Reward Learning from Suboptimal Demonstrations with Applications in Surgical Electrocautery

Zohre Karimi*, Shing-Hei Ho*, Bao Thach, Alan Kuntz, Daniel S. Brown

***Abstract*— Automating robotic surgery via learning from demonstration (LfD) techniques is extremely challenging. This is because surgical tasks often involve sequential decision-making processes with complex interactions of physical objects and have low tolerance for mistakes. Prior works assume that all demonstrations are fully observable and optimal, which might not be practical in the real world. This paper introduces a sample-efficient method that learns a robust reward function from a limited amount of ranked suboptimal demonstrations consisting of partial-view point cloud observations. The method then learns a policy by optimizing the learned reward function using reinforcement learning (RL). We show that using a learned reward function to obtain a policy is more robust than pure imitation learning. We apply our approach on a physical surgical electrocautery task and demonstrate that our method can perform well even when the provided demonstrations are suboptimal and the observations are high-dimensional point clouds. Code and videos available here: https://sites.google.com/view/lfdinelectrocautery**

I. Introduction

As medical care demands increase worldwide, the human surgeon shortage is becoming more pressing [1]. Training surgical robots for specific tasks has the potential to help decrease surgeon workload and enhance the precision of surgeries [2]. However, surgical tasks are challenging as they require sequential decision making with complex deformable physical interactions and a low tolerance for error. Furthermore, an ideal surgical robot should be able to infer a human's underlying task objectives and intent even if optimal human demonstrations are not available [3]. Our proposed approach uses pairwise preference labels over suboptimal trajectory data to capture the demonstrator's intent in the form of a learned reward function that can be optimized via reinforcement learning to yield a robust robot policy.

Our work builds on prior research on Learning from Demonstration (LfD), which has shown to be one of the most effective solutions for enabling robots to learn to perform complex tasks [4]–[6]. However, existing methods often suffer from two major drawbacks. First is the assumption of fully observable states. While some past surgical LfD work has assumed full knowledge of object positions and properties [7]–[9], in practice, a fully observable state space is not achievable for surgical robots with point cloud observations [7], [10]. Second, most prior work in surgical LfD only considers optimal or near-optimal demonstrations [7], [11], which may not always be available. This assumption can lead to potential overfitting to the suboptimalities in the demonstrations and poor performance. [4], [5].

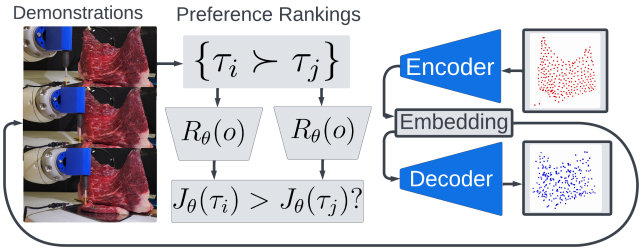


Fig. 1: Our proposed method first learns a latent feature representation by pre-training an autoencoder to reconstruct partial-view point clouds. Then, given pairwise preferences over demonstrations with observations encoded by the latent feature representation, our method learns a reward function that maximizes the likelihood of the pairwise preferences.

To address the problem of partial observability, we use a point cloud autoencoder to learn a low-dimensional feature vector of the partial point cloud scene. To address the problem of suboptimal demonstrations, we leverage ideas from prior work on learning reward functions based on preference labels over suboptimal trajectories [12] to learn a robust reward function suitable for surgical tasks with point cloud observation embeddings as input to the reward function.

We first demonstrate our approach in two simulated surgical electrocautery tasks where we demonstrate 64.13% and 44.70% improvements over pure imitation learning, respectively. Next, we demonstrate proof of concept on a physical electrocautery task with ex vivo bovine muscle tissue, achieving five successful trials out of seven trials. Our work takes the first steps towards learning complex surgical tasks via reward learning from human feedback. Importantly, our approach is able to learn from preference labels over suboptimal task executions. This reduces the need for near-optimal demonstrations and opens the door to surgical policy learning from qualitative human evaluations.

II. Related Work

A variety of LfD approaches have been developed to solve tasks in the surgical domain. Kim et al. use behavioral cloning with image observation to automate tool navigation in retinal surgery [10]. Huang et al. develop a policy network

* Equal Contribution. Robotics Center and the Kahlert School of Computing at the University of Utah, Salt Lake City, UT 84112, USA; (email: {zohre.karimi, shinghei.ho, bao.thach, alan.kuntz, daniel.brown}@utah.edu). This material is based upon work supported in part by the National Science Foundation under grant number 2133027. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

to automate context-dependent surgical tasks [8]. Pore et al. combine generative adversarial imitation learning and model-free reinforcement learning (RL) to automate soft-tissue retraction [7]. However, prior work requires near-optimal demonstrations. By contrast, we learn policies from suboptimal training data. Furthermore, performing RL on the learned reward allows us to avoid the common LfD problem of compounding error since the learner visits states induced by its policy during training [13]. In contrast to prior work that leverages deep RL on surgical tasks [14]–[16], our approach does not require a hand-crafted reward function and works with complex, partial observations of the surgical scene.

