Multilateral Surgical Pattern Cutting in 2D Orthotropic Gauze with Deep Reinforcement Learning Policies for Tensioning

Brijen Thananjeyan,[2] Animesh Garg,[2,3] Sanjay Krishnan,[2] Carolyn Chen,[2] Lauren Miller,[2] Ken Goldberg[1,2]

***Abstract*— In the Fundamentals of Laparoscopic Surgery (FLS) standard medical training regimen, the Pattern Cutting task requires residents to demonstrate proficiency by maneuvering two tools, surgical scissors and tissue gripper, to accurately cut a circular pattern on surgical gauze suspended at the corners. Accuracy of cutting depends on tensioning, wherein the gripper pinches a point on the gauze in $\mathbb{R}^3$ and pulls to induce and maintain tension in the material as cutting proceeds. An automated tensioning policy maps the current state of the gauze to output a direction of pulling as an action. The optimal tensioning policy depends on both the choice of pinch point and cutting trajectory. We explore the problem of learning a tensioning policy conditioned on specific cutting trajectories. Every timestep, we allow the gripper to react to the deformation of the gauze and progress of the cutting trajectory with a translation unit vector along an allowable set of directions. As deformation is difficult to analytically model and explicitly observe, we leverage deep reinforcement learning with direct policy search methods to learn tensioning policies using a finite-element simulator and then transfer them to a physical system. We compare the Deep RL tensioning policies with fixed and analytic (opposing the error vector with a fixed pinch point) policies on a set of 17 open and closed curved contours in simulation and 4 patterns in physical experiments with the da Vinci Research Kit (dVRK). Our simulation results suggest that learning to tension with Deep RL can significantly improve performance and robustness to noise and external forces.**

I. Introduction

In robotic surgery, scissors are one of the most effective tools for cutting and removing thin tissue [19]. Deformable materials, such as soft tissue, are most effectively cut when held in tension. This requires a second tool to pinch and tension the material as it is being cut. The optimal direction and magnitude of the tensioning force changes as the cutting progresses, and these forces must adapt to any deformations that occur. Algorithms to automatically generate tensioning policies for a given cutting trajectory can assist both human surgeons [24] and automated surgical cutting procedures to improve the reliability and accuracy of surgical cutting [17, 21].

We formalize *tensioning* as a position constraint on a point called a *pinch point* on the surface of a deformable sheet. The pinch point pins a position on the sheet at a fixed location in $\mathbb{R}^3$ and tension is generated by translating the pinch point in an allowable set of directions. This raises several questions in optimal choice of these time-varying tensions: (1) selecting the pinch point, (2) designing a tensioning policy, (3) and generating the cutting trajectory. While the manipulation of soft bodies is a well-studied field [2, 4, 20, 31], these optimization problems can be quite challenging without closed-form solutions. Furthermore, we would like the solution to be in the form of a closed-loop control policy that adaptively maps the state of the material to tension.

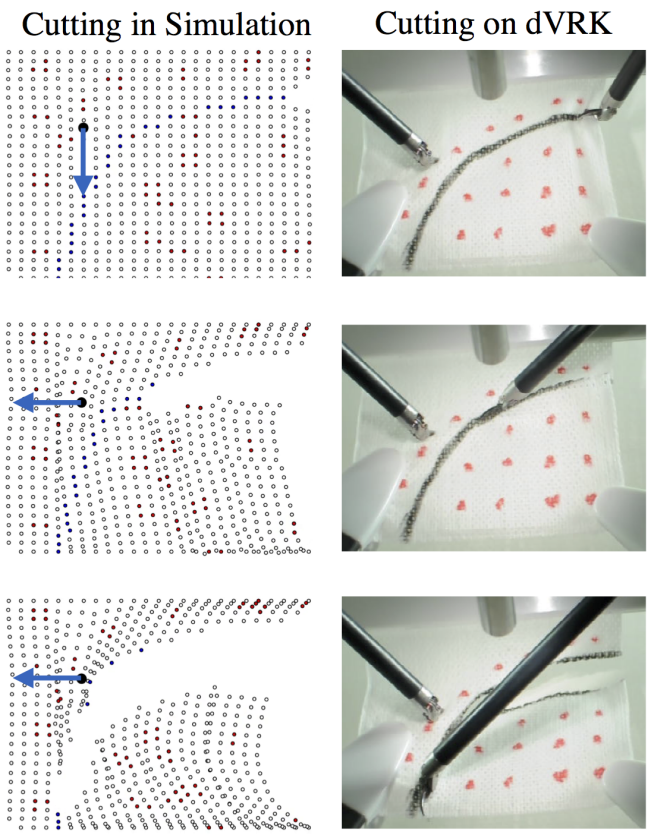

Fig. 1: Surgical Pattern Cutting involves tensioning and cutting deformable tissue phantoms. We propose the use of a simulator (left) and Deep Reinforcement Learning to learn robust tensioning policies that pinch the guaze and adaptively tension it during the Pattern Cutting task. The trained policy is then executed on the physical system (right).

