SurRoL: An Open-source Reinforcement Learning Centered and dVRK Compatible Platform for Surgical Robot Learning

Jiaqi Xu[1,*], Bin Li[2,*], Bo Lu[2], Yun-Hui Liu[2], Qi Dou[1], and Pheng-Ann Heng[1]

***Abstract*— Autonomous surgical execution relieves tedious routines and surgeon's fatigue. Recent learning-based methods, especially reinforcement learning (RL) based methods, achieve promising performance for dexterous manipulation, which usually requires the simulation to collect data efficiently and reduce the hardware cost. The existing learning-based simulation platforms for medical robots suffer from limited scenarios and simplified physical interactions, which degrades the real-world performance of learned policies. In this work, we designed SurRoL, an RL-centered simulation platform for surgical robot learning compatible with the da Vinci Research Kit (dVRK). The designed SurRoL integrates a user-friendly RL library for algorithm development and a real-time physics engine, which is able to support more PSM/ECM scenarios and more realistic physical interactions. Ten learning-based surgical tasks are built in the platform, which are common in the real autonomous surgical execution. We evaluate SurRoL using RL algorithms in simulation, provide in-depth analysis, deploy the trained policies on the real dVRK, and show that our SurRoL achieves better transferability in the real world.**

I. INTRODUCTION

Nowadays, robotic surgery systems, such as the da Vinci® system, have been widely used in minimally invasive surgeries, including urology, gynecology, cardiothoracic, and many other procedures. Recently, people have raised increasing interest in autonomous execution for surgical tasks or sub-tasks [1], especially with the help of the open-source da Vinci Research Toolkit (dVRK) [2], which significantly relieves tedious routines and reduces the surgeon's fatigue. Nonetheless, substantial specific expertise of individual skills and a complicated development process are required to design the manually-tuned control policies [3], [4], [5].

Learning-based methods, especially reinforcement learning (RL) based methods, provide a promising alternative to automating manual effort. These approaches are able to develop controllers for complex skills and generalize to a broader range of tasks and environments [6], [7]. However, robot learning typically requires a large amount of labeled data and interactions with the environment [8], [9], [10], usually infeasible on real surgical robots due to the expensive time cost and the hardware wear and tear issue.

One intuitive choice to efficiently collect data and fast prototype for learning-based algorithms is to use the simulation, where we generate a set of labeled training data through the computer. Preliminary works mitigate the limited access situation by proposing medical robot simulation platforms with robotics tasks [11], [12]. More recently, the learning-based platforms, dVRL [13] and UnityFlexML [14], build the RL simulation environments for surgical robots on top of [11] and Unity, paving the way for follow-up research on surgical manipulation [15] and perception [16].

However, the existing learning-based platforms only support limited scenarios in the simulated environments [13], [14], detailed in Table I. The models trained on such platforms ignore some important scenarios, such as bimanual patient side manipulator (PSM) manipulation and endoscopic camera manipulator (ECM) control. Moreover, the physical interactions supported by the current learning-based simulators are simplified. For example, they consider it is successfully grasping the objects when the relative distance between the jaw tip and the object is smaller than a threshold. The modeled trained on such simulated settings may suffer from the reality gap and fail to transfer to the real world [14].

In this work, we build a novel surgical robotic simulation platform, **SurRoL**, which is an open-source RL-centered and dVRK compatible simulation platform for **Sur**gical **Ro**bot **L**earning. The system design of SurRoL is shown in Fig. 1. Our SurRoL is able to support more surgical operation scenarios by incorporating more single-handed/bimanual PSM(s) and ECM control tasks. Further, the designed SurRoL with carefully modeled assets can successfully deal with more realistic physical interactions. Code is publicly available at https://github.com/med-air/SurRoL.

Our main contributions are summarized as follows:

- We design an open-source surgical robot learning simulation platform centered on reinforcement learning for surgical skills, which benefits low-cost data collection and accelerates the development of learning-based surgical robotic methods.
- We build the dVRK compatible simulated environment based on the real-time physics engine, which includes diverse surgical contents and physical interaction. We build ten tasks (e.g., single-handed/bimanual PSM and ECM manipulation) in the platform, which are common in the real autonomous surgical execution.
- We conduct extensive experiments for RL algorithm evaluation in simulation using the proposed tasks, provide in-depth analysis, and deploy the trained policies on the real dVRK. Results show that our SurRoL considering more rich physical interactions achieves better transferability in the real world.

[1]J. Q, Xu, Q. Dou, and P. A. Heng are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. Q. Dou and P. A. Heng are also with the T Stone Robotics Institute, CUHK.

[2]B. Li, B. Lu, and Y. H. Liu are with the Department of Mechanical and Automation Engineering, and T stone Robotics Institute, The Chinese University of Hong Kong.

The first two authors contributed equally.

Corresponding author: Qi Dou (qidou@cuhk.edu.hk)

