Nail It! A learning framework for autonomous surgical suturing and teleoperation on the dVRK

International Journal of Computer Assisted Radiology and Surgery
https://doi.org/10.1007/s11548-026-03735-8

REVIEW ARTICLE

Nail It! A learning framework for autonomous surgical suturing and teleoperation on the dVRK

Luigi Muratore[1,2] · Matteo Pescio[2,4] · Federica Barontini[2] · Francesco Marzola[2] · Pietro Leoncini[2,4] · Carlo Alberto Ammirati[2] · Alberto Arezzo[2] · Giulio Dagnino[2,3] · Giuseppe Averta[5]

Received: 9 January 2026 / Accepted: 27 May 2026

Abstract

Purpose Reliable automation of surgical suturing requires accurate and flexible simulation tools to effectively learn robot autonomous skills that match clinical practice. Yet, high-fidelity, ready-to-use learning frameworks remain scarce even for popular platforms like the da Vinci Research Kit (dVRK). We release Nail It!, a complete Unity–ROS–dVRK learning framework supporting both teleoperation data collection and Reinforcement Learning (RL) training of surgical primitives like needle grasping and placement.

Methods Nail It! features (i) a physics-based Unity environment with full and accurate dVRK kinematic modelling, (ii) a real-time ROS communication with the dVRK master console for active surgeon control of Patient-Side Manipulators (PSMs), and (iii) an integrated Graphical User Interface (GUI) for environment tuning, reward design, learning algorithms development, and sim-to-real validation. We also provide full support for RL policies training, e.g. through Proximal Policy Optimization (PPO) with curriculum learning and domain randomization for robust policy training of suturing steps, and Multi Agent RL for multi-arm coordination.

Results We experiment with Nail It! to learn autonomous suturing with our dVRK using PPO with curriculum learning to handle multi-step surgical procedures, and domain randomization techniques to mitigate the sim-to-real gap and enable policy deployment. Nail It! delivers policies that can solve the suturing task in simulation with 95% accuracy, with proven robustness over visual and positional variations up to 15 mm (target).

Conclusions Nail It! provides a modular, high-fidelity platform for developing and benchmarking autonomous surgical skills. By integrating accurate kinematics, reinforcement learning, and teleoperation, it supports both autonomous training and human-in-the-loop control in realistic surgical scenarios. Direct ROS connectivity and an integrated GUI further facilitate rapid development and sim-to-real experimentation. Supplementary materials associated with this work are publicly available at https://luigimuratore.github.io/Nail-it/.

Keywords Surgical robotics · Reinforcement learning · Da Vinci Research Kit (dVRK) · Autonomous suturing · Policy optimization · Multi-agent RL

State of the art and open challenges

Robotic-assisted surgery has significantly advanced minimally invasive procedures by improving dexterity, precision, and ergonomics compared to conventional laparoscopy. Among surgical subtasks, suturing remains one of the most fundamental challenging operations, requiring precise needle handling, bimanual coordination, and consistent execution over long periods [1]. Being a repetitive and time-consuming activity, suturing is a prime candidate for automation, with the potential to reduce surgeon workload, standardize outcomes, and improve procedural efficiency in robotic-assisted surgery [2].

Despite sustained research interest, autonomous surgical suturing remains an open problem. Early approaches focused on classical control and motion planning techniques, often relying on pre-defined trajectories or simplified task assumptions [3]. More recent works have explored learning-based methods, including Imitation Learning (IL) and Reinforcement Learning (RL), to cope with the variability and complexity inherent in surgical manipulation [4–6]. These approaches have demonstrated promising results in subtasks such as needle grasping, positioning, and simple insertion motions.

However, fully autonomous suturing requires coordinated multi-step manipulation, precise handling of tools, and

Luigi Muratore and Matteo Pescio have contributed equally to this work.

Extended author information available on the last page of the article

Published online: 02 July 2026

robustness to geometric and visual shifts. The problem is further exacerbated for systems such as the da Vinci Research Kit (dVRK), where even and accurate modelling of the robot's kinematics and joint constraints—essential for transferring learned behaviours to real hardware [7]–still lacks. As a result, many existing approaches restrict their scope to simplified tasks, single manipulators, or heavily constrained environments, limiting their applicability to realistic surgical scenarios. A growing body of work highlights that progress in autonomous suturing is increasingly bottlenecked not by learning algorithms alone, but by the quality and fidelity of the simulation environments used for training and validation [8].

Unified framework

Accurate and reliable simulation plays a crucial role for the development of autonomous surgical systems, enabling safe policy learning, large-scale data generation, and systematic benchmarking before deployment on real robots. Several simulation frameworks have been proposed to support research in surgical robotics and learning-based control, particularly in conjunction with the dVRK platform [9–17].

While these simulators provide valuable capabilities, such as rigid-body dynamics, simplified soft tissue interaction, or task-specific environments, they share important limitations when considered in the context of autonomous suturing. Most notably, existing frameworks often rely on approximate or incomplete kinematic models of the dVRK, without explicitly accounting for joint-level behaviour and passive mechanisms that are essential for precise needle manipulation. This limits the physical consistency of the simulated motions and dramatically increases the sim-to-real gap when transferring learned policies to hardware. In addition, many simulators lack tight integration with ROS-based control architectures and the physical dVRK master console, preventing seamless transitions between autonomous execution, teleoperation, and human-in-the-loop data collection.

RL pipelines are therefore frequently implemented in an ad hoc or task-specific manner, with limited support for advanced learning paradigms like curriculum learning, domain randomization, or systematic reward-function design, elements that are essential for stable training in complex manipulation tasks such as suturing.