Our work is an instance of reinforcement learning from human feedback (RLHF) [17]–[20]. Compared to near-optimal demonstrations and numerical reward labels, RLHF methods only require relative judgment over behaviors which is easier to provide [17], [21]. Prior work leverages online human preferences over trajectories generated by an RL agent to update a learned reward function and a policy interactively [18]. However, on-policy RL is sample inefficient, so querying humans during policy learning may require a prohibitive amount of human supervision. Our research method is inspired by prior work on offline preference-based reward learning that leverages suboptimal demonstrations and learns a reward model from pairwise preferences over these trajectories, enabling better-than-demonstrator performance [12], [22].

Our work seeks to learn electrocautery robot policies from demonstrations. While electrocautery is a common surgical task [23]–[27], we are not aware of prior work applying learning from demonstrations and reward learning to electrocautery. We model surgical electrocautery as sequentially reaching attachment points between surfaces to remove them. Our work is similar to Krishnan et al. [28], who approximate a long-horizon sequential task as a sequence of sub-tasks each represented by a local reward function learned from Inverse Reinforcement Learning (IRL) [6]; however, prior work assumes demonstrations are optimal and execute sub-tasks in the same order for computational tractability. By contrast, we drop the restriction that the trajectory has to reach attachment points in a specific order, which enables learning from suboptimal demonstrations efficiently.

III. Problem Definition

We model our problem as a Partially-Observable Markov Decision Process (POMDP), which is formulated as a tuple of state space $\mathcal{S}$, action space $\mathcal{A}$, transition probability $\mathcal{T}$: $\mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, observation space Ω, emission probability $\mathcal{O}$: $\mathcal{S} \rightarrow [0, 1]$, reward function R: $\mathcal{S} \rightarrow \mathbb{R}$, discount factor $\gamma \in [0, 1]$ and horizon T [29]. We assume no direct access to the true reward function, nor the true state. Thus, we seek to learn the function R_θ: $\Omega \rightarrow \mathbb{R}$ that approximates the actual reward function and leads to optimal behavior under the true, unobserved reward function. In the surgical robotics domain that we consider, Ω is the space of partial-view point clouds $\mathcal{P} \subseteq \mathbb{R}^3$ of the workspace scene augmented with the task-related robot state s^{robot} such as end-effector position. A demonstration is defined as a trajectory $\tau = (o_1, o_2, ..., o_T)$ consisting of T observations. Note that τ could potentially be a suboptimal demonstration. We denote that a trajectory τ_i is more preferred than τ_j by $\tau_j \prec \tau_i$. To learn the reward function, we assume access to a dataset of trajectories, $D = \{\tau_i\}_{i=1}^{M}$, and access to pairwise preference rankings, $\{(\tau_i, \tau_j, \mathbb{1}_{\{\tau_i \prec \tau_j\}}) : \tau_i, \tau_j \sim D\}$.

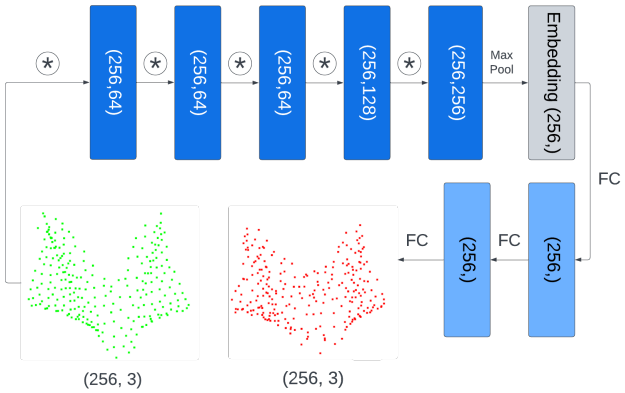


Fig. 2: Our autoencoder takes in the green point cloud and outputs the red reconstructed point cloud. * denotes (RELU ∘ group norm ∘ 1D convolution), FC denotes (ReLU ∘ linear layer) and the tuples denote the shape of the input to each layer. Convolution and group norm are done along the second dimension of the input. Max pooling is done along the first dimension of the input

IV. Method

As summarized in Fig. 1, we train an autoencoder to obtain low-dimensional feature representations of partial-view point clouds. These representations are then used to learn a parameterized reward function from preference rankings over trajectories of observations, and the learned reward function is used to train a policy. We discuss these steps below.