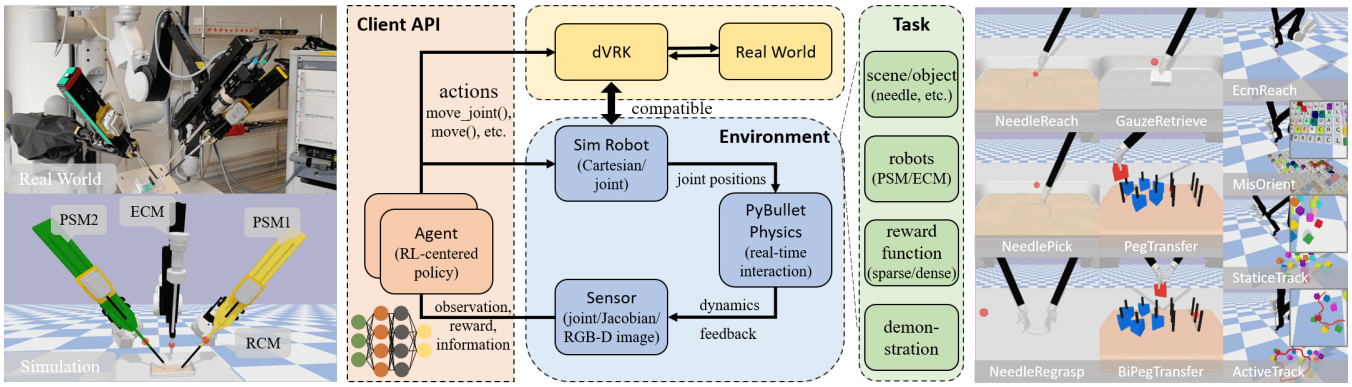

Fig. 1. **System design of SurRoL.** SurRoL provides dVRK compatible simulation environments for surgical robot learning (left), with Gym-like interfaces for reinforcement learning algorithm development and ranges of surgical contents with physical interaction (middle). Ten constructed surgical relevant tasks with difficulty levels and varying scenes are presented for learning-based algorithm evaluation (right).

TABLE I

COMPARISON TO EXISTING SURGICAL ROBOT LEARNING SIMULATION ENVIRONMENT

	Physics	Objects	ECM Support	Action DoF	Bimanul Task	Task Number	Interface
dVRL [13]	Static+	Cylinder	✗	3	✗	2	Python, V-REP
UnityFlexML [14]	Static+	Fat tissue	✗	3	✗	1	Python, Unity
SurRoL (ours)	Dynamic	Needle, Block, etc.	✓	4	✓	10	Python

Static+: grasp the object using the simplified attachment manner with limited physical interaction.

II. RELATED WORK

A. Reinforcement Learning for Robotics

Most of the deep RL's success for complex robotics manipulation skills originates from large amounts of interactions, using real-world robots or physics simulations. Recent approaches leverage the data-driven manner to iteratively collect the data with physical robots and optimize the policy for continuous control, including grasping [8], poking [17], door opening [6], etc. However, there are limited dVRK available worldwide with more strict safety concerns. Alternatively, simulation is an proxy to real robots, with the benefits of low time cost and safety guarantee [18]. Still, due to the non-trivial development process, there is no learning-based dVRK compatible simulation environment with ranges of surgical contents, tasks, and reasonable physical interaction.

B. Learning Surgical Manipulation

Previous robotics efforts in surgical skills concentrate on the sophisticated controllers specifically design for sub-tasks including, looping [4], knot-tying [4], needle manipulation [3], [5], cutting [19], tissue dissection [20], endoscopic guidance [21], [22]. Although these carefully tuned methods can handle separate tasks reasonably, designing these algorithms exhibits substantial expertise requirements and generalization ability concerns. Instead, learning-based methods, typically RL, demonstrate a significant advantage in task generalization and surgical automation with improved performance [23]. Therefore, we propose an easy-to-use simulated environment with low data collection cost and state-of-the-art reinforcement learning to facilitate surgical robotics manipulation development.

C. Simulation Platform for Robot Learning

With the advancement of physics simulation and the demand for RL algorithm development, there has been a surge in robotics simulation platforms. OpenAI Gym [24] is widely used in RL as a benchmark. Other platforms focus on different features, e.g. RLBench with a wide range of manipulation tasks [25], SAPIEN with home assistant robots [26], and RoboSuite with reproducible research [27]. However, surgical robots remain few attempts in simulations. Fontanelli et al. [11] relieve the situation by developing a dVRK V-REP simulator with operation scenes. Though AMBF [12] can produce dynamic environments with medical robots, it provides minimal learning environment support. The most related works to ours are dVRL [13] and UnityFlexML [14], reinforcement learning platforms for dVRK. However, the low capacity of tasks with limited physical interaction restricts their functionality and sim-to-real transferability. In this work, we develop a robot learning environment with improved scenarios and physics simulation, opening ways for future progress in surgical manipulation.

III. METHODS

To provide a simulated platform for surgical robot learning, we first build a user-friendly RL library for agents to interact with. Then, we construct the dVRK robots and surgical contents on top of the physics engine. Finally, ten surgical learning-based tasks are built for algorithm development and evaluation. SurRoL builds on top of the open-source PyBullet because of its state-of-the-art physics simulation, wide adoption in the machine learning community, and removal of the commercial software limits, e.g., V-REP.

为每个任务设计algo太难太麻烦了，RL全学会，泛化性能又佳

A. SurRoL RL Library

SurRoL enables surgical robot learning by providing the widely used Gym-like RL environment interface for algorithm development and evaluation.

1) Background: Given the partially observed model of the system dynamics, we formulate the manipulation problem into the Markov Decision Process (MDP), represented by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. The agents interact with the environment, receive the current state $s_t \in \mathcal{S}$, reward $r_t \in \mathcal{R}$ based on the task specification, and generate the action $a_t = \pi(s_t) \in \mathcal{A}$ according to their policies π at each step t, which forms the trajectory $\tau = \{s_0, r_0, a_0, ..., a_{t-1}, s_T, r_T\}$ with a discount factor $\gamma \in [0, 1)$, where T denotes the episode time horizon. The transition probability $\mathcal{P}(s_{t+1}|s_t, a_t) \in [0, 1]$ is computed by the underlying physics engine.