Despite suturing being inherently a bimanual and cooperative task, current simulation platforms rarely support Multi-Agent Reinforcement Learning (MARL) paradigms to allow coordinated training of multiple Patient-Side Manipulators (PSMs) [18, 19].

These limitations highlight the lack of a unified, high-fidelity simulation environment that simultaneously provides:

- Accurate dVRK kinematics at the joint level;
- Flexible and scalable RL-bases training;
- Real-time bidirectional ROS communication;
- Support for both autonomous control and teleoperation within the same framework.

To bridge this gap, we release Nail It!, a Unity–ROS simulation framework specifically designed for learning-based autonomous surgical suturing with the dVRK. By combining accurate kinematic reconstruction, real-time ROS connectivity, and an integrated GUI, with a RL-based learning infrastructure potentially supporting multiple learning methods (e.g. PPO, SAC) with or without additional advanced learning strategies to increase policy robustness (domain randomization and curriculum learning), Nail It! enables reliable training, evaluation, and sim-to-real validation of autonomous needle manipulation and suturing-related tasks.

Materials and methods

System architecture and simulation environment

The proposed framework integrates the Unity real-time 3D engine with ROS to enable synchronized simulation, teleoperation, and autonomous control of the dVRK. The overall architecture follows a modular design (Fig. 1a), which separates simulation, middleware, and learning functionalities while maintaining tight real-time coupling.

At the simulation level, Unity is responsible for kinematic modelling, physics simulation, and visual rendering of the surgical scene. The environment simulates realistic interactions between instruments and task objects while maintaining deterministic physics to ensure reproducibility across learning experiments.

The middleware layer ensures bidirectional communication between Unity and the dVRK software stack, enabling real-time exchange of joint states, end-effector poses, and control commands.

The learning layer handles RL agents, data logging, and policy evaluation, and interfaces with ROS to access simulated and teleoperated data streams. This enables both autonomous training in simulation and the collection of joint-level and end-effector trajectories during teleoperation for IL and offline analysis.

This modular structure supports extensibility and reproducibility, allowing new task environments, sensors, or learning algorithms to be integrated with minimal reconfiguration.

A complete surgical scene replicates the workspace and spatial constraints of a standard dVRK setup (Fig. 1b). The environment includes two Patient Side Manipulators (PSM1, PSM2), the endoscopic camera, and the passive support

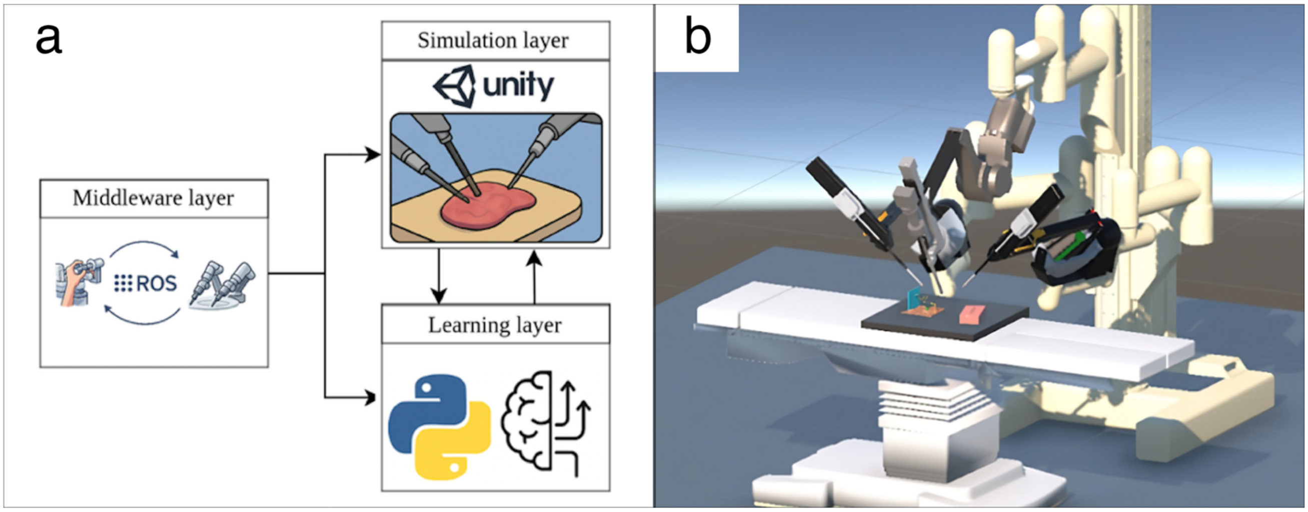

Fig. 1 Overview of the proposed Unity–ROS–dVRK framework: **a** Layered design architecture scheme, **b** Complete surgical scene in Unity

assembly, all modelled at a 1:1 scale to preserve the real platform's geometry and reachability.

A surgical bed and laparoscopic camera are placed to replicate anatomical entry points, line-of-sight constraints, and occlusions of a real dVRK setup. A surgical phantom [20] and auxiliary assets [21] are integrated into the scene, which is designed to match the spatial layout, kinematic limits, and visibility conditions of the surgical cart, providing a realistic and modular environment for teleoperation and reinforcement learning.

Kinematic modelling and ROS integration

A full kinematic reconstruction of the dVRK PSM was implemented in Unity including both active and passive kinematic elements [22, 23].

The modelling process moves from a complete Denavit–Hartenberg (DH) formulation of the PSM, defining link frames, joint axes, and geometric offsets (Fig. 2). These parameters served as the analytical backbone for rebuilding the manipulator in Unity, guiding the placement and orientation of each virtual link and preserving the spatial relationships of the physical robot.