A. Point Cloud Autoencoder

Rather than directly training a reward model from partial-view point clouds, we suggest a more scalable approach that leverages a pre-trained point cloud autoencoder to map the high-dimensional point clouds into a lower-dimensional latent feature representation. We use this low-dimensional latent representation together with the task-related robot state as the input for the learned reward function. Fig. 2 shows the architecture of our point cloud autoencoder. We first downsample the point cloud to 256 points as a pre-processing step. We then pass the partial-view point cloud $P_I \in \mathbb{R}^{256\times3}$ through the encoder ϕ: $\mathcal{P} \rightarrow \mathbb{R}^{256}$ consisting of five 1D convolution layers with non-linearity to get a latent feature vector with dimension 256. Using a fully-connected decoder ψ: $\mathbb{R}^{256} \rightarrow \mathcal{P}$, we decode this latent representation back to a reconstructed output point cloud P_o. Our reconstruction loss function is defined as $L = CD(P_I, P_o) + \lambda * EMD(P_I, P_o)$, where the Chamfer Distance (CD) [30] is the sum of the squared distance of every point to the nearest point the other point cloud, and the Earth Mover's Distance (EMD) [31]

computes the distance between distributions of point clouds by computing the minimum amount of work to transform one point set to another one. CD encourages matching the coarse geometry of the point clouds but not the density distributions of point clouds [30]. A linear combination of CD and EMD encourages matching both large-scale and local geometry of point clouds. We explored several values for the tradeoff constant λ and observed that best results occur when the initial $\lambda * EMD(P_I, P_o)$ loss is approximately equal to one-fifth of the initial $CD(P_I, P_o)$ loss.

$$CD(P_I, P_o) = \sum_{x \in P_I} \min_{y \in P_o} ||x - y||_2^2 + \sum_{y \in P_o} \min_{x \in P_I} ||x - y||_2^2$$

$$EMD(P_I, P_o) = \min_{B: P_I \to P_o} \sum_{x \in P_I} ||x - B(x)||_2$$

B. Preference-Based Reward Learning

We assume a discount factor of $\gamma = 1$ since trajectories have finite horizon T. We also assume that the preferences $\tau_i \prec \tau_j$ positively correlate with $J(\tau_i) < J(\tau_j)$ where $J(\tau) = \sum_{o \in \tau} R(o)$ is the return of a trajectory under the unobserved true reward function. We use Trajectory-ranked Reward Extrapolation (T-REX) [12] to learn a reward function that explains the pairwise preferences over demonstrations and potentially recovers the true reward function. Denote $R_\theta(o)$ as the parameterized learned reward function and define $J_\theta(\tau) = \sum_{o \in \tau} R_\theta(o)$ as the return of a trajectory τ according to learned reward function. The ideal learned reward function should satisfy the constraint $\forall \tau_i, \tau_j \sim \Pi, \tau_i \prec \tau_j \to J_\theta(\tau_i) < J_\theta(\tau_j)$, where Π is the demonstration distribution. Thus we minimize the following loss function to learn the reward function $R_\theta(o)$ that maximizes the likelihood of the preference rankings:

$$L(\theta) = E_{\tau_i, \tau_j \sim \Pi}[\xi(P(J_\theta(\tau_i) < J_\theta(\tau_j)), \tau_i \prec \tau_j)] \quad (1)$$

where ξ is cross-entropy loss and P is a softmax-normalized probability distribution defined as follows:

$$P(J_\theta(\tau_i) < J_\theta(\tau_j)) \approx \frac{\exp \sum_{o \in \tau_j} R_\theta(o)}{\exp \sum_{o \in \tau_i} R_\theta(o) + \exp \sum_{o \in \tau_j} R_\theta(o)}$$

$$L(\theta) = -\sum_{\tau_i \prec \tau_j} log \frac{\exp \sum_{o \in \tau_j} R_\theta(o)}{\exp \sum_{o \in \tau_i} R_\theta(o) + \exp \sum_{o \in \tau_j} R_\theta(o)}.$$

There are several potential ways to obtain trajectories and pairwise preference rankings in practice: (1) the trajectories can come from one or more non-expert human demonstrations, (2) they can be automatically generated by the robot [32] and then used as active queries for the human to compare, and (3) pairwise preferences can be automatically generated by adding noise to an imitation policy [22], [32]. Notably, it has been shown that untrained individuals can generally assess surgical skills rapidly, efficiently, and accurately across different specialties and types of surgeries by watching surgical recordings [33]. Untrained individuals can rapidly provide evaluations on basic robotic surgical dry-laboratory tasks that highly correlate with expert evaluations [34]. Given the learned reward function, a stochastic policy π: $\Omega \times \mathcal{A} \to [0, 1]$ can be learned by maximizing the expected return $E[\sum_{t=1}^{T} \gamma^{t-1} R_\theta(o_t) | \pi]$ using any RL algorithm. Our approach is summarized in Algorithm 1.

Algorithm 1 Preference-based Reward Learning with Partial Observations

1. Collect a random set of partial-view point clouds $D_{AE} = \{p_i\}_{i=1}^{N}$
2. Pre-train autoencoder on D_{AE} with ϕ: $\mathcal{P} \to \mathbb{R}^{256}$ as the encoder
3. Collect random demonstrations consisting of partial-view point cloud embedding concatenated with the task-related robot state
$D = \{\{(\phi(p_t), s_t^{robot})\}_{t=1}^{T}\}_{i=1}^{M}$
4. Collect pairwise preference rankings $D_{rank} = \{(\tau_i, \tau_j, \mathbb{1}_{\{\tau_i \prec \tau_j\}}) : \tau_i, \tau_j \sim D\}$
5. Optimize R_θ by minimizing $L(\theta)$ on D_{rank} (Eq. (1))
6. Find the optimal policy π under R_θ using RL

V. Policy Learning

The observation space contains the task-related robot state s^{robot} so that the robot gets dense reward at every action. We choose the action to be end-effector position $p_{eef} \in \mathbb{R}^3$ instead of end-effector velocity $v_{eef} \in \mathbb{R}^3$ or joint velocity $v_{joint} \in \mathbb{R}^k$ (k is the number of joints).