2) Action Space: In practice, people frequently change the dVRK robot base frame relative to the virtual world frame, so the Cartesian-space control is used as action space, which is easier to transfer across different settings. Though SurRoL supports six degrees of freedom (DoF) motion, we focus on the PSM tasks with the rotation within a plane in this work. Specifically, we restrict the action space to $(d_x, d_y, d_z, d_{yaw}/d_{pitch}, j)$, where d_x, d_y, d_z determine the position movement in the Cartesian space, d_{yaw}/d_{pitch} determines the orientation movement in a top-down or vertical space setting, respectively, and j determines whether the jaw is open ($j \geq 0$) or close ($j < 0$). For ECM, besides the Cartesian-space position control, the velocity of the camera c in its own fame ${}^cV_c$ or the roll angle control d_{roll} is used when the observation is in the camera space.

3) Observation Space: SurRoL supports two observation methods, *i.e.* low-dimensional ground-truth states (e.g., object 3D Cartesian positions, 6D poses), and high-dimensional RGB, depth, mask images rendered by OpenGL. The first manner abstracts away the perception procedure and lets the agents concentrate on continuous control learning with sample efficiency. The latter requires raw image perception, which is essential in robotic control. In this work, we focus on the low-level continuous control skills for reinforcement learning as some built tasks are challenging even in this setting. Unless stated otherwise, we use the low-dimensional object state (object position, orientation, etc.) and robot proprioceptive features (tip position, jaw status) represented by a fixed-length vector as the observation.

4) Reward Function: As reward shaping can be difficult to scale in practice [18], most SurRoL tasks are goal-based. The agent receives a binary reward $r_g(s, a) = -\mathbb{I}_{f(s,a,g)}$ given the goal g requirement and the condition success check function $f(s, a, g)$, and receives a negative reward unless the goal requirement is met. While in the ECM continuous tracking task, the tracked object is constantly moving. A dense reward function $r_d(s, a)$ is designed, which encourages the agent to follow the target.

5) Algorithms: Reinforcement learning algorithms aim to achieve the specified goal by learning a policy π to maximize the expected return $\mathbb{E}_\pi[\sum_{t=0}^{T} \gamma^t r_t]$. Our RL library is compatible with the popular OpenAI Gym [24], which provides an easy-to-use interface for state-of-the-art RL algorithm evaluation and benchmark, such as DDPG [9], PPO [10], etc. Meanwhile, our tasks detailed in section III-C involve long-horizon reasoning and can enable future research with more recent RL advances, e.g., learned skill priors [28].

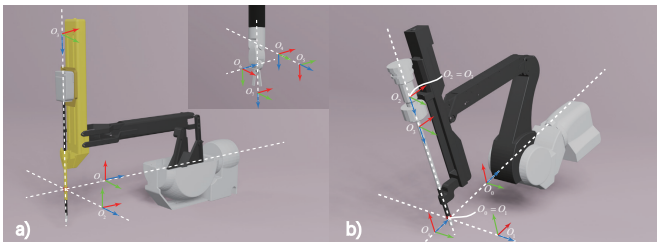

Fig. 2. **PSM and ECM kinematics.** a) PSM is a 6-DoF actuated robot with instruments. b) ECM is a 4-DoF actuated robot with the camera mounted.

B. SurRoL Physics Engine

We build the dVRK compatible simulation environment by supporting PSM and ECM manipulation with diverse surgical contents, based on the state-of-the-art physics simulation with relatively rich robotic interactions.

1) Physics Simulation: We build our simulation environment based on PyBullet [29], a Python wrapper API for the real-time Bullet physics. Unlike previous works that approximate the grasping by attaching the object to the jaw when the tip-object relative distance is below a certain threshold [13], [14], we consider more realistic scenarios by enabling inter-object physical interactions and friction-based grasping. The grasping is stabilized only if the PSM can lift the grasped object above a threshold, which introduces the realism and difficulties in low-level skill learning.

2) Compatible dVRK Robot: Our simulation platform considers the manipulation of both PSM and ECM, which is compatible with the dVRK interface, as shown in Fig. 2. We build our dVRK robots based on the meshes from AMBF [12]. As dVRK robots contain many redundant mechanisms with parallel linkages, we rebuild the link frames into a serially linked kinematic chain and use the built-in inverse kinematics. While PyBullet supports the off-the-shelf velocity and torque control, the dynamics discrepancy between the simulation and the real world is more significant than position control [30], beyond the scope of this work. The simulated robots behave the identical joint-space and Cartesian-space action with the real dVRK, which allows commonly used high-level control and smooth transfer.

PSM has seven DoFs, where we consider the first six DoFs (q_i) since the last DoF corresponds to the jaw angle. PSM includes the revolute (R) and prismatic (P) actuated joints formed in an RRPRRR sequence (Fig. 2, a). ECM is a 4-DoF actuated arm with an RRPR sequence (Fig. 2, b). Note that the calculated tip pose from the forward kinematics is not the final jaw/camera pose [2]. We adopt the transformation matrix ${}^{tip}T_{tool}$ to transform the tip pose to the tool pose. Finally, we acquire the tool pose ${}^{base}T_{tool}$ relative to the remote center of motion (RCM) as the base frame.

TABLE II

SURROL TASK SPECTRUM SUMMARY

	Arm	Action	Bimanual	Reward
NeedleReach	PSM	d_{pos}		Sparse
GauzeRetrieve	PSM	d_{pos}, j		Sparse
NeedlePick	PSM	d_{pos}, d_{yaw}, j		Sparse
PegTransfer	PSM	d_{pos}, d_{yaw}, j		Sparse
NeedleRegrasp	PSM	d_{pos}, d_{pitch}, j	✓	Sparse
BiPegTransfer	PSM	d_{pos}, d_{yaw}, j	✓	Sparse
EcmReach	ECM	d_{pos}		Sparse
MisOrient	ECM	d_{roll}		Sparse
StaticTrack	ECM	${}^cV_c$		Sparse
ActiveTrack	ECM	${}^cV_c$		Dense

d_{pos}: d_x, d_y, d_z; j: jaw open/close.