Passive joints and calibrated coupling factors are introduced to reproduce physical interdependencies and mechanical constraints of the real dVRK mechanism, enabling realistic tool motion despite the absence of direct actuation. The resulting kinematic model yields a mechanically consistent simulation suitable for teleoperation and RL.

The reconstructed PSM kinematics are validated through representative joint-space motions with progressively enabled compensation constraints (Fig. 3), confirming that passive joints and coupling terms restore realistic joint coordination and end-effector trajectories. This level of kinematic fidelity is critical for learning-based control, where small modelling errors can destabilize policies or hinder sim-to-real transfer.

The communication scripts were adapted from the open-source STEVE framework[1] and extended to support dVRK-specific joint structures, message types, and update frequencies. This enables full control of the simulated PSMs through the physical dVRK master console while preserving compatibility with standard ROS topics, transforms, and controller nodes.

Unity subscribes to master-console input topics and publishes simulated PSM states back to ROS, forming a closed feedback loop that supports real-time teleoperation with visual feedback from Unity's 3D scene.

The same interface enables synchronized data logging, allowing human demonstrations collected during teleoperation to be directly reused for IL or hybrid training pipelines.

Reinforcement learning framework and task design

Reinforcement learning is implemented using PPO, chosen for its stability in continuous control, while SAC is evaluated for comparison under identical task and observation settings. Learning is interfaced through Unity ML-Agents.

Observations comprise joint states, tooltip poses relative to targets, and binary contact information (20 features), with a seven-dimensional continuous joint action space per PSM.

Hyperparameters were tuned empirically, and the final configuration is reported in Table 1 of the Supplementary Material. All experiments were repeated across three random seeds. CL is employed to progressively increase task

[1] STEVE framework repository: https://github.com/alberto-rota/STEVE.

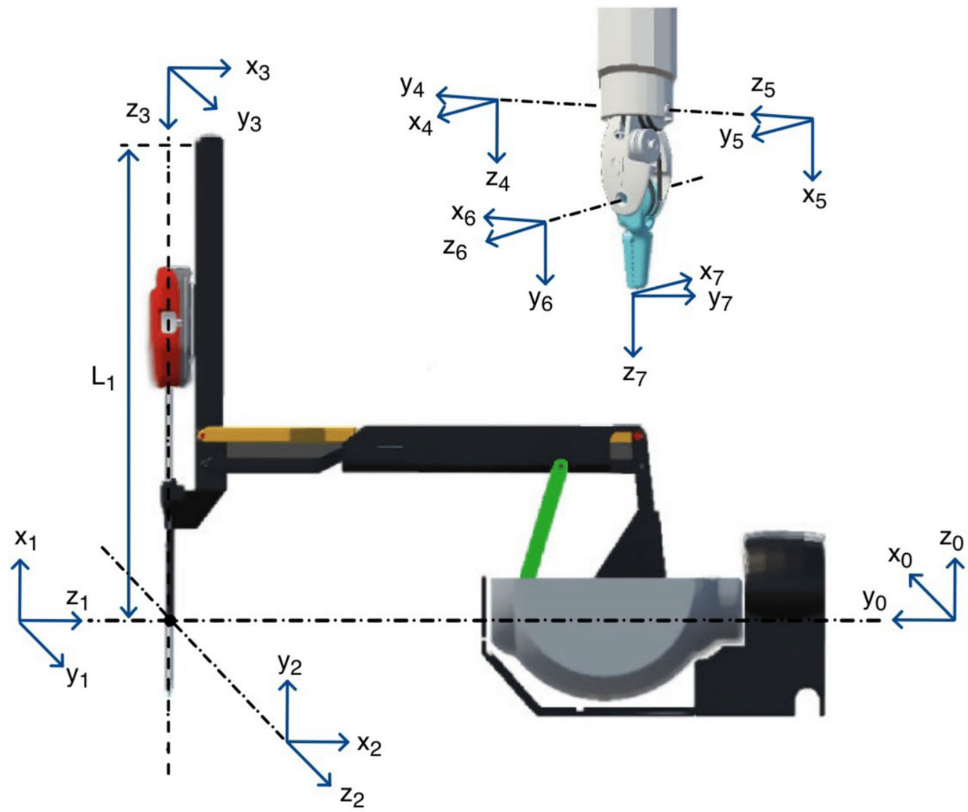

Fig. 2 Denavit–Hartenberg parameters of the PSM

Frame	Joint Name	a	α	d	θ
1	Arm Jaw	0	$\pi/2$	0	$q_1 + \pi/2$
2	Arm Pitch	0	$-\pi/2$	0	$q_2 + \pi/2$
3	Arm Insertion	0	$\pi/2$	$q_3 - L_1$	0
4	Tool Roll	0	0	L_{tool}	q_4
5	Tool Pitch	0	$-\pi/2$	0	$q_5 + \pi/2$
6	Tool Yaw	0	$-\pi/2$	0	$q_6 + \pi/2$
7	End-Effector	-	$-\pi/2$	-	0

difficulty, while DR improves robustness by varying object geometry and spatial parameters during training (CL details are reported in Table 2 of the Supplementary Material).

Four tasks were defined to evaluate both single-agent and multi-agent control scenarios (Fig. 4):

1. Reaching—The agent moves the end-effector from a randomized initial configuration to pick up the needle.
2. Placement—The agent positions and stabilizes the needle on a designated target plane.
3. Complete manipulation—A sequential episode combining reaching, grasping, and placement.
4. Multi-Agent Task—Two PSMs operate concurrently to perform coordinated needle manipulation and hand-off.