The benefit of such a design choice is two-fold. First, using p_{eef} as the action makes RL more sample efficient. The task-related robot state added to the observation space should match the robot action. For example, if the action is v_{joint}, then the joint position should be added to the observation space instead of end-effector position. Otherwise, vastly different joint actions can result in the same task-related robot state, which makes learning the policy difficult. Hence, using p_{eef} as the action keeps the dimension of the task-related robot state low, making RL more sample efficient. This action definition also causes larger end-effector dispalcement that may enable more efficient exploration. Second, using p_{eef} as the action is more interpretable, as a trajectory in end-effector position explicitly indicates where the end-effector will reach in the next time step.

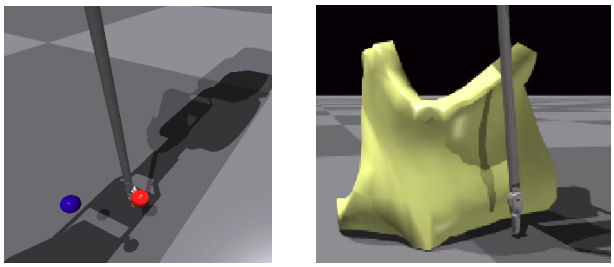
In order to command the robot in parallel in the RL environment, we transform end-effector actions output by the policy into joint velocity actions via a resolved rate controller [35].

VI. Experiments

A. Simulation Experimental Setup

We first apply our method to simulated surgical electrocautery tasks. Simulated experiments are conducted in the Isaac Gym simulation environment [36], using a simulated patient-side manipulator of the da Vinci Research Kit (dVRK) [37] robot. Point clouds of the workspace scenes are obtained via a simulated RGBD camera at a fixed position.

The surgical electrocautery task is modeled as sequentially moving the end-effector to attachment points between tissues and removing them. To simplify training, we assume having



(a) Sphere Task (b) Cutting Task

Fig. 3: Experimental setups with the dVRK surgical robot in the Isaac Gym simulator.

TABLE I: Training data size, where K is the number of trajectory sets, C is the number of trajectories per set, and M is the number of preference rankings sampled with replacement.

Task	K	C	M
Sphere	30	30	14000
Cutting	60	30	14000

the end-effector reach the points is sufficient to remove them due to the limitations of the simulator (although we demonstrate real electrocautery in the physical experiments).

In the first simulated experiment (Sphere Task), the attachment points are represented as two spheres, and no tissues are present, as shown in Fig. 3a. The robot aims to move its end-effector to reach both spheres in any order. The reward function and the policy are trained on an augmented observation space, which concatenates the robot's end-effector's cartesian coordinates and the partial point cloud embedding of the sphere(s) output by the encoder. To simulate electrocautery, the sphere disappears when the end-effector reaches one sphere. Since the learned reward function depends on the observation, a change in the point cloud embedding provides a signal that the end-effector should reach the remaining sphere.

In the second simulation experiment (Cutting Task), we attach a simulated rectangular tissue onto a flat surface through a single attachment point sampled randomly, as shown in Fig. 3b. In order to reveal the attachment point, the tissue is retracted using a deterministic policy. The reward function and the policy are trained on an augmented observation space which is a concatenation of the robot's end-effector's cartesian coordinates and the partial point cloud embedding of the retracted tissue output by the encoder. The goal of the robot is to move its end-effector to the attachment point. The position of the attachment point must be inferred from the deformation of the retracted tissue.

B. Data collection

To simulate suboptimal demonstrations, we programmatically collect a set of trajectories via a suboptimal motion planner. We generated multiple sets of random trajectories, each corresponding to a random configuration of the scene that the robot observes. The training data details, including the number of trajectory sets (K), the number of trajectories per set (C), and the number of preference rankings (M), are summarized in Table I. Note that given m ranked trajectories, we obtain $(m^2 - m)/2$ pairwise preferences. This allows us to obtain a large M from a much smaller number of ranked trajectories. For the Sphere Task, a random configuration of the scene is the random cartesian coordinates of the two spheres. Spheres are sampled along a random horizontal straight line in the 3D space with slope in [-1,1]. For the Cutting Task, a random configuration of the scene is the retracted tissue with a random attachment point. Attachment points are sampled within a (2.5 cm x 5 cm) rectangle in the front half of the (20 cm x 20 cm) tissue closer to the robot. Trajectories are within the robot's workspace, which is a bounding box in the 3D space.

The random trajectories are generated as follows: given a fixed trajectory length and initial end-effector position, we sample the number of attachment point(s) to be reached. At each discrete timestep of the trajectory, we sample which attachment point should be reached. If no attachment point should be reached at this timestep, we sample a random point in the workspace to be reached. Finally, we execute the trajectory to collect a sequence of observations using inverse kinematics to control the end-effector.

