Learning Surgical Augmented Dexterity: A Reinforcement Learning Framework for Dual-Arm Robotic Control

Amog Rao, Ananya Shukla, Malhar Bhise, Utkarsh Agarwal
Plaksha University, India

{amog.rao, ananya.shukla, malhar.bhise, utkarsh.agarwal}@plaksha.edu.in

***Abstract* -** This paper presents a reinforcement learning framework for training robotic arms to perform precise needle handover tasks in surgical settings. Using the ORBIT-Surgical simulation environment, we segment the complex handover task into three distinct subtasks: approaching, grasping, and lifting the needle. Our approach employs Proximal Policy Optimization (PPO) with carefully designed dense and sparse reward functions that effectively guide exploration through key behavioral milestones. Experimental results demonstrate successful learning of dexterous manipulation skills through structured reward shaping, producing a policy that reliably achieves needle grasping and stable lifting to a predetermined height. The performance improvements across grasp success, lift height, and overall mean reward demonstrate that our framework effectively addresses the computational challenges of high-dimensional surgical skill learning, advancing the possibilities for automated surgical assistance.

***Keyword* -** Surgical Robotics, Reinforcement Learning, Proximal Policy Optimization, Reward Shaping

I. Introduction

Robotic-assisted surgery (RAS) has transformed modern healthcare by increasing precision, reducing intraoperative fatigue, and enabling minimally invasive procedures with shorter patient recovery times. Despite these advances, existing RAS systems predominantly rely on manual teleoperation and lack autonomy. Surgical automation has the potential to further optimize surgical workflows, reduce surgeon workload, and improve procedural consistency. However, automating dexterous, high-precision tasks such as suturing or instrument handover remains challenging due to their complex, sequential, and multi-modal nature.

Recent efforts [4-6] have introduced simulation platforms that support the application of machine learning for robotic control. However, many of these environments compromise on photorealism [7] or physical realism [8], and focus on relatively simple tasks such as reaching or placing. For more complex and interactive surgical scenarios involving multi-arm coordination and tool manipulation, there remains a need for structured learning frameworks and rigorous benchmarks.To address these challenges, ORBIT-Surgical [9] was introduced as an open-source, GPU-accelerated simulation platform built on NVIDIA Isaac Sim, offering high-fidelity physics, photorealistic rendering, and modular APIs for RL, IL, and teleoperation. Unlike earlier platforms, ORBIT-Surgical supports parallelized data collection, deformable object interactions, and realistic multi-arm control scenarios, making it well-suited for training and benchmarking complex surgical tasks with improved sim-to-real transfer capabilities. In this work, we leverage ORBIT-Surgical to study dual-arm coordination for the needle handover task, a key component in suturing.

We develop a reinforcement learning framework using Proximal Policy Optimization (PPO) to train low-level control policies. Our approach incorporates structured reward shaping and efficient exploration strategies to enable precise and autonomous handovers between robotic arms, advancing the pursuit of surgical augmented dexterity through scalable, data-driven learning.

II. Related Works

2.1 Single-Stage Reinforcement Learning Tasks

Prior applications of reinforcement learning in surgical robotics have predominantly focused on short-horizon, single-arm tasks such as reaching and grasping [10-12]. These tasks are typically framed within limited environments that do not reflect the multi-stage nature of real-world surgical workflows. While these efforts demonstrate initial feasibility, they fall short of addressing more complex and temporally extended procedures such as suturing or dual-arm coordination.

2.2 Limitations of Current Approaches

Conventional RL algorithms face key limitations in surgical automation due to sparse reward signals, long-horizon credit assignment problems, and high-dimensional continuous action spaces [13, 14]. These challenges are exacerbated in multi-stage settings, where naive exploration can be prohibitively inefficient. As a workaround, many works resort to manual reward shaping [15], which requires expert engineering of task-specific reward functions. However, this introduces inductive biases, risks local optima, and often limits generalization to new settings. In addition, most learning frameworks neglect dual-arm robotic control, limiting the scalability of existing systems to tasks that require only a single manipulator. The omission of coordinated bimanual skills is a significant gap, given that real surgical procedures frequently involve complex interactions between both patient-side manipulators (PSMs).