For the reaching task, the agent is rewarded proportionally to the distance reduction between the end-effector and the needle, with larger incentives within a proximity band and a terminal success reward upon entering the grasping region.

The placement task combines height-based shaping with a lateral centring term to encourage precise alignment with the target plane, rewarding stable contact and penalizing loss of contact.

The complete task integrates these components sequentially, issuing a one-time reward upon successful grasping and a terminal reward upon correct placement, while penalizing needle drops.

Across all tasks, the reward function includes a small time penalty and a velocity penalty to promote smooth, efficient motion.

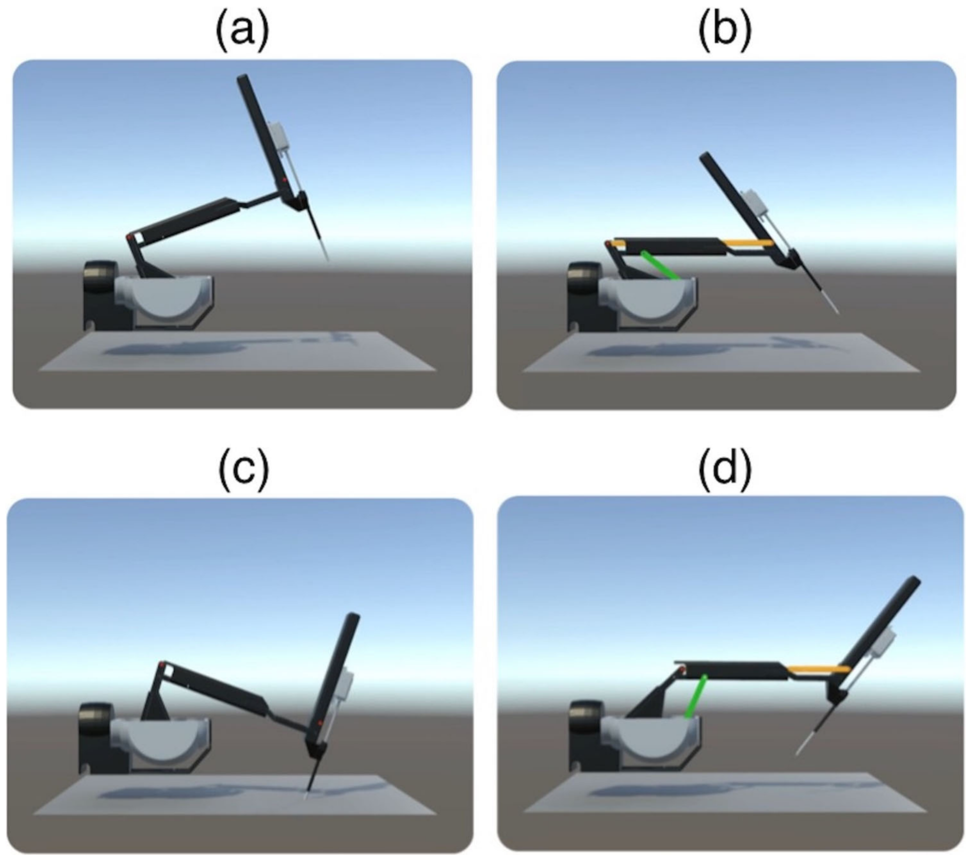

Fig. 3 Comparison of PSM arm pitch motion without and with kinematic constraints: (a, c) Motions executed without constraints, (b, d) Corresponding motions with constraints enabled. Panels **a**–**b** show counterclockwise rotations, while panels **c**–**d** show clockwise rotations

Table 1 Quantitative comparison across learning methods. For each task, success rate (SR) and mean final target error are reported. The error is the Euclidean distance between the needle tip and the target centre at the final timestep

Method	Reaching		Placement		Complete	
	SR (%)	Error (mm)	SR (%)	Error (mm)	SR (%)	Error (mm)
PPO	92	1.8	85	2.4	78	3.1
SAC	30	12.4	24	15.2	20	18.9
PPO + CL	96	1.4	91	2.0	85	2.7
PPO + DR	94	2.0	88	2.6	81	3.3
PPO + CL + DR	95	1.2	90	2.1	88	2.9

Detailed pseudocode descriptions of the reward functions for the reaching, placement, and complete manipulation tasks are provided in Algorithms 1–3 of the Supplementary Material.

The defined tasks are intentionally structured as manipulation primitives rather than complete suturing procedures. This decomposition reflects a staged approach to surgical autonomy, where reliable execution of low-level skills precedes the integration of higher-level behaviours. By isolating reaching, grasping, and placement, the framework enables targeted analysis of learning dynamics and failure modes, while providing reusable components for future composition into full suturing pipelines.

To extend beyond single-arm autonomy, a MARL architecture is introduced to coordinate the two PSMs. Each arm is governed by an independent policy network under a decentralized execution scheme, while centralized training allows agents to exploit shared environmental information.

The design follows the Centralized Training with Decentralized Execution (CTDE) paradigm, enabling coordinated bimanual behaviours such as hand-off motions and cooperative manipulation.

In the defined multi-agent task, PSM1 performs reaching, grasping, and transport of the needle to the entry location. A deterministic controller then executes a scripted insertion arc to emulate tissue penetration. Control subsequently transfers to PSM2, which learns to approach, align, and re-grasp the exposed needle tip to follow-up the manipulation.

