration spikes near detectable boundaries. The progress-reward term, balanced with collision cost shaping, guided the policy away from fully greedy forward acceleration while retaining clear directional impetus toward the goal location.

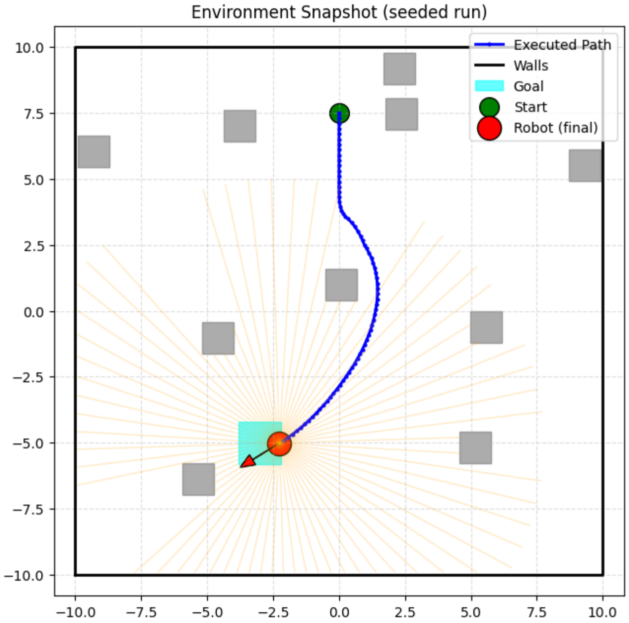

Figure 5*: trained agent trajectory with 10 obstacles*

III. Complex Configuration (≈ 15+ Obstacles)

The final experimental tier presented a highly congested pipetting deck in which the goal and obstruction geometry were positioned nearly contiguously. In several randomized layouts, the workspace resembled realistic laboratory behavior—top boxes and reagent platforms unintentionally overcrowded near target wells. Within this regime, LiDAR beams frequently returned saturation-level proximity values, producing conditionally conflicting navigation cues. As depicted in **Figure 6**, the complex arrangement occasionally created situations in which no forward-valid corridor existed. When obstacle geometry collapsed into near-goal adjacency, certain trials exhibited partial policy ambiguity. In those edge cases, avoidance logic dominated, and progress diminished, sometimes culminating in policy stagnation or looping avoidance while attempting to maintain safety margins. This illustrates the inherent tradeoff between strict collision avoidance and reliable target acquisition when geometric feasibility and safety constraints directly conflict. Nevertheless, a substantial portion of trained episodes succeeded even within these severe deck layouts. In successful trajectories, the agent executed multi-turn steering sequences resembling deliberate detour planning, wherein velocity decreased significantly, angular momentum was restrained, and LiDAR-derived free-space traces were dynamically re-evaluated at each timestep. The policy did not rely on memorized path templates but reacted adaptively to real-time spatial inputs, confirming reinforcement rather than scripted path replay.

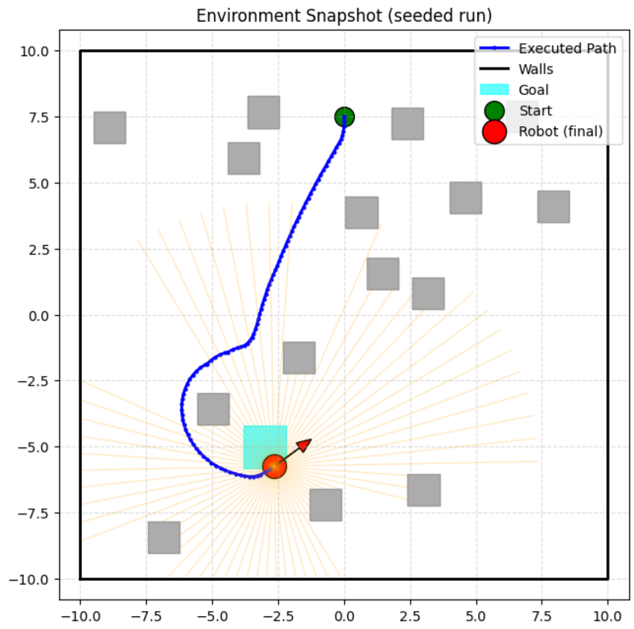

Figure 6: *trained agent trajectory with 15 obstacles*

The three-step experimental framework reveals a coherent ascent in navigational competency

Table 4: *Interpretation & Experimental Implications*

Stage	Layout Difficulty	Policy Behavior	Observed Outcome
Simple (5)	Wide corridors, minimal occlusion	Direct approach, minimal turning	High reliability, uninterrupted motion
Moderate (10)	Channel formations, partial occlusion	Balanced avoidance & approach	Stable performance with strategic detours
Complex (15+)	Goal adjacency, severe occlusion	Obstacle-dominant reasoning	High safety, occasional route deadlock

The progression confirms that the RL policy generalizes more strongly when exposed to incremental spatial entropy rather than learning from static deck geometry. The complex-case limitations also highlight the importance of future multi-sensor integration (stereo depth, tactile stop feedback, or semantic mapping) to assist in cases where deck reconfiguration physically removes any feasible collision-free path.