2.3 Imitation Learning and Demonstration-Based Approaches

Imitation Learning (IL), especially Behavioral Cloning (BC), has been a widely used paradigm in surgical task automation, leveraging teleoperated demonstrations to train policies for suturing or debridement. While BC enables fast learning from limited data, it suffers from distribution shift, where minor execution errors can compound due to a lack of corrective feedback. More advanced IL techniques involve adversarial learning or reward inference, but these require substantial quantities of expert demonstrations and often suffer from poor training stability. Hierarchical frameworks have recently been proposed to mitigate these issues by decomposing long-horizon tasks into modular subtasks. This approach improves training efficiency, promotes policy reuse, and aligns closely with the structured nature of surgical workflows. However, few studies have applied this paradigm to the problem of dual-arm coordination in needle handover, which we specifically address in our work.

III. Methodology

3.1 Environment Setup

We used the ORBIT-Surgical framework built on top of NVIDIA Isaac Sim to simulate the dual-arm needle handover task. The environment models two dVRK patient-side manipulators (PSMs), a needle object, and a surgical table. We had access to the rsl_rl training pipeline, allowing for parallel PPO-based training across hundreds of environments. The task selected was Isaac-Handover-Needle-Dual-PSM-v0, and our primary focus was on lifting and moving the needle using one of the arms.

3.2 Limitations and Constraints

Despite access to advanced simulation tools, the project was shaped by several critical constraints. The ORBIT-Surgical environment, while highly realistic, is computationally demanding—especially with dVRK-based dual-arm setups and parallelized PPO training. Due to limited GPU availability, we were restricted in how many environments we could run efficiently, which affected training speed and experimentation cycles.

The environment could only be run in headless mode inside a remote Docker container, with no GUI support, making real-time debugging impossible. As a result, reference or goal poses could not be visualized or used, ruling out any form of target-based supervision. All reward modelling had to be built from scratch using raw scene data, requiring trial-and-error to approximate behavior that would typically be guided by pose errors. This made reward design more time-consuming and brittle. Moreover, the setup of the environment itself was completed late in the timeline, leaving limited time to scale beyond basic lift-and-move behaviors. Evaluating agent behavior required long training sessions and offline inspection of saved video files, slowing down iteration further.

These factors constrained our ability to implement and train the full handover policy in the available time.

3.3 RL Formulation

1. Task Overview

To address the computational difficulties of a robotic needle handover task, we segment the process into distinct subtasks: approaching, grasping, and lifting the needle. This segmentation follows natural transition points observed during expert demonstrations, facilitating clear task definition and policy learning.

Approaching: The robot end-effector must navigate to the needle's vicinity. Success requires positioning the end-effector within 5mm of the needle with orientation error less than 15 degrees.

Grasping: Starting near the needle, the gripper must precisely align with a designated grasping point. Success requires jaw-to-grasp-point distance within 2mm, orientation alignment within 10 degrees, and positive contact detection between gripper and needle.

Lifting: With the needle grasped, the robot must elevate it to a minimum height of 6cm above the workspace and hold stably for at least 1 second. Success requires maintaining this elevation without excessive needle rotation or dropping.

This modular approach enhances training efficiency and system robustness. Each subtask can be independently optimized for different needle types or environmental conditions. By chaining these specialized policies together, the system performs the complete handover procedure effectively while managing the computational complexity inherent in high-dimensional manipulation tasks.

2. Observation Space

The observation vector $\mathbf{o}_t \in \mathbb{R}^{14}$ at time t comprises the following components:

- Relative position between needle and end-effector: $\mathbf{p}_{\text{needle}} - \mathbf{p}_{\text{ee}} \in \mathbb{R}^3$
- Needle velocity (linear and angular): $\mathbf{v}_{\text{needle}} = [\mathbf{v}_{\text{lin}}, \boldsymbol{\omega}] \in \mathbb{R}^6$
- Orientation alignment between gripper X-axis and needle Z-axis: $\cos(\theta_{\text{ee-x,needle-z}}) \in \mathbb{R}^1$
- Distance from end-effector to needle tip and tail: $[||\mathbf{p}_{\text{ee}} - \mathbf{p}_{\text{tip}}||, ||\mathbf{p}_{\text{ee}} - \mathbf{p}_{\text{tail}}||] \in \mathbb{R}^2$
- Needle height from table: $z_{\text{needle}} \in \mathbb{R}^1$
- Gripper joint position: $j_{\text{gripper}} \in \mathbb{R}^1$