The insertion motion is intentionally scripted, to decouple tissue interaction from coordination learning, allowing the

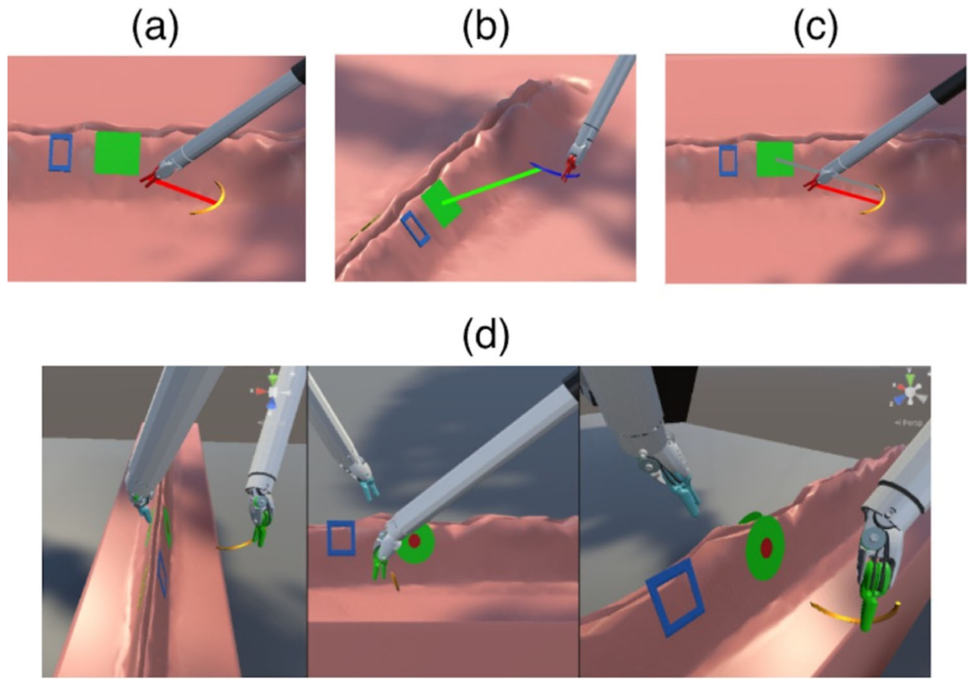

Fig. 4 Tasks design: **a** reaching, **b** placement, **c** complete, **d** multi-agent

agents to focus on perception, grasping, and hand-off strategies without introducing additional sources of uncertainty at this stage.

This architecture provides a scalable foundation for future extensions toward fully autonomous suturing involving synchronized dual-arm control.

All implementation details, including hardware specifications and software versions, are reported in Table 3 of the Supplementary Material.

A dedicated GUI was developed to streamline experimental configuration, teleoperation, and RL workflows. The GUI allows users to select surgical scenes, configure task parameters, adjust learning hyperparameters, enable curriculum learning and domain randomization, and monitor joint states, rewards, and policy convergence in real time.

The interface supports both teleoperation and autonomous training modes, enabling efficient testing, demonstration recording, and hybrid learning scenarios. By exposing key parameters without code-level modification, the GUI significantly accelerates reward design and algorithm development.

Results

The proposed framework was evaluated on both single-agent and multi-agent manipulation tasks to assess learning stability, convergence behaviour, task success, and coordination performance. All experiments were conducted using PPO with results aggregated over three independent random seeds.

Single-agent task performance

All single-agent tasks—reaching, placement, and complete manipulation—were successfully learned using PPO. Figure 5 reports the cumulative reward curves for each task, illustrating distinct learning dynamics corresponding to task complexity.

Across all single-agent tasks, PPO converged reliably for all random seeds, with learning curves reaching a stable steady state after training. Final policies achieved task success rates exceeding 90% across evaluation episodes, demonstrating consistent execution under the randomized conditions introduced during training.

In contrast SAC failed to achieve task completion across all evaluated scenarios, exhibiting high variance and unstable oscillatory behaviour in the learning curves and never reaching meaningful reward levels within the same training horizon. This behaviour is consistent with known limitations of off-policy methods such as SAC in long-horizon manipulation tasks with sparse terminal rewards and tight geometric tolerances. In the considered environment, sub-millimetre placement constraints, delayed success signals, and a high-dimensional continuous action space significantly increase the difficulty of credit assignment, favouring the on-policy, clipped-update structure of PPO.

A quantitative comparison across learning methods is reported in Table 1, summarizing task success rates and mean final target errors for all single-agent tasks.

Training runs incorporating CL converged consistently for all seeds. In contrast, training without CL frequently resulted

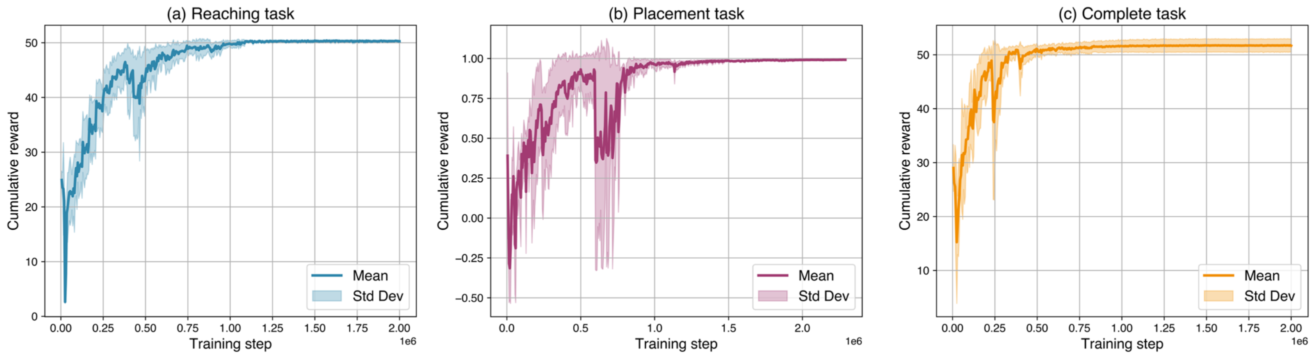

Fig. 5 Cumulative reward over time of the tasks: **a** reaching task, **b** placement task, **c** complete task

in unstable behaviour or failure to converge. Policies trained with domain randomization demonstrate strong performance across variations in object pose, plane geometry, and grasping conditions not seen at training time.