Conclusion and Future Work

The present work demonstrates that reinforcement learning–driven collision-aware navigation can successfully augment the motion intelligence of liquid-handling systems that historically operate under fixed spatial assumptions. The experimental results across simple, moderate, and highly congested laboratory layouts confirm that an agent equipped with LiDAR-based perceptual feedback can learn to navigate safely within constrained deck environments, even when obstacles are repositioned or introduced unexpectedly. In doing so, the robot transitions from executing rigidly pre-calibrated travel paths to exhibiting adaptive spatial reasoning, path re-evaluation, and dynamic avoidance behavior. The photograph of the TECAN liquid-handling workstation shown as **Figure 1**. illustrates the architectural reality that motivated this work: a dense operational chamber filled with microplates, reservoirs, tip racks, carriers, and peripheral modules, all of which occupy fixed spatial coordinates under traditional automation assumptions. While these systems excel in pipetting accuracy and precision, they lack the situational awareness necessary to respond when hardware placement changes between runs. The reinforcement learning approach developed here addresses precisely that limitation by enabling the robot to perceive the chamber, infer free-space corridors, and prevent collisions before executing transfer motion. In effect, the pipetting robot becomes not merely mechanically repeatable, but contextually intelligent and capable of adapting to physical variability inherent in real laboratory workflows.

A central next step involves transferring the trained navigation policy from simulation into the actual TECAN platform environment. This transition will require integration with existing motion controllers, calibration layers, and deck coordinate systems, but it will directly validate whether the spatial behaviors learned in AutoPipetteEnv are robust to real-world lighting reflections, transparent plastics, glassware edges, and metal carrier geometries. A real instrument deployment will also expose the learned policy to true laboratory disturbances such as human operator interventions, ad-hoc plate placement, and equipment servicing interruptions. Achieving reliable collision-aware motion in this context would meaningfully reduce downtime, limit consumable loss, and protect delicate pipette tooling. Furthermore, the same perceptual navigation architecture can be extended beyond pipetting heads to other automated sampling robots in the laboratory, including colony pickers, automated microbalance loaders, qPCR sample handlers, cryovial sorters, and bead-based extraction platforms. Any robotic system operating in a crowded bench-top zone with physically movable consumables can benefit from real-time sensing and adaptive motion planning rather than assuming static spatial programming. By integrating active perception and continuous spatial reasoning, these instruments can evolve from meticulously calibrated devices to resilient and autonomous laboratory co-workers capable of functioning under dynamic and unpredictable procedural demands. In future iterations, additional sensing modalities (such as stereo depth imaging, ultrasonic proximity feedback, or structured light scanning) may be incorporated alongside LiDAR simulation to enhance geometry resolution. Multi-goal motion planning could further enable the robot to execute sequential transfers across multiple plates, and domain randomization techniques may strengthen robustness to deck variability, lighting shifts, and reflective labware surfaces. Ultimately, the transitioning of RL-based collision-aware motion into real deck-top automation represents a substantial step toward laboratories in which robotic systems not only perform reliably under ideal calibration but continue to operate safely and intelligently even when the working environment changes unexpectedly.

All simulation code, trained policy artifacts, and experiment logs used in this study are available publicly on GitHub.

Reference:

1 Xuesu Xiao, Zifan Xu, Zizhao Wang, Yunlong Song, Garrett Warnell, Peter Stone, Tingnan Zhang, Shravan Ravi, Gary Wang, and Haresh Karnan, "Autonomous ground navigation in highly constrained spaces: Lessons learned from the benchmark autonomous robot navigation challenge at icra 2022 [competitions]," IEEE Robotics & Automation Magazine **29** (4), 148-156 (2022).

2 OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, and Alex Ray, "Learning dexterous in-hand manipulation," The International Journal of Robotics Research **39** (1), 3-20 (2020); Rongrong Liu, Florent Nageotte, Philippe Zanne, Michel de Mathelin, and Birgitta Dresp-Langley, "Deep reinforcement learning for the control of robotic manipulation: a focussed mini-review," Robotics **10** (1), 22 (2021).

3 Dong Han, Beni Mulyana, Vladimir Stankovic, and Samuel Cheng, "A survey on deep reinforcement learning algorithms for robotic manipulation," Sensors **23** (7), 3762 (2023).

4 Xuesu Xiao, Zifan Xu, Zizhao Wang, Yunlong Song, Garrett Warnell, Peter Stone, Tingnan Zhang, Shravan Ravi, Gary Wang, and Haresh Karnan, "Autonomous ground navigation in highly constrained spaces: Lessons learned from the barn challenge at icra 2022," arXiv preprint arXiv:2208.10473 (2022).

5 PyBullet, "PyBullet".

6 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347 (2017).