Thus the overall observation space is,

$$\mathbf{o}_t = \left[\mathbf{p}_{\text{rel}}, \mathbf{v}_{\text{needle}}, \cos(\theta), \mathbf{d}_{\text{tip/tail}}, z_{\text{needle}}, j_{\text{gripper}}\right] \in \mathbb{R}^{14}$$

3. State Space

The full system state $\mathbf{s}_t \in \mathbb{R}^{d_s}$ includes simulator-specific internal variables that are not directly observable to the policy. These include:

Full robot joint positions and velocities

- 6-DoF pose and velocities of the needle

- End-effector kinematic pose
- Internal task flags (e.g., holding state, contact state)

Only a subset $\mathbf{o}_t \subset \mathbf{s}_t$ is exposed to the policy.

4. Action Space

The agent outputs an action vector $\mathbf{a}_t \in \mathbb{R}^7$ at each timestep:

$$\mathbf{a}_t = [\Delta x, \Delta y, \Delta z, \Delta\theta_x, \Delta\theta_y, \Delta\theta_z, a_{\text{grip}}]$$

This represents,

$\Delta\mathbf{p} \in \mathbb{R}^3$: Cartesian displacement of end-effector position

$\Delta\boldsymbol{\theta} \in \mathbb{R}^3$: End-effector orientation delta (e.g., SO(3) parameterization)

$a_{\text{grip}} \in \mathbb{R}$: Scalar gripper open/close command

5. Reward Function

As we progressed through the training process, we transitioned from dense to sparse reward structures. Initially, a dense reward scheme was employed to facilitate baseline evaluations. This included a combination of positive rewards for desirable behaviors and penalties for undesirable actions. While performance metrics and visual inspections (via recorded videos) indicated that the robotic arm was effectively learning to reach and lift the needle, we observed certain anomalies. For instance, after introducing a penalty to discourage repetitive gripper open-close actions, the agent circumvented this by avoiding gripper use altogether, instead balancing the needle through arm motion alone. Similarly, when a penalty was applied to the gripper joint movement, the arm adapted by lifting objects without actuating the gripper joint (similar to picking up food without bending the wrist). These behaviours highlighted the agent's tendency to exploit dense reward signals in unintended ways. To mitigate such issues and better align the learned policy with the intended task objectives, we adopted a sparser reward structure. Our revised scheme entailed rewards for key milestones - reaching, holding, and lifting - along with a penalty based on the distance from the receiving arm, as the ultimate goal was to perform a handover by moving the object toward the other arm.

Sparse Reward $r_t^{\text{sparse}} \in \{0, 1\}$

Sparse rewards are binary (0 or 1) and triggered upon completion of key subgoals:

1. Needle Lift with Grip

$$r_t^{\text{sparse}} = \begin{cases} 1, & \text{if needle is lifted above } h_{\min} \text{ and held for } \tau \text{ seconds} \\ 0, & \text{otherwise} \end{cases}$$

2. Grasp Success (time-based)

$$r_t = \min\left(\frac{t_{\text{grasp}}}{T}, 1.0\right)$$

3. Needle Stability

$r_t = \exp\left(-\|\boldsymbol{\omega}_{\text{needle}}\|_2\right)$ where, $\boldsymbol{\omega}_{\text{needle}}$ is the angular velocity.

Dense Reward $r_t^{\text{dense}} \in \mathbb{R}$

The dense reward at time step t, denoted as $r_t^{\text{dense}} \in [-\infty, 1]$, is defined as the weighted sum of multiple shaped reward components:

$$r_t^{\text{dense}} = \sum_i w_i \cdot \phi_i(s_t, a_t)$$

where each ϕ_i corresponds to a shaped reward term and $w_i \in \mathbb{R}$ is its associated weight. The shaping terms are defined as follows:

- $\phi_1 = 1 - \tanh\left(\frac{\|\mathbf{p}_{\text{ee}} - \mathbf{p}_{\text{needle}}\|}{\sigma}\right) \quad \text{(approach reward)}$
- $\phi_2 = \text{sigmoid}(h_{\text{needle}} - h_{\text{min}}) \quad \text{(lifting reward)}$
- $\phi_3 = \nVdash[\text{holding}] \quad \text{(binary hold bonus)}$
- $\phi_4 = -\nVdash[\text{drop}] \quad \text{(drop penalty)}$
- $\phi_5 = \cos\left(\angle(\mathbf{x}_{\text{ee}}, \mathbf{z}_{\text{needle}})\right) \quad \text{(orientation alignment)}$
- $\phi_6 = e^{-\|\boldsymbol{\omega}_{\text{needle}}\|} \quad \text{(angular stability)}$
- $\phi_7 = -\|\dot{\mathbf{q}}\|^2 \quad \text{(joint velocity smoothness)}$
- $\phi_8 = \nVdash[\text{grasp success}] \quad \text{(task success bonus)}$
- $\phi_9 = \nVdash[\text{correct grip}] \quad \text{(tip-based grasp bonus)}$
- $\phi_{10} = -\|a_t - a_{t-1}\|^2 \quad \text{(action smoothness penalty)}$
- $\phi_{11} = -\nVdash[\text{finger toggle}] \quad \text{(gripper spamming penalty)}$

The reward structure is designed to guide exploration through early shaping rewards and encourage successful grasping and handover completion.

6. Curriculum Shaping

To facilitate learning in this long-horizon dexterous task, we implemented a curriculum shaping strategy that gradually increases task difficulty while modulating the reward structure. For this, an action rate penalty was introduced only after *n* timesteps, allowing early-stage exploration while progressively encouraging smoother control trajectories as the policy matured. This strategy enables the agent to bootstrap on easier sub-skills (e.g., reaching, gripping, lifting, and orienting) before mastering the complete handover task under high variability and sparse success conditions. Action rate penalty after n number of time steps

7. Termination

Episodes terminate under the following conditions:

- **Timeout:** After a predefined number of timesteps
- **Success:** When the needle is lifted above a threshold height (0.08m) and held stably for a required duration (200 steps)
- **Failure:** When the needle falls below a minimum height (-0.05m)

8. Algorithms for Multi-Stage Handover Task

In complex robotic control tasks, such as our challenge of training a robotic arm to reach, grasp, and manipulate objects, policy gradient methods have shown considerable success. One such method — Proximal Policy Optimization (PPO), introduced by Schulman et al. (2017), is an on-policy algorithm that was designed to work in both discrete and continuous action spaces. The main objective of PPO is to improve training stability and sample efficiency by avoiding large, destabilizing policy updates while still allowing for sufficient exploration. It achieves this by optimizing a surrogate objective function that includes a clipping mechanism, which constrains the policy update to stay within a ‘proximal’ region. This balance between exploration and stability makes PPO particularly effective for high-dimensional, continuous control problems (like ours).

PPO belongs to the family of policy gradient methods, which optimize the parameters of a stochastic policy,

$\pi_\theta(a|s)$, directly by estimating gradients of expected reward. Traditional policy gradient methods suffer from high variance and instability due to unbounded policy updates. PPO addresses this by constraining the policy update, preventing the new policy from diverging too far from the current one.

In our implementation, we use the PPO-Clip variant, which modifies the objective function by clipping the probability ratio between the new and old policy:

$$L(s,a,\theta_k,\theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a), \quad \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}, 1-\epsilon, 1+\epsilon\right) A^{\pi_{\theta_k}}(s,a)\right)$$

- θ_k and θ are the parameters of the old and new policies, respectively.
- $A^{\pi_{\theta_k}}(s,a)$ is the advantage estimate, which measures how much better an action a is than the average action at state s.
- The ratio $\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}$ represents how much the new policy changes the likelihood of taking action compared to the old policy.
- ϵ is a small positive hyperparameter (e.g., 0.1–0.3), controlling how far the policy is allowed to deviate.