Quantitatively, the learned policies achieved high spatial accuracy despite substantial injected variability. Across evaluation episodes, the average centring error with respect to the target entry point was about 3.0 mm, even under positional randomization of up to 15 mm for the target plane and 5 mm for the needle pose. This result highlights the ability of the proposed method to learn precise control policies that generalize beyond nominal training configurations, supporting its use as a reference baseline for future algorithmic improvements.

Single tasks were completed within 100–150 simulation steps (approximately 1.7–2.5 s at 60 Hz) while complete manipulation task required more steps due to its sequential structure but still achieved consistent end-to-end execution across evaluation episodes.

Representative rollouts for all single-agent tasks are provided in the Supplementary Material (Videos S1–S9), including video demonstrations for three independent random seeds per task.

Analysing each task individually shows distinct learning behaviours that reflect increasing task complexity:

- Reaching task: exhibited rapid learning, with cumulative rewards increasing sharply during early training and stabilizing after approximately 1–1.25 million steps. After convergence, the agent consistently reached the needle from randomized initial configurations with low variance across episodes.
- Placement task: demonstrated smooth and monotonic reward growth throughout training. Convergence occurred after approximately 3–4 million steps, followed by a stable, low-variance reward plateau, indicating consistent placement accuracy across evaluation episodes.
- Complete manipulation task: presented the most complex learning dynamics. Reward curves exhibited larger variance during the initial 0–500 k steps, corresponding to exploration and the difficulty of discovering reliable sequential strategies. After convergence, policies achieved stable end-to-end execution, with occasional reward drops associated with exploration or imperfect grasps.

Overall, these results define a strong and reproducible baseline for learning surgical manipulation tasks using RL, against which future methods—such as alternative reward shaping, improved observation models, or hybrid learning strategies—can be quantitatively compared.

These results establish the proposed Nail It! training pipeline as a stable baseline for learning fine manipulation behaviours in surgical robotics simulation.

Multi-agent task performance

Multi-agent experiments evaluated coordinated manipulation between PSM1 and PSM2. PSM1 reliably executed reaching, grasping, and pre-insertion placement, consistently positioning the needle at the entry site. The scripted insertion arc behaved deterministically, providing a stable transition between agents.

PSM2 did not achieve reliable re-grasping of the needle tip following insertion. Although the policy occasionally moved the end-effector toward the correct spatial region, small pose deviations introduced during the insertion phase—combined with ambiguities in the observation space shared between agents—resulted in needle configurations that were not consistently represented during training. Consequently, PSM2 failed to precisely localize and re-grasp the needle in a stable manner.

Rather than representing a limitation of the Nail It framework itself, this outcome highlights an important open research direction. The current results serve as a baseline multi-agent benchmark, illustrating the sensitivity of coordinated surgical manipulation to inter-agent state estimation, observation design, and policy coupling. Future work may address these challenges through improved cross-agent perception, shared latent representations, explicit communication mechanisms, or hierarchical coordination strategies.

Qualitative multi-agent executions are illustrated in the Supplementary Material (Videos S10–S11).

Discussion

The results demonstrate that Nail It! provides a stable and effective environment for learning autonomous surgical manipulation skills under realistic kinematic and spatial constraints. Across all single-agent tasks, PPO consistently achieved reliable convergence, confirming its suitability for high-dimensional, tightly coupled robotic systems such as the dVRK. In contrast, the unstable behaviour observed with SAC highlights the sensitivity of surgical manipulation tasks to algorithmic choices, where smooth convergence, low variance, and predictable behaviour are critical for both safety and reproducibility.

A central outcome of this work is the validation of task decomposition into manipulation primitives as a viable strategy for autonomous suturing research. While the presented tasks do not yet constitute a complete suturing procedure, they capture essential subtasks—needle reaching, grasping, orientation, and placement—that form the foundation of suturing. Reliable execution of these primitives is a necessary prerequisite for higher-level behaviours such as tissue interaction, stitch tightening, and knot tying. Framing autonomy around validated primitives enables a modular development paradigm, where increasingly complex surgical behaviours can be composed from well-understood and robust low-level skills.

Curriculum learning emerged as a decisive factor in enabling precise manipulation. When exposed immediately to full task complexity, agents frequently failed to discover stable strategies, particularly for grasping and placement. Gradually increasing task difficulty allowed the agent to acquire coarse positioning behaviours first and refine them into precise, goal-directed motions. This progression mirrors human surgical training and supports the view that curriculum-based approaches are particularly well suited to surgical robotics, where precision requirements are high, and exploration must be carefully constrained.

Domain randomization further enhanced robustness by exposing policies to variability in object pose, workspace geometry, and grasping conditions. Policies trained under randomized conditions consistently maintained high success rates in previously unseen configurations, addressing one of the primary barriers to sim-to-real transfer. While real-world deployment will inevitably introduce additional sources of uncertainty, these results indicate that the proposed framework provides a solid foundation for bridging the sim-to-real gap in future experiments on physical dVRK systems.