For our experiments, we designed a ground-truth (GT) reward for ranking demonstrations to allow us to quantitatively measure how well our method recovers the ideal reward function and to allow us to better compare against baseline approaches. Note that our algorithm never observes the ground truth reward. To ensure the comparability of trajectory pairs using the GT reward function, only trajectories that have the same initial configuration of the scene are paired and ranked. Trajectory pairs are sampled randomly with replacement. The GT reward function is

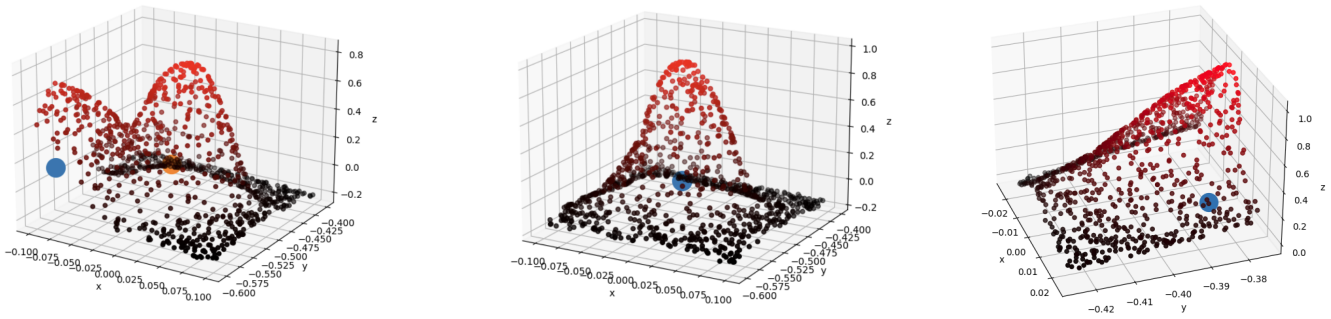
$$R(eef, B) = \max_{b \in B} \frac{1}{||eef - b||_2^2 + \epsilon}$$

where eef is the 3D cartesian coordinates of the end-effector, b is the 3D cartesian coordinates of an attachment point, B is the set of attachment points and ϵ is a small number. We use $\epsilon = 1e-4$ for Sphere Task and $\epsilon = 1e-5$ for Cutting Task.

We collect partial point clouds of random scene configurations in simulation to pre-train the autoencoder. For the Sphere Task, partial point clouds of 71,000 random positions of two spheres were collected. For each random position of the spheres, the data collection is repeated for different permutations of spheres disappearing. For the Cutting Task, partial point clouds are collected for 10,000 configurations of tissues each determined by a random attachment point. Since complex geometry can be reconstructed with just 10,000 random point clouds, the number of partial point clouds for the Sphere Task can be potentially lowered in the future.

C. Policy Learning

The policy is trained using Proximal Policy Optimization (PPO) [38]. We run 450 robots in parallel for efficiency purposes. Our action space is the space of end-effector position so that RL can be more sample efficient and



(a) Sphere Task: heat map of learned reward when two spheres remain

(b) Sphere Task: heat map of learned reward when one sphere remains

(c) Cutting Task: heat map of learned reward

Fig. 4: Visualization: given a specified number of attachment point(s) in the scene, end-effector x-y positions (red points) are sampled on the same horizontal plane of the attachment point(s). The z-value of each red point is the predicted reward given the partial point cloud observation and the coordinates of the corresponding end-effector. Brighter color means higher predicted reward.

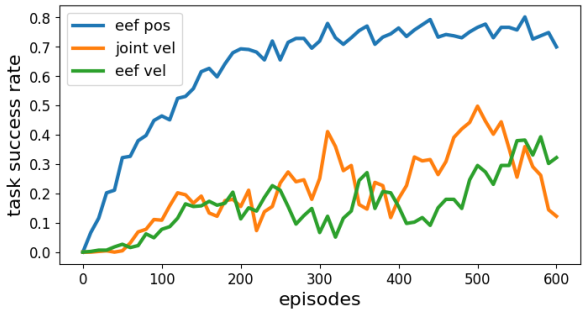


Fig. 5: Learning curves of RL with different action spaces: (blue) end-effector position control, (green) end-effector velocity control, and (orange) joint velocity control. End-effector position control achieves the highest success rate.

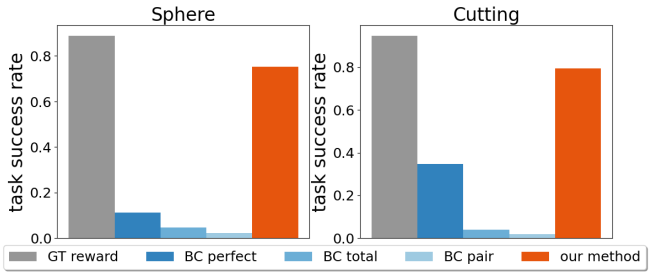


Fig. 6: Baseline comparison in terms of task success rate for both tasks. Our method achieves success rates close to those of the GT reward oracle baseline.