The clipping mechanism serves as a form of regularization, so if the advantage is positive, increasing the action's probability improves the objective. Similarly, if the advantage is negative, decreasing the action's probability helps. This prevents the policy from making overly aggressive updates, which might cause performance collapse, leading to more stable learning and less manual hyperparameter tuning. PPO-Clip updates the policy parameters by maximizing the expected clipped objective:

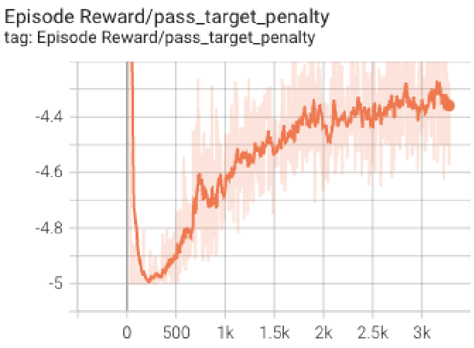
$$\theta_{k+1} = \arg\max_\theta \ \mathbb{E}_{(s,a)\sim\pi_{\theta_k}}\left[L(s,a,\theta_k,\theta)\right]$$

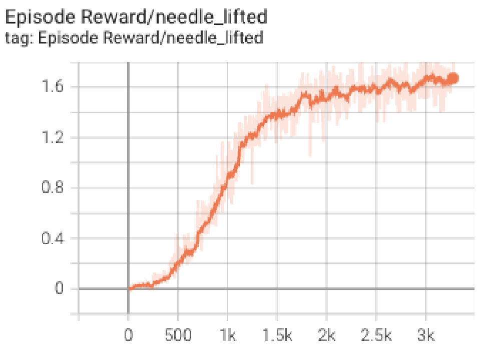
where the expectation is taken over states and actions sampled from the current policy π_{θ_k}, ensuring updates are both stable and grounded in the agent's actual experience.

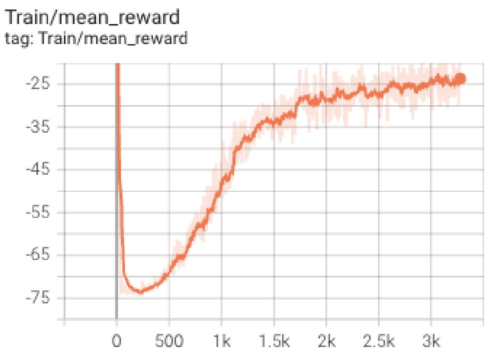
IV. Results and Discussion

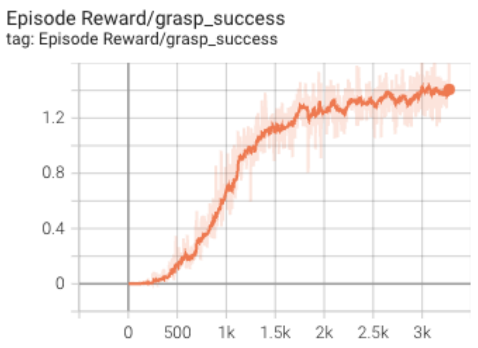
We began training with dense rewards to guide the agent more explicitly during early learning, as no predefined goals or reference poses were available. These dense terms provided shaping for grasping, lifting, and partial movement. Once stable behaviors emerged, we transitioned to a sparser reward formulation focused only on key outcomes, such as grasp success, lift height, and distance to the handover target. Despite the reduced supervision, the agent achieved comparable performance, demonstrating its ability to learn the task structure even under minimal reward feedback.