The observed residual jerk during rapid reorientation or fine adjustments highlights an important limitation of learning-based control in rigid-body simulators. Although velocity penalties reduce extreme accelerations, agents may still exploit dynamic shortcuts that are suboptimal for surgical motion quality. This behaviour is expected in physics engines not explicitly designed for surgical smoothness. Ongoing work will explore jerk-aware reward terms, differentiable smoothing layers, and hybrid low-level controllers to further align learned trajectories with the motion characteristics required for safe and precise surgical execution.

In the multi-agent setting, reliable performance by PSM1 demonstrates that coordinated dual-arm manipulation is a promising direction within the proposed framework. At the same time, the failures observed during transfer to PSM2 reveal the significant sensitivity of multi-agent coordination to upstream variability, observation design, and intermediate task states. Small pose deviations accumulated during the insertion arc were sufficient to degrade downstream performance, underscoring the inherent difficulty of multi-stage bimanual manipulation. These findings motivate future work on improved curriculum strategies, expanded domain randomization over intermediate configurations, and explicit cross-agent feedback mechanisms.

Conclusions

This work presents Nail It!, a unified Unity–ROS–dVRK simulation framework for studying autonomous surgical manipulation through reinforcement learning and teleoperation. The framework integrates accurate kinematic modelling, learning-based control, and real-time interaction within a single modular architecture, enabling systematic investigation of surgical manipulation tasks under realistic constraints.

This work demonstrates the successful learning of needle reaching, placement, and complete manipulation tasks using reinforcement learning. Through CL and DR, stable convergence and robust performance were achieved across increasing task complexity, supporting a staged approach toward autonomous suturing in a physically consistent simulation environment.

The framework further provides a high-fidelity kinematic model of the dVRK PSM, capturing passive joints, couplings, and mechanical constraints to closely reproduce real system behaviour and improve sim-to-real consistency. In addition, support for real-time surgeon-in-the-loop teleoperation enables demonstration collection and analysis, positioning the simulator as a unified testbed for reinforcement, imitation, and hybrid learning approaches in surgical robotics.

The integrated GUI supports efficient experimentation by enabling configuration of tasks, learning parameters, curriculum schedules, and domain randomization without code-level modification. While not the central contribution of this work, the GUI facilitates rapid iteration, monitoring, and debugging of learning experiments, improving usability and reproducibility during algorithm development.

Taken together, these components form a complete and extensible simulation framework that supports autonomous learning, surgeon-in-the-loop teleoperation, and data collection within a unified environment. Future work will extend the system toward soft-tissue interaction, force-aware control, and deployment on physical dVRK hardware, enabling investigation of shared autonomy, bimanual coordination, and full suturing pipelines. These developments will further advance the framework toward safe, robust, and clinically relevant autonomous surgical systems.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11548-026-03735-8.

Acknowledgements This work was supported by the European Research Council (ERC) under the Horizon Europe programme (Endotheranostics, Grant Agreement No. 101118626, https://doi.org/10.3030/101118626), by the CLASSICA project (Grant Agreement No. 101057321, https://doi.org/10.3030/101057321) and the PALPABLE project (Grant Agreement No.101092518, https://doi.org/10.3030/101092518). The work has also been partially supported by the Italian Ministry of University and Research (MUR) under the PRIN 2022 programme "Towards Intelligent ROBOTIC ENDOSCOPIC DISSECTION (TI-RED)". Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union, the European Research Council Executive Agency, or the Health and Digital Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Funding Open access funding provided by Università degli Studi di Torino within the CRUI-CARE Agreement. HORIZON EUROPE European Research Council, 101118626, Alberto Arezzo, HORIZON EUROPE Framework Programme, 101057321, Alberto Arezzo, 101092518,Alberto Arezzo.