3) Object Asset: To enrich the manipulated contents and reflect the challenges during control, we create the SurRoL object asset (e.g., $40mm$ suture needle and pegboard), modeled using Blender. All the articulated object links are organized in the tree structure following the URDF format. We randomly or manually tune the object's physically related parameters, including shape, mass, friction, to mimic the real-world counterparts. To enable reliable collision detection and physical interaction between instruments and objects, we extract the mesh convex decomposition using V-HACD [31].

C. SurRoL Task Spectrum

We have established a spectrum of learning-based tasks given the dexterity and precision properties in the surgical context, which covers levels of surgical skills and involves manipulating PSM(s) and ECM. We build ten tasks with diversity, including nine goal-based tasks (four PSM single-handed tasks, two PSM bimanual tasks, three ECM tasks) and one reward-based ECM task, ranging from entry-level to sophisticated counterparts, as summarized in Table II.

1) NeedleReach: This serves as a validation task for the environment since, with hindsight experience replay [32], the policy can quickly acquire the skill. The goal is to move the PSM jaw tip to the location slightly above the needle within a tolerance ϵ, where the needle is randomly placed on a surgical tray, and the jaw is close and of fixed orientation.

2) GauzeRetrieve: Imagine that we want to retrieve the suture gauze during a surgical operation. The goal is to sequentially pick the gauze and get it back (place it at the target position), with one DoF to indicate the jaw open/close.

3) NeedlePick: Based on *GauzeRetrieve*, *NeedlePick* involves an additional yaw angle DoF, which considers the pose of the needle.

4) PegTransfer: Peg transfer is one of the Fundamentals of Laparoscopic Surgery (FLS) tasks for hand-eye coordination [33], which requires collision avoidance and long-horizon reasoning. We build a single-handed version that moves the block from one peg to the other peg without handover.

5) NeedleRegrasp: Initial needle grasp with one PSM often results in a non-ideal picking pose. This task requires to hand over the held needle from one arm to the other arm with bimanual operations [34].

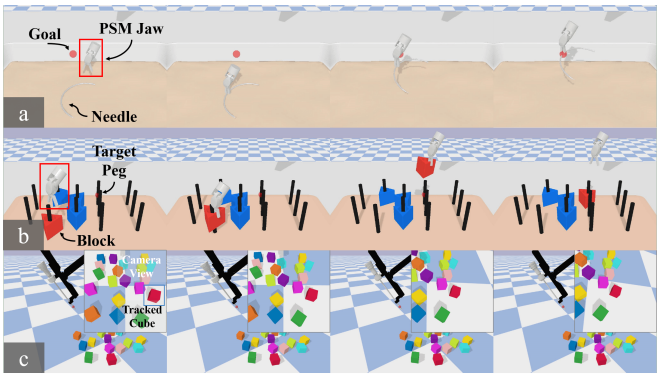

Fig. 3. **Examples of the demonstration.** To demonstrate the proposed tasks and overcome the sample complexity, we provide the scripted policy for each and collect small amounts of demonstration data for complex ones. With efficient imitation learning, the policy can reasonably solve many challenging tasks that failed otherwise, e.g., *PegTransfer*.

6) BiPegTransfer: This is an advanced version of *PegTransfer* with bimanual operations, where the grasping arm needs to hand the block to the other arm before placing it.

7) EcmReach: Similar to the *NeedleReach*, the goal is to move the camera mounted on ECM to a randomly sampled position. Note that the 4th joint is fixed since it does not affect the camera position but only alters the orientation.

8) MisOrient: Misorientation, the difference between the camera orientation and the Natural Line-of-Sight (NLS), is inevitable during surgery since the endoscope moves under the RCM constraint. This task requires adjusting the ECM's 4th joint such that the misorientation θ^* with the desired NLS is minimized, which is computed from an affine transformation $\mathbf{A}$. The goal is achieved when θ^* is within δ.

9) StaticTrack: The goal is to let the ECM track a static target cube with red color, disturbed by other surrounding cubes, that mimics the scenario to focus on the primary instrument during surgery. A successful tracking requires the tracked cube position p_t^{ij} in image space close to the image center p_c and the misorientation θ^* is less than δ.

$$r_g(s,a) = -\mathbb{I}_{\left\|p_t^{ij}-p_c\right\|_2<\epsilon \cap |\theta^*|<\delta} \tag{1}$$

10) ActiveTrack: Instead of remaining static in the given place, the target cube keeps moving and follows an online generated path at a constant speed. The goal is to keep the ECM tracking the moving cube, with a relaxed misorientation requirement but a chance to lose the target out of the view. A dense reward $r_d(s,a)$ is designed as follows:

$$r_d(s,a) = C - (\left\|p_t^{ij} - p_c\right\|_2 + \lambda \cdot |\theta^*|) \tag{2}$$

where p_t^{ij} and p_c are the same as Equ. 1, and hyperparameters C and λ are chosen as 1 and 0.1, respectively.

IV. EXPERIMENTS

In this section, we focus on the proposed learning-based tasks and want to answer the following questions: 1) Do the tasks in our simulation platform cover a range of surgical

scenarios and difficulties? 2) Does the physical interaction matter for surgical robots with low-level skills? 3) Can we smoothly transfer the policy trained in the simulated environment to the real dVRK?

To answer these questions, we first evaluate recent reinforcement learning algorithms optional with imitation learning in SurRoL. Secondly, we give an in-depth analysis of the physical interaction effect using the PSM manipulation task. Finally, we demonstrate that with the highly compatible interface, the policy from the simulation can successfully deploy on the dVRK, including PSM *GauzeRetrieve*, *NeedlePick*, *PegTransfer*, and ECM *StaticTrack*.