TABLE II: Testing accuracy of the learned reward function trained on decreasing numbers of pairwise preference rankings (from 13050 to 407) sampled without replacement

Task	13050	6525	3262	1631	815	407
Sphere	0.867	0.866	0.859	0.81	0.766	0.721
Cutting	0.769	0.779	0.794	0.774	0.764	0.794

explainable, so policies can be transferred between robots with different embodiments, and to enable easier sim2real transfer. In order to restrict the robot's cartesian action within the workspace, we use a sigmoid function σ to clip the cartesian action along each dimension i of x,y,z as follows: $action \leftarrow min_i + (max_i - min_i)\sigma(action)$ where max_i and min_i are the upper and lower bounds of dimension i.

D. Simulation Experiment Results

For both tasks, Fig. 4 visualizes that our learned reward function captures the intention of the demonstrations well as it assigns high reward to observations where the end-effector position is close to any remaining attachment point(s). Fig. 5 also shows that commanding end-effector position control actions is important for maximizing RL performance.

For both tasks, we compare against a Behavioral Cloning (BC) baseline [39], a standard approach for learning from demonstrations [4] that uses supervised learning to learn a policy that maps from states to actions. *BC pair* denotes the policy trained on the more preferred demonstrations in every pairwise preference, *BC total* denotes the policy trained on the top 20 percent of the suboptimal demonstrations, and *BC perfect* denotes the policy trained on expert demonstrations of the same amount as suboptimal demonstrations. For Sphere Task, Fig. 6 shows that our method achieves close to 80 percent task success rate, upper-bounded by the 85 percent task success rate achieved by the policy trained on the ground-truth (GT) reward. For Cutting Task, Fig. 6 shows that our method achieves 80 percent task success rate, upper-bounded by the almost 90 percent task success rate achieved by the policy trained on GT reward. Fig. 6 shows that our method outperforms BC policies that have access to the same amount of demonstrations: our method yields improvements of 64.13% for Sphere Task and 44.70% for Cutting Task over BC perfect.

To evaluate the sample efficiency of our method, we repeatedly halved the training data size and computed the testing accuracy of the learned reward function as shown in table II. We empirically found that 6,525 pairwise preferences are needed to learn a robust reward function that achieves a task success rate of 80 percent in Sphere Task after policy learning. For Cutting Task, we empirically found

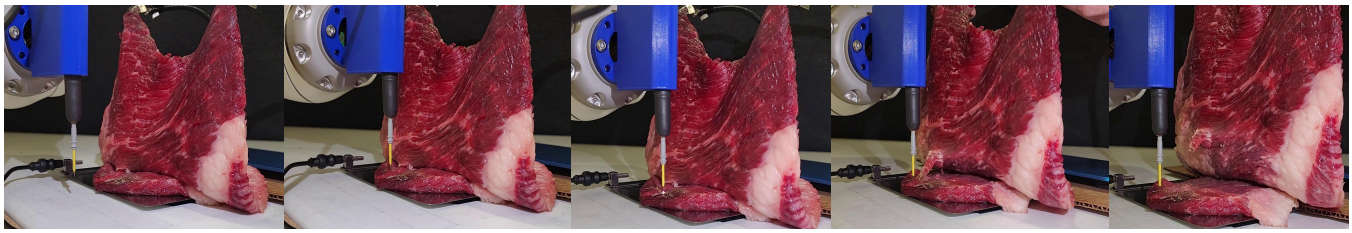


Fig. 7: Sample of successful manipulation sequence in real robot experiment.

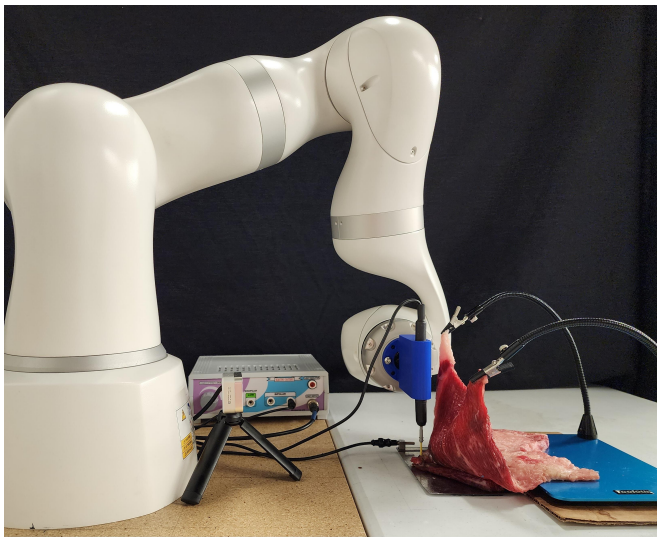


Fig. 8: Experimental setup for electrocautery cutting task

that only 815 pairwise preferences are needed to learn a robust reward function that achieves a task success rate of 80 percent.

E. Real Robot Experimental Setup

A successful execution of the electrocautery policy is shown in Fig. 7 and our full physical experimental setup is illustrated in Fig. 8. We use bovine muscle tissue as an ex vivo tissue evaluation platform. We use one piece of tissue as a flat surface to which another retracted piece of tissue is attached. In each experimental trial, the retracted tissue is attached to a point of the flat surface within a 2.5 cm x 5 cm rectangle. An electrocautery tool is mounted on the end-effector of a KUKA LBR Med Robot. An Intel Realsencse depth camera D405 is employed in our setup for recording the tissue point cloud.