Declarations

Conflict of interest All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. Kenngott HG, Müller-Stich BP, Reiter MA, Rassweiler J, Gutt CN (2008) Robotic suturing: technique and benefit in advanced laparoscopic surgery. Minim Invasive Ther Allied Technol 17(3):160–167. https://doi.org/10.1080/13645700802103381
2. Leoncini P, Marzola F, Pescio M, Casadio M, Arezzo A, Dagnino G (2025) A reproducible framework for synthetic data generation and instance segmentation in robotic suturing. Int J Comput Assist Radiol Surg 20(8):1567–1576. https://doi.org/10.1007/s11548-025-03460-8
3. Yang G-Z et al (2017) Medical robotics—regulatory, ethical, and legal considerations for increasing levels of autonomy. Sci Robot 2(4):eaam8638. https://doi.org/10.1126/scirobotics.aam8638
4. Shahkoo AA, Abin AA (2023) Deep reinforcement learning in continuous action space for autonomous robotic surgery. Int J Comput Assist Radiol Surg 18(3):423–431. https://doi.org/10.1007/s11548-022-02789-8
5. Elguea-Aguinaco Í, Serrano-Muñoz A, Chrysostomou D, Inziarte-Hidalgo I, Bøgh S, Arana-Arexolaleiba N (2023) A review on reinforcement learning for contact-rich robotic manipulation tasks. Robot Comput Integr Manuf 81:102517. https://doi.org/10.1016/j.rcim.2022.102517
6. Padoy N (2019) Machine and deep learning for workflow recognition during surgery. Minim Invasive Ther Allied Technol 28(2):82–90. https://doi.org/10.1080/13645706.2019.1584116
7. "An Approach to Modeling Closed-Loop Kinematic Chain Mechanisms, Applied to Simulations of the da Vinci Surgical System," *Acta Polytech. Hung.*, vol. 16, no. 8, Sep. 2019, https://doi.org/10.12700/APH.16.8.2019.8.3.
8. Zhao W., Queralta J. P., and Westerlund T., Dec. 2020 "Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 737–744. https://doi.org/10.1109/SSCI47803.2020.9308468.
9. Collins J, Chand S, Vanderkop A, Howard D (2021) A review of physics simulators for robotic applications. IEEE Access 9:51416–51431. https://doi.org/10.1109/ACCESS.2021.3068769
10. Kazanzides P., Chen Z., Deguet A., Fischer G. S., Taylor R. H., and DiMaio S. P., May 2014 "An open-source research kit for the da Vinci® Surgical System," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong: IEEE, pp. 6434–6439. https://doi.org/10.1109/ICRA.2014.6907809.
11. Richter F., Orosco R. K., and Yip M. C., Jan. 27, 2020 "Open-sourced reinforcement learning environments for surgical robotics", *arXiv*: arXiv:1903.02090. https://doi.org/10.48550/arXiv.1903.02090.
12. P. M. Scheikl *et al.*, "LapGym -- An open source framework for reinforcement learning in robot-assisted laparoscopic surgery," Feb. 19, 2023, *arXiv*: arXiv:2302.09606. https://doi.org/10.48550/arXiv.2302.09606.
13. Wu J, Zhou H, Kazanzides P, Munawar A, Liu A (2024) SurgicAI: a hierarchical platform for fine-grained surgical policy learning and benchmarking. Adv Neural Inf Process Syst 37:63771–63789
14. Varier V. M., Rajamani D. K., Tavakkolmoghaddam F., Munawar A., and Fischer G. S., Apr. 2022 "AMBF-RL: A real-time simulation based Reinforcement Learning toolkit for Medical Robotics," in *2022 International Symposium on Medical Robotics (ISMR)*, GA, USA: IEEE, pp. 1–8. https://doi.org/10.1109/ISMR48347.2022.9807609.
15. Xu J., Li B., Lu B., Liu Y.-H., Dou Q., and Heng P.-A., Aug. 30, 2021 "SurRoL: An Open-source reinforcement learning centered and dvrk compatible platform for surgical robot learning", *arXiv*: arXiv:2108.13035. https://doi.org/10.48550/arXiv.2108.13035.

16. Tagliabue E., Pore A., Dall'Alba D., Piccinelli M., and Fiorini P., "UnityFlexML: Training Reinforcement Learning Agents in a Simulated Surgical Environment".
17. Y Ou M Tavakoli 2025 CRESSim-MPM: A material point method library for surgical soft body simulation with cutting and suturing arXiv arXiv:2502.18437 https://doi.org/10.48550/arXiv.2502.18437
18. Abdelaal AE, Liu J, Hong N, Hager GD, Salcudean SE (2021) Parallelism in autonomous robotic surgery. IEEE Robot Autom Lett 6(2):1824–1831. https://doi.org/10.1109/LRA.2021.3060402
19. Varier V. M., Rajamani D. K., Goldfarb N., Tavakkolmoghaddam F., Munawar A., and Fischer G. S., Aug. 2020 "Collaborative suturing: a reinforcement learning approach to automate hand-off task in suturing for surgical robots," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, Naples, Italy: IEEE, pp. 1380–1386. https://doi.org/10.1109/RO-MAN47096.2020.9223543.
20. Munawar A, Wu JY, Fischer GS, Taylor RH, Kazanzides P (2022) Open simulation environment for learning and practice of robot-assisted surgical suturing. IEEE Robot Autom Lett 7(2):3843–3850. https://doi.org/10.1109/LRA.2022.3146900

21. K. Fan *et al.*, "A unity-based da vinci robot simulator for surgical training," in *2022 9th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, Aug. 2022, pp. 1–6. https://doi.org/10.1109/BioRob52689.2022.9925319.
22. Fontanelli G. A., Ficuciello F., Villani L., and Siciliano B., "Da Vinci research kit: PSM and MTM dynamic modelling".
23. M. Hwang *et al.*, "Efficiently Calibrating Cable-Driven Surgical Robots with RGBD Fiducial Sensing and Recurrent Neural Networks," 2020, https://doi.org/10.48550/ARXIV.2003.08520.

Authors and Affiliations

Luigi Muratore[1,2] · Matteo Pescio[2,4] · Federica Barontini[2] · Francesco Marzola[2] · Pietro Leoncini[2,4] · Carlo Alberto Ammirati[2] · Alberto Arezzo[2] · Giulio Dagnino[2,3] · Giuseppe Averta[5]

✉ Federica Barontini
federica.barontini@unito.it

Luigi Muratore
gigiomuratore@gmail.com

Matteo Pescio
matteo.pescio@unito.it

Francesco Marzola
francesco.marzola@unito.it

Pietro Leoncini
p.leoncini@unito.it

Carlo Alberto Ammirati
carloalberto.ammirati@gmail.com

Alberto Arezzo
alberto.arezzo@unito.it

Giulio Dagnino
g.dagnino@utwente.nl

Giuseppe Averta
giuseppe.averta@polito.it

1 DET, Politecnico di Torino, Corso Duca Degli Abruzzi 24, 10129 Turin, TO, Italy

2 Department of Surgical Sciences, Università Degli Studi di Torino, Corso Dogliotti 14, 10126 Turin, TO, Italy

3 Robotics and Mechatronics, University of Twente, Drienerlolaan 5, 7522 Enschede, NB, The Netherlands

4 DIMEAS, Politecnico di Torino, Corso Duca Degli Abruzzi 24, 10129 Turin, TO, Italy

5 DAUIN, Politecnico di Torino, Corso Duca Degli Abruzzi 24, 10129 Turin, TO, Italy