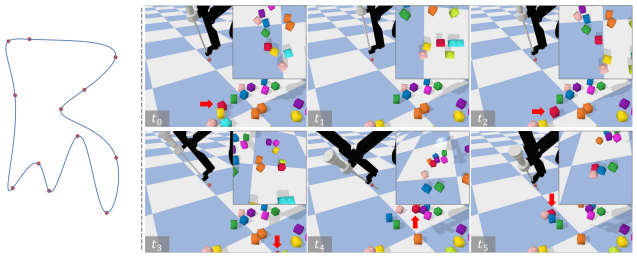

Fig. 4. **Example of the reward-based environment ECM *ActiveTrack*.** Each time environment resets, waypoints are sampled in the workspace randomly, generating the moving path online with B-spline interpolation (left). One trajectory of the policy trained using DDPG in simulation with the tracked cube marked by the red arrow is shown (right).

A. Evaluation in SurRoL

The initial experiment is to verify our proposed tasks are solvable using existing reinforcement algorithms. As we find the manipulation tasks extremely challenging, mainly due to the tiny objects with the high precision requirement, we present the results with low-dimensional state observations.

1) Experiment Setup: In our RL environments, we set up the manipulation workspace for robots and objects to interact within. For PSM tasks, the workspace is of the size $10cm^2$ and the goal tolerance distance $\epsilon = 0.5cm$. Every time the environment resets, the initial object and goal positions are randomly sampled from the workspace. For ECM tasks, the workspace for the target cube is $80cm^2$, the misorientation tolerance $\delta = 0.01rad$, and the normalized image position error $\epsilon = 0.01$. Each episode lasts for 50 timesteps for goal-based tasks and 500 timesteps for reward-based tasks.

For all tasks, we evaluate with the model-free RL algorithms, including the off-policy method deep deterministic policy gradient (DDPG) [9] and the on-policy method proximal policy optimization (PPO) [10]. We collect the agent experience interacting in multiple separate environments during training and maintain a shared replay buffer for gradient update. As model-free methods suffer from the sample complexity, we also evaluate the hindsight experience replay (HER) [32], a sample efficient learning algorithm desirable for goal-based tasks. The success rates and episode returns are used as the evaluation metrics for goal-based and reward-based tasks, respectively, as in [32], [9], [10].

2) Profiling Analysis: Our SurRoL can run at a real-time rate, at about 150Hz simulation in the reaching tasks with position control and random actions, where the environment is stabilized at each time with multiple simulation steps. Most of the training and testing experiments are performed on a desktop with Ubuntu 18.04, Inter 3.6GHz CPU with 32GB RAM, and an Nvidia TITAN RTX GPU.

3) Demonstration using Scripted Policies: To demonstrate our manipulation tasks, we design scripted policies with heuristics given the ground-truth states available in the simulation, with the help of manual engineering [18]. Meanwhile, it is yet challenging to obtain satisfactory RL performance for the PSM tasks, such as *NeedlePick* and *PegTransfer*, which contains rich physical contacts between the instruments and the objects. RL algorithms typically suffer from the exploration problem to discover the high reward space when the agents are trained from scratch, especially in the sparse reward setting. To sidestep exploration challenges and ease the training, we integrate the demonstrations into the learning process by collecting a small number of samples using scripted policies for behavior cloning.

Specifically, we can divide the PSM manipulation tasks into a multi-stage sequence, where waypoints are utilized to indicate the critical changing conditions between each simplified operations. E.g., the trajectories for *NeedlePick* and *PegTransfer* are composed of approaching, picking, placing, and optional releasing, as shown in Fig. 3 (a), (b). Waypoints are built manually with the position, orientation, and collision avoidance consideration, while the trajectories in-between are generated using the interpolation method. Besides, we demonstrate the ECM tracking tasks using visual servoing, implemented by a null space method for camera velocity ${}^cV_c$ control as in Fig. 3 (c).

4) Evaluation Results: A summary of the evaluation results for RL baselines is shown in Fig. 5. For ECM goal-based tasks without instrument-object physical interaction, the agent can successfully capture the complicated action-observation relationship using HER, even for *MisOrient* and *StaticTrack*, which involve complex matrix transformations. We also observe that in *StaticTrack*, the learned policy can smoothly center the target object without the jittering effect, which is non-trivial for the visual servoing method that requires careful parameter tuning. For the reward-based task, DDPG, with the proposed dense reward Equ. 2, incentivizes the agent to actively control the ECM and track the moving object in a dynamic environment, which follows online generated moving paths, as illustrated in Fig. 4.

However, in PSM settings, HER alone cannot solve all tasks within the given time horizon, mainly due to the tiny object and physically rich interaction nature. By visually inspecting the training progress, we find that the agents can quickly learn to approach the object such as the needle and attempt to pick reasonably, but failed because of the approximate positioning exceeding millimeters tolerance and unstable grasping. Few experiences with high reward lead the learning to diverge in the early stage, as the policy gradually finds that random actions produce similar no-gain returns.