We test whether we can directly transfer the learned reward function, autoencoder, and policy trained in simulation to our real experimental setup. Given the fixed initial end-effector position and the initial point cloud of the scene containing the retracted tissue, we generate 200 end-effector trajectories, each of length 30, using the stochastic learned policy in an open-loop manner. The trajectory with the highest predicted learned reward is executed. We terminate the open-loop trajectory generated by the learned policy when the trajectory converges at the attachment point, and the policy switches to a heuristic cutting motion that oscillates left and right on the attachment point to remove it. To compute convergence of the open loop policy, at each timestep, we compute the component-wise mean and standard deviation of the end-effector positions from the initial timestep to the current timestep. This results in a vector of means and a vector of standard deviations at each timestep. As the end-effector motion from the learned policy converges to an attachment point, we expect the l2 norm of the difference of successive mean vectors and successive standard deviation vectors to decrease. The open-loop trajectory of the learned policy is terminated when these two numbers are lower than a threshold. We found that 0.005 as the threshold for the difference of mean vectors and 0.001 for the threshold on the standard deviation vectors worked well in practice.

F. Real Robot Experiment Results

We conducted seven trials of the experiment, each featuring a different attachment point location. The robot successfully accomplished both reaching the attachment points and executing the cutting task in 5 of the 7 trials. In the other two experiments, the robot end-effector approached very close to the attachment points but ultimately halted prematurely before reaching the desired locations. Upon careful analysis of these particular instances, we identified a common factor: the point clouds associated with these failure cases were out of distribution and consequently poorly reconstructed. This discrepancy in reconstruction adversely affected the quality of the latent embedding used for feature representation, resulting in a suboptimal policy. A plausible cause of this problem is the visual disparity between real tissue and the simulated tissue object used for training data collection.

VII. CONCLUSION

In this paper, we propose a novel preference-based reinforcement learning approach that is well suited for partially observable surgical tasks. Our empricial results in simulation demonstrate that our approach is superior to pure imitation learning and is able to achieve high task success despite only having access to suboptimal demonstrations. We demonstrate that our method achieves 80% task success rate in two simulated surgical electrocautery tasks. We also demonstrate a proof of concept physical surgical electrocautery task, in which our method achieved five successful trials out of seven total trials. Future research includes conducting a user study to evaluate how well non-expert humans rank demonstrations and how sensitive the learned reward is to noisy preference rankings. Since large numbers of offline demonstrations and preferences can be prohibitive, incorporating sample-efficient active reward learning [21], [40], [41] into our approach is also an important future research direction.

REFERENCES

[1] X. Zhang, D. Lin, H. Pforsich, and V. W. Lin, "Physician workforce in the united states of america: forecasting nationwide shortages," *Human resources for health*, vol. 18, no. 1, pp. 1–9, 2020.

[2] A. Attanasio, B. Scaglioni, E. De Momi, P. Fiorini, and P. Valdastri, "Autonomy in surgical robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 651–679, 2021.

[3] M. Ginesi, D. Meli, A. Roberti, N. Sansonetto, and P. Fiorini, "Autonomous task planning and situation awareness in robotic surgery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3144–3150.

[4] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[5] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual review of control, robotics, and autonomous systems*, vol. 3, pp. 297–330, 2020.

[6] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, p. 103500, 2021.

[7] A. Pore, E. Tagliabue, M. Piccinelli, D. Dall'Alba, A. Casals, and P. Fiorini, "Learning from demonstrations for autonomous soft-tissue retraction," in *International Symposium on Medical Robotics (ISMR)*, 2021, pp. 1–7.

[8] Y. Huang, M. Bentley, T. Hermans, and A. Kuntz, "Toward learning context-dependent tasks from demonstration for tendon-driven surgical robots," in *International Symposium on Medical Robotics (ISMR)*, 2021, pp. 1–7.

[9] H. Su, A. Mariani, S. E. Ovur, A. Menciassi, G. Ferrigno, and E. De Momi, "Toward teaching by demonstration for robot-assisted minimally invasive surgery," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 2, pp. 484–494, 2021.

[10] J. W. Kim, C. He, M. Urias, P. Gehlbach, G. D. Hager, I. Iordachita, and M. Kobilarov, "Autonomously navigating a surgical tool inside the eye by learning from demonstration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 7351–7357.

[11] K. L. Schwaner, D. Dall'Alba, P. T. Jensen, P. Fiorini, and T. R. Savarimuthu, "Autonomous needle manipulation for robotic surgical suturing based on skills learned from demonstration," in *IEEE 17th international conference on automation science and engineering (CASE)*, 2021, pp. 235–241.

[12] D. S. Brown, W. Goo, P. Nagarajan, and S. Niekum, "Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations," in *International conference on machine learning*. PMLR, 2019, pp. 783–792.