1. Rewards

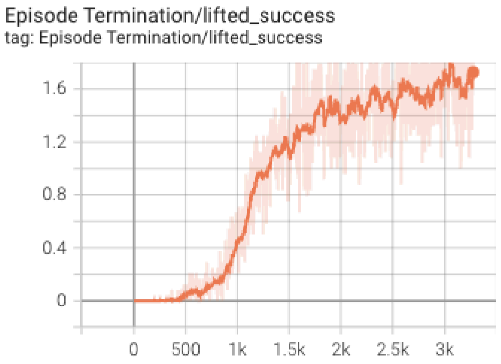

Episode Reward/pass_target_penalty
tag: Episode Reward/pass_target_penalty

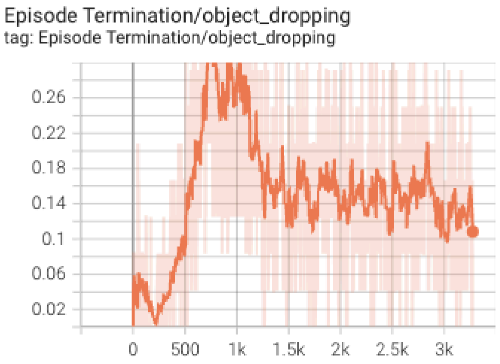

Episode Reward/needle_lifted
tag: Episode Reward/needle_lifted

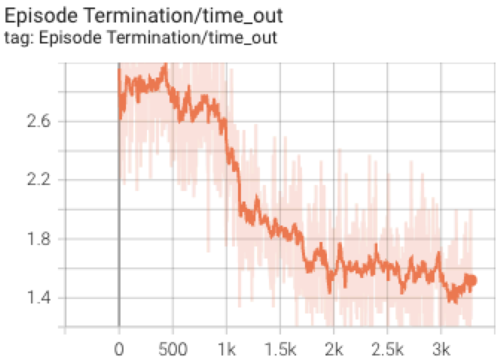




The agent demonstrates consistent improvement across key sub-rewards. Grasp success and needle lifting rewards show steady, monotonic increases, indicating reliable gripping and elevation behavior. Mean episode reward also improves significantly, reflecting overall task mastery. The pass target penalty decreases (becomes less negative), suggesting progress toward positioning the needle near the desired location.



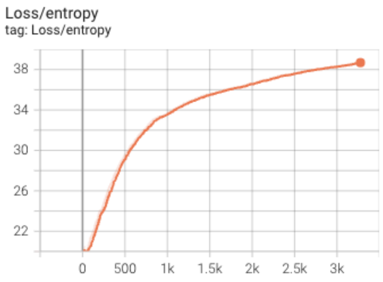


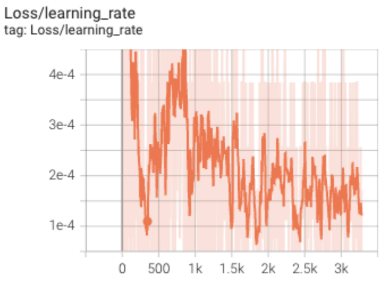


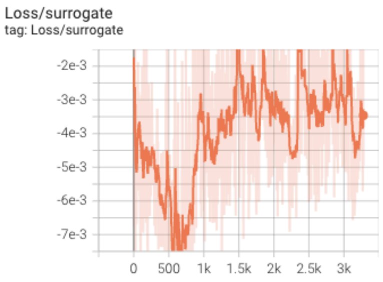
2. Termination

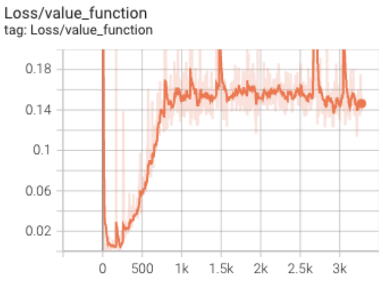
Termination trends indicate that the agent increasingly meets the lift success criterion over time, aligning with grasping and lifting improvements. Object dropping initially spikes as the agent starts lifting attempts, but stabilizes and slightly declines, suggesting better grip maintenance. Timeout terminations steadily decrease, reflecting more efficient completion of the task within the episode limit.

3. PPO Algorithm









The loss curves show stable policy learning. Entropy steadily increases, suggesting the policy remains exploratory. The surrogate loss fluctuates mildly around convergence, while the value function loss stabilizes after an initial rise, indicating the critic is learning a consistent value estimate. The learning rate adapts dynamically but remains within reasonable bounds, supporting stable optimization.

V. Conclusion

In this project, we explored the application of Proximal Policy Optimization (PPO) to train a robotic arm for needle manipulation tasks. By breaking down the complex process into subtasks of approaching, grasping, and lifting, we were able to manage the challenges of continuous control. Our transition from dense to sparse rewards helped the agent learn with less direct supervision, while PPO's stability prevented destructive policy updates during training. Despite facing computational constraints and limited time, we achieved promising results in teaching the arm to successfully grasp and lift the needle. While our implementation focused on basic manipulation rather than complete handover, there remain many opportunities to improve and expand upon these initial results by exploring more reward modelling.