To overcome the exploration challenge, we record a small amount of demonstration data using the scripted policies for

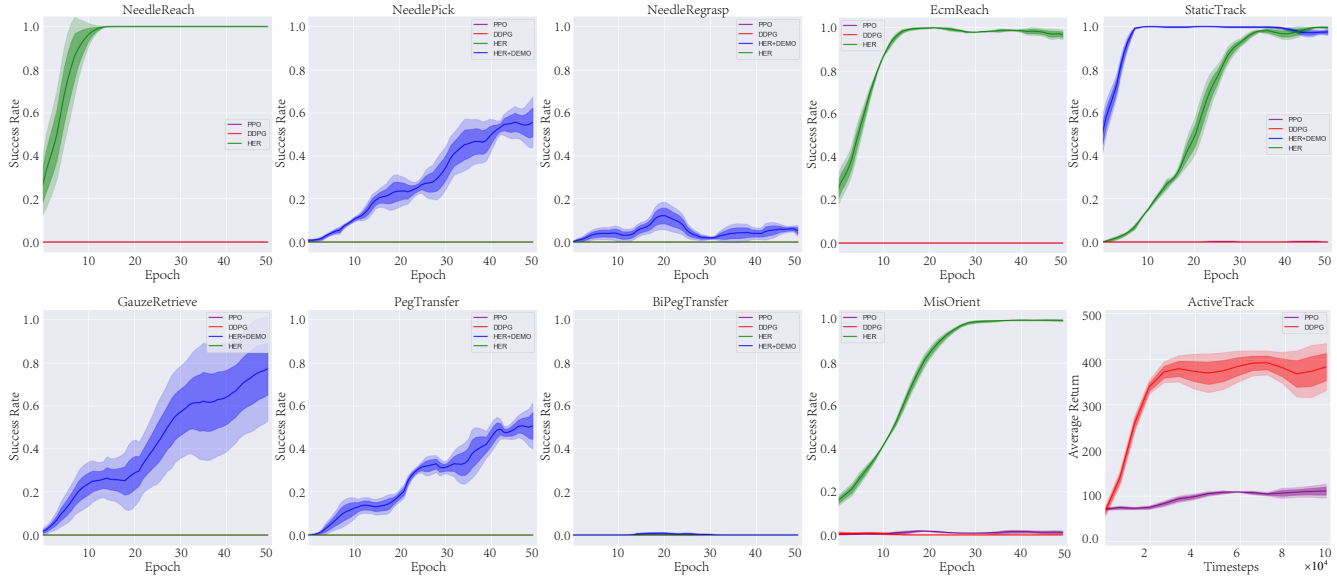

Fig. 5. **Evaluation results for ten proposed tasks.** The average success rates for goal-based tasks and episode returns for the reward-based task (*ActiveTrack*) are shown over three random seeds, with one epoch equalling 40 episodes. The light shaded region denotes the standard deviation (std); the dark shaded region denotes standard error (std divided by the square root of the epoch length).

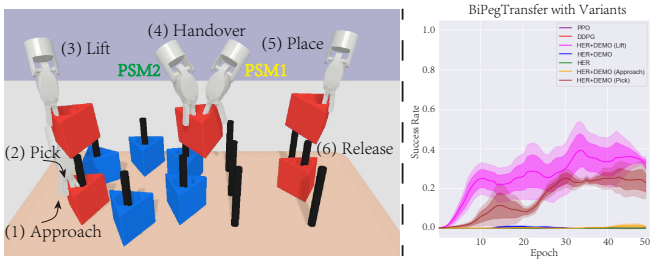

Fig. 6. **Analysis of the *BiPegTransfer* using HER+DEMO.** We analyze the difficulty for long-range skill learning by segmenting the bimanual peg transfer task into multiple steps with simplified initialization variants during environment reset (left, 1, 2, 3). Comparing the learning curves when the initialization is accomplished with "approach" and "pick," the stable picking skill is challenging to acquire (right).

imitation learning. After combining HER and demonstration (HER+DEMO) with Q-filtered behavior cloning [35], the agents manage to solve many challenging tasks with physics-rich simulation within 50 epochs of training, e.g. *PegTransfer*. From the results, though HER(+DEMO) performs well for robots with relatively large grippers and error tolerance [32], it performs poorly with tiny surgical instruments and objects (around 10 times smaller error tolerance), which indicates the difficulties in the medical robot field.

We further analyze the most challenging long-range *BiPegTransfer* failed even with imitation learning by constructing several variants with different levels of simplification. As shown in Fig. 6 left, we initialize the environment by letting the PSM2 accomplish the approach, pick and lift step manually, to inspect which part makes HER+DEMO suffer. Surprisingly, even with the correct grasping points, HER+DEMO fails to learn the picking action, which shows the extreme exploration difficulties during learning (Fig. 6, right). With successful picking and lifting, the agents succeed in handing over the blocks from PSM2 to PSM1, a non-trivial coordination skill. From the disentangled analysis, integrating motion planning and low-level control is one way to solve long-range peg transfer efficiently [36].

5) Physics-based Grasping Analysis: As we find the simplified instrument-object interaction in [13], [14] may cause unstable grasping with further sim-to-real reality gap, we evaluate different physical interaction levels using *NeedlePick*. Note that the simulation backends are different among the works, so we construct a similar environment to mimic the simplified setting, i.e., the needle is attached to the jaw when the relative distance is less than $2mm$ [14], which is denoted as "Approx@$2mm$". Meanwhile, the needle picking point is restricted to the jaw tip to avoid unsafe jaw collisions with the holding surface. We compare the approximate manner with ours using physical interaction and friction-based grasping (denoted as "Interact"), shown in Fig. 7 top. The mean success rate and standard deviation of three trained policies for the two manners are presented based on the evaluation of 200 episodes per model in Table. III.

We show the experimental results when the policy is trained in one physical interaction manner and tested in other settings. Though the transition probability $\mathcal{P}$ is changed with interaction manners, polices trained with Interact are robust in the Approx settings (from 81.3% in Interact to 70.7% in Approx@$2mm$), which indicates the learned accurate picking points. However, polices trained in the Approx settings suffer from dynamics confusion and significant performance degeneration while in Interact (from 76.5% to 34.2%) and usually fail with unrealistic no-contact grasping. A relatively large performance improvement in a relaxed setting also reflects the inaccurately learned dynamics (from 76.5% in Approx@$2mm$ to 88.8% in Approx@$3mm$).