[13] G. Swamy, S. Choudhury, J. A. Bagnell, and S. Wu, "Of moments and matching: A game-theoretic framework for closing the imitation gap," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 022–10 032.

[14] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2371–2378.

[15] A. Pore, D. Corsi, E. Marchesini, D. Dall'Alba, A. Casals, A. Farinelli, and P. Fiorini, "Safe reinforcement learning using formal verification for tissue retraction in autonomous robotic-assisted surgery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4025–4031.

[16] Z.-Y. Chiu, F. Richter, E. K. Funk, R. K. Orosco, and M. C. Yip, "Bimanual regrasping for suture needles using reinforcement learning for rapid motion planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 7737–7743.

[17] C. Wirth, R. Akrour, G. Neumann, and J. Fürnkranz, "A survey of preference-based reinforcement learning methods," *Journal of Machine Learning Research*, vol. 18, no. 136, pp. 1–46, 2017.

[18] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.

[19] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[20] J. Tien, J. Z.-Y. He, Z. Erickson, A. Dragan, and D. S. Brown, "Causal confusion and reward misidentification in preference-based reward learning," in *The Eleventh International Conference on Learning Representations*, 2022.

[21] D. Shin, A. D. Dragan, and D. S. Brown, "Benchmarks and algorithms for offline preference-based reward learning," *arXiv preprint arXiv:2301.01392*, 2023.

[22] D. S. Brown, W. Goo, and S. Niekum, "Better-than-demonstrator imitation learning via automatically-ranked demonstrations," in *Conference on robot learning*. PMLR, 2020, pp. 330–359.

[23] I. Cordero, "Electrosurgical units–how they work and how to use them safely," *Community eye health*, vol. 28, no. 89, p. 15, 2015.

[24] G. Groot and E. W. Chappell, "Electrocautery used to create incisions does not increase wound infection rates," *The American journal of surgery*, vol. 167, no. 6, pp. 601–603, 1994.

[25] K. C. Un, Y. C. Wang, W. Wu, and G. K. K. Leung, "Systemic progesterone for modulating electrocautery-induced secondary brain injury," *Journal of Clinical Neuroscience*, vol. 20, no. 9, pp. 1329–1330, 2013.

[26] M. L. Morris, R. D. Tucker, T. H. Baron, and L. M. W. K. Song, "Electrosurgery in gastrointestinal endoscopy: principles to practice," *Official journal of the American College of Gastroenterology— ACG*, vol. 104, no. 6, pp. 1563–1574, 2009.

[27] A. Ismail, A. I. Abushouk, A. Elmaraezy, A. Menshawy, E. Menshawy, M. Ismail, E. Samir, A. Khaled, H. Zakarya, A. El-Tonoby *et al.*, "Cutting electrocautery versus scalpel for surgical incisions: a systematic review and meta-analysis," *journal of surgical research*, vol. 220, pp. 147–163, 2017.

[28] S. Krishnan, A. Garg, R. Liaw, B. Thananjeyan, L. Miller, F. T. Pokorny, and K. Goldberg, "Swirl: A sequential windowed inverse reinforcement learning algorithm for robot tasks with delayed rewards," *The international journal of robotics research*, vol. 38, no. 2-3, pp. 126–145, 2019.

[29] M. T. Spaan, "Partially observable markov decision processes," in *Reinforcement learning: State-of-the-art*. Springer, 2012, pp. 387–414.

[30] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin, "Density-aware chamfer distance as a comprehensive metric for point cloud completion," *arXiv preprint arXiv:2111.12702*, 2021.

[31] ——, "Balanced chamfer distance as a comprehensive metric for point cloud completion," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 088–29 100, 2021.

[32] Y. Liu, G. Datta, E. Novoseller, and D. S. Brown, "Efficient preference-based reinforcement learning using learned dynamics models," in *International Conference on Robotics and Automation (ICRA)*, 2023.

[33] R. G. Olsen, M. F. Genét, L. Konge, and F. Bjerrum, "Crowdsourced assessment of surgical skills: A systematic review," *The American Journal of Surgery*, 2022.

[34] L. W. White, T. M. Kowalewski, R. L. Dockter, B. Comstock, B. Hannaford, and T. S. Lendvay, "Crowd-sourced assessment of technical skill: a valid method for discriminating basic robotic surgery skills," *Journal of endourology*, vol. 29, no. 11, pp. 1295–1301, 2015.

[35] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on man-machine systems*, vol. 10, no. 2, pp. 47–53, 1969.

[36] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox, "Gpu-accelerated robotic simulation for distributed reinforcement learning," in *Conference on Robot Learning*. PMLR, 2018, pp. 270–282.

[37] 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 *IEEE international conference on robotics and automation (ICRA)*, 2014, pp. 6434–6439.

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

[39] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in neural information processing systems*, vol. 1, 1988.

[40] D. S. Brown, Y. Cui, and S. Niekum, "Risk-aware active inverse reinforcement learning," in *Conference on Robot Learning*. PMLR, 2018, pp. 362–372.

[41] E. Biyik and M. Palan, "Asking easy questions: A user-friendly approach to active reward learning," in *Proceedings of the 3rd Conference on Robot Learning*, 2019.