VI. References

[1] *Sen, S., Garg, A., Gealy, D. V., McKinley, S., Jen, Y., & Goldberg, K. (2016).* ***Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization.*** *In 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4178–4185). IEEE. https://doi.org/10.1109/ICRA.2016.7487611*

[2] *Ginesi, M., Meli, D., Nakawala, H., Roberti, A., & Fiorini, P. (2019).* ***A knowledge-based framework for task automation in surgery.*** *In 2019 19th International Conference on Advanced Robotics (ICAR) (pp. 37–42). IEEE. https://doi.org/10.1109/ICAR46387.2019.8981619*

[3] *Hari, K., et al. (2024).* ***STITCH: Augmented dexterity for suture throws including thread coordination and handoffs.*** *In 2024 International Symposium on Medical Robotics (ISMR) (pp. 1–7). IEEE. https://doi.org/10.1109/ISMR63436.2024.10585751*

[4] *Kumar, V., Shah, R., Zhou, G., Moens, V., Caggiano, V., Vakil, J., Gupta, A., & Rajeswaran, A. (2023).* ***RoboHive: A unified framework for robot learning****. In Proceedings of the 37th International Conference on Neural Information Processing Systems (Article No. 1918, pp. 1–18). Curran Associates Inc. https://sites.google.com/view/robohive*

[5] *Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., & Levine, S. (2020, May).* ***Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning.*** *In Conference on Robot Learning (pp. 1094–1100). PMLR.*

[6] *James, S., Ma, Z., Arrojo, D. R., & Davison, A. J. (2020).* ***RLBench: The robot learning benchmark & learning environment.*** *IEEE Robotics and Automation Letters, 5(2), 3019–3026. https://doi.org/10.1109/LRA.2020.2977217*

[7] *Xu, J., Li, B., Lu, B., Liu, Y. H., Dou, Q., & Heng, P. A. (2021, September).* ***Surrol: An open-source reinforcement learning centered and dvrk compatible platform for surgical robot learning.*** *In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1821-1828). IEEE.*

[8] *Schmidgall, S., Krieger, A., & Eshraghian, J. (2024, May).* ***Surgical Gym: A high-performance GPU-based platform for reinforcement learning with surgical robots.*** *In 2024 IEEE International Conference on Robotics and Automation (ICRA) (pp. 13354-13361). IEEE.*

[9] *Yu, Q., Moghani, M., Dharmarajan, K., Schorp, V., Panitch, W. C.-H., Liu, J., Hari, K., Huang, H., Mittal, M., Goldberg, K., & Garg, A. (2024).* ***ORBIT-Surgical: An open-simulation framework for learning surgical augmented dexterity.*** *arXiv preprint arXiv:2404.16027. https://arxiv.org/abs/2404.16027*

[10] *Schorp, V. et al. S****elf-supervised learning for interactive perception of surgical thread for autonomous suture tail-shortening.*** *In IEEE Intl. Conf. on Automation Science and Engineering (CASE), 1–6 (IEEE, 2023).*

[11] *Amirshirzad, N., Sunal, B., Bebek, O. & Oztop, E.* ***Learning medical suturing primitives for autonomous suturing.*** *In IEEE Intl. Conf. on Automation Science and Engineering (CASE), 256–261 (IEEE, 2021).*

[12] *Zhou, H. et al.* ***Suturing tasks automation based on skills learned from demonstrations: A simulation study.*** *In IEEE Intl. Symp. on Medical Robotics (ISMR) (IEEE, 2024).*

[13] *Zhu, R., Li, S., Dai, T., Zhang, C. & Celiktutan, O.* ***Learning to solve tasks with exploring prior behaviours.*** *In IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 7501–7507 (IEEE, 2023).*

[14] *Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W. & Abbeel, P.* ***Overcoming exploration in reinforcement learning with demonstrations.*** *In IEEE Intl. Conf. on Robotics and Automation (ICRA), 6292–6299 (IEEE, 2018).*

[15] *Pore, A. et al.* ***Learning from demonstrations for autonomous soft-tissue retraction.*** *In IEEE Intl. Symp. on Medical Robotics (ISMR), 1–7 (IEEE, 2021).*