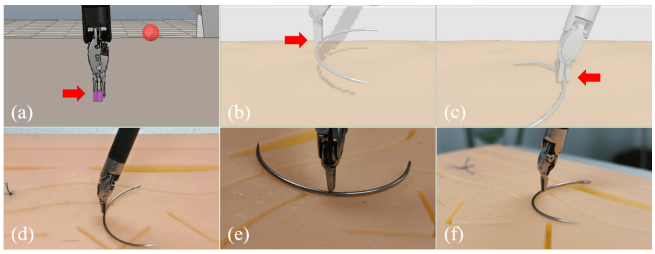

Fig. 7. **Different levels of physical interaction.** The object is attached to the jaw if the tip-object distance is below a certain threshold with limited interaction [13] in (a). We compare the physical interaction effects by constructing a similar setting "Approx@$2mm$" with an unrealistic simulated grasp example in (b), also with our physical interaction setting "Interact" in (c). Some failure cases caused by the approximate picking point when deploying the trained policy using the "Approx@$2mm$" manner on the real-world dVRK from different viewpoints are shown in (d), (e), and (f).

TABLE III

THE EVALUATION OF *NeedlePick* IN SIMULATION.

Approach	Success Rate (%)			
	Approx @$1mm$	Approx @$2mm$	Approx @$3mm$	Interact
Approx @$2mm$	36.0±12.4	76.5±4.3	88.8±5.9	34.2±16.5
Interact	52.5±9.6	70.7±19.9	76.8±14.2	81.3±1.5

B. Deployment on the Real-World dVRK

To demonstrate transferability, we conduct physical experiments by deploying the policies trained in SurRoL to the real-world dVRK platform. Four tasks, PSM *GauzeRetrieve*, *NeedlePick*, *PegTransfer*, and ECM *StaticTrack*, are selected for demonstration. Thanks to the compatible dVRK interface, we can smoothly transfer the learned skills, with experiment snapshots shown in Fig. 8.

For the first three PSM tasks, we set up the physical experiment following the setting of [37] and carefully align a $10cm^2$ workspace to ensure consistency between the simulated and the real environment. Since the tasks can be solvable only with HER+DEMO, we select the corresponding best-performance policies trained in simulation with actions generated for deployment. With 4-DoF actions to adjust the PSM position and the jaw's open/close state, the learned *GauzeRetrieve* policy can pick and retrieve the gauze to the target position within $5mm$ with a 96% success rate on 25 episodes. For the *PegTransfer*, the learned policy sequentially picks the block, lifts it, and puts it to the target peg while avoiding collisions in the complex environment.

To investigate the reality gap that different levels of simulated interaction may cause, we conduct the physical *NeedlePick* experiment using the policies introduced earlier. We choose the best policies trained in the Approx@$2mm$ and Interact manner, with a success rate of 82.0% and 83.5% in their corresponding simulated settings, respectively. The physical evaluation environments are set the same with only successful episodes for both policies in simulation to ensure fair comparisons. The success rates are reported based on 50 episodes for each method, as shown in Table. IV. From the result, the policy trained in the Approx@$2mm$ manner suffers from low real-world deployment success rates, mainly due to the imprecise picking points close but without physical contact with the needle (Fig. 7 bottom). By contrast, the policy trained in the Interact manner with improved physics simulation is more robust to environment changes with a high success rate. Besides, we find some failure cases resulting from dynamics discrepancies between the simulation and the real world, also observed in [14].

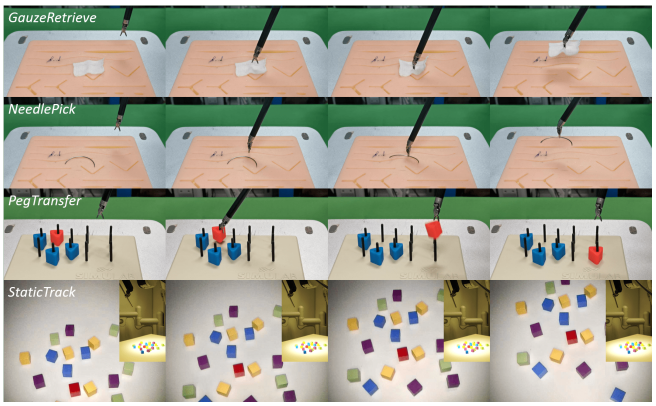

Fig. 8. **Deployment on the real-world dVRK.** Trajectories of four HER(+DEMO) policies on the real dVRK demonstrate that our platform enables the smooth transfer of learned skills from the simulation to the real world. More experiment details can be found in our supplementary video.

TABLE IV

THE EVALUATION OF *NeedlePick* ON THE REAL DVRK.

Approach	Trials	Success Rate (%)
Approx@$2mm$	29/50	58
Interact (ours)	43/50	**86**

For the ECM *StaticTrack*, we mimic the simulated scene with some colored cubes, where the target cube is in red. The best-trained policy using HER is selected to deploy into the real dVRK for ten episodes. The target cube is segmented from the image captured by ECM first, and then the extracted position from the segmentation is served as the observation. The policy generates joint position actions in step, converted from corresponding ${}^cV_c$ expressed in the camera frame, and center the cube in the captured image within a 0.03 normalized position error and $0.1rad$ misorientation error, with a 90% success rate.

V. CONCLUSION

In this work, we present SurRoL, a simulated platform for surgical robot learning compatible with dVRK. Ten learning-based surgical relevant tasks with enriched assets and physical interaction are constructed, which involves manipulating PSM(s) and ECM with difficulty levels. Extensive experiments in simulation with further physical deployment are conducted and reveal the difficulty in low-level surgical skills learning. Moreover, the physical interaction experiments in SurRoL show that reproducing physics is one step towards a realistic simulation for surgical robot learning with transferability to the real world. We believe SurRoL will embrace the advances in learning-based methods, especially RL and surgical robotics, to enable more researchers to be involved in the development of surgical robot manipulation.

ACKNOWLEDGMENT

This project was supported by Hong Kong Research Grants Council TRS Project No. T42-409/18-R, CUHK Shun Hing Institute of Advanced Engineering (project MMT-p5-20), and Hong Kong Multi-Scale Medical Robotics Center.

REFERENCES

[1] T. Haidegger, "Autonomy for surgical robots: Concepts and paradigms," *IEEE Transactions on Medical Robotics and Bionics*, vol. 1, no. 2, pp. 65–76, 2019.

[2] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci® surgical system," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 6434–6439.

[3] J. Schulman, A. Gupta, S. Venkatesan, M. Tayson-Frederick, and P. Abbeel, "A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 4111–4117.

[4] T. Osa, N. Sugita, and M. Mitsuishi, "Online trajectory planning in dynamic environments for surgical task automation." in *Robotics: Science and Systems (RSS)*, 2014.

[5] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, and K. Goldberg, "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)*. IEEE, 2016, pp. 4178–4185.

[6] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3389–3396.

[7] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[8] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[9] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2016.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[11] G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, and B. Siciliano, "A v-rep simulator for the da vinci research kit robotic platform," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*. IEEE, 2018, pp. 1056–1061.

[12] A. Munawar, Y. Wang, R. Gondokaryono, and G. S. Fischer, "A real-time dynamic simulator and an associated front-end representation format for simulating complex robots and environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1875–1882.

[13] F. Richter, R. K. Orosco, and M. C. Yip, "Open-sourced reinforcement learning environments for surgical robotics," *arXiv preprint arXiv:1903.02090*, 2019.

[14] E. Tagliabue, A. Pore, D. Dall'Alba, E. Magnabosco, M. Piccinelli, and P. Fiorini, "Soft tissue simulation environment to learn manipulation tasks in autonomous robotic surgery," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3261–3266.

[15] M. Hwang, D. Seita, B. Thananjeyan, J. Ichnowski, S. Paradis, D. Fer, T. Low, and K. Goldberg, "Applying depth-sensing to automated surgical manipulation with a da vinci robot," in *2020 International Symposium on Medical Robotics (ISMR)*. IEEE, 2020, pp. 22–29.

[16] Y. Li, F. Richter, J. Lu, E. K. Funk, R. K. Orosco, J. Zhu, and M. C. Yip, "Super: A surgical perception framework for endoscopic tissue manipulation with surgical robotics," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2294–2301, 2020.

[17] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: Experiential learning of intuitive physics," in *Neural Information Processing Systems (NeurIPS)*, 2016.

[18] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: lessons we have learned," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 698–721, 2021.

[19] B. Thananjeyan, A. Garg, S. Krishnan, C. Chen, L. Miller, and K. Goldberg, "Multilateral surgical pattern cutting in 2d orthotropic gauze with deep reinforcement learning policies for tensioning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2371–2378.

[20] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W. D. Boyd, S. Lim, P. Abbeel, and K. Goldberg, "Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1202–1209.

[21] T. Osa, C. Staub, and A. Knoll, "Framework of automatic robot surgery system using visual servoing," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 1837–1842.

[22] B. W. King, L. A. Reisner, A. K. Pandya, A. M. Composto, R. D. Ellis, and M. D. Klein, "Towards an autonomous robot for camera control during laparoscopic surgery," *Journal of laparoendoscopic & advanced surgical techniques*, vol. 23, no. 12, pp. 1027–1030, 2013.

[23] M. Yip and N. Das, "Robot autonomy for surgery," in *The Encyclopedia of MEDICAL ROBOTICS: Volume 1 Minimally Invasive Surgical Robotics*. World Scientific, 2019, pp. 281–313.

[24] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[25] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[26] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, *et al.*, "Sapien: A simulated part-based interactive environment," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 097–11 107.

[27] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín, "robosuite: A modular simulation framework and benchmark for robot learning," in *arXiv preprint arXiv:2009.12293*, 2020.

[28] K. Pertsch, Y. Lee, and J. J. Lim, "Accelerating reinforcement learning with learned skill priors," in *Conference on Robot Learning (CoRL)*, 2020.

[29] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[30] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," in *Robotics: Science and Systems (RSS)*, 2018.

[31] K. Mamou, E. Lengyel, and A. Peters, "Volumetric hierarchical approximate convex decomposition," in *Game Engine Gems 3*. AK Peters, 2016, pp. 141–158.

[32] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *Neural Information Processing Systems (NeurIPS)*, 2017.

[33] G. M. Fried, L. S. Feldman, M. C. Vassiliou, S. A. Fraser, D. Stanbridge, G. Ghitulescu, and C. G. Andrew, "Proving the value of simulation in laparoscopic surgery," *Annals of surgery*, vol. 240, no. 3, p. 518, 2004.

[34] S. Lu, T. Shkurti, and M. C. Çavuşoğlu, "Dual-arm needle manipulation with the da vinci® surgical robot," in *2020 International Symposium on Medical Robotics (ISMR)*. IEEE, 2020, pp. 43–49.

[35] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6292–6299.

[36] M. Hwang, B. Thananjeyan, D. Seita, J. Ichnowski, S. Paradis, D. Fer, T. Low, and K. Goldberg, "Superhuman surgical peg transfer using depth-sensing and deep recurrent neural networks," *arXiv preprint arXiv:2012.12844*, 2020.

[37] A. R. Mahmood, D. Korenkevych, G. Vasan, W. Ma, and J. Bergstra, "Benchmarking reinforcement learning algorithms on real-world robots," in *Conference on robot learning (CoRL)*, 2018, pp. 561–591.