This paper presents a tensioning planner using Deep Reinforcement Learning over a finite-element model to simulate the effect of different tensions and learn a tensioning policy. We focus on tension planning for a generalization of the Pattern Cutting training task that is included in The Fundamentals of Laparoscopic Surgery (FLS) [25] and Fundamental Skills of Robotic Surgery (FSRS) [30] to cut open and closed cutting contours in a 10cm x 10cm sheet of gauze. We focus on tensioning at a single point on the surface of the gauze. The input to the planner is a desired cutting contour. The planner selects a tensioning point and sequence of tensioning actions as the surgical scissors follow

All authors are affiliated to AUTOLAB at UC Berkeley. [1] IEOR, [2] EECS, UC, Berkeley, CA USA; [3] CS, Stanford University, Stanford CA US; {bthananjeyan, animesh.garg, sanjaykrishnan, carolyn.chen, laurenm, goldberg}@berkeley.edu

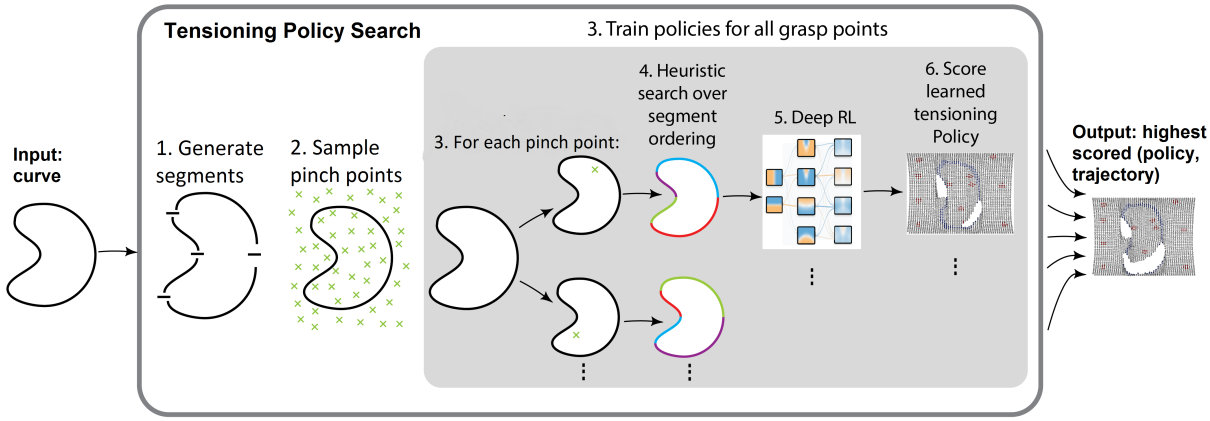

Fig. 2: The algorithm for generating tensioning policies for cutting using Deep RL involves several subcomponents: 1) the input contour is segmented, 2) a set of candidate grasp/pinch locations for tensioning are chosen, 3) search is performed to select the order to cut segments for all candidate pinch points, and 4) tensioning policies are found for each pinch point/segment ordering using Deep RL. The pinch point, cutting trajectory, and policy which have the highest score are then selected. For details on each component, see Section V.

a pre-planned trajectory. The tensioning policy is learned with a state of the art policy gradient algorithm, Trust Region Policy Optimization [27] using a Multi-Layer Perceptron with 2-hidden layers.

Simulation results evaluating robustness suggest that the Deep RL policy performs on average $43.3 \pm 8.6\%$ better than a non-tensioned baseline over a variety of open and closed contours. Our experiments that suggest the policies are not sensitive to bounded changes in gravity, tissue elasticity, and resolution. We also evaluate the learned tensioning policies with physical experiments on the da Vinci Research Kit (dVRK). The gauze is registered to the simulator using colored fiducial points tracked with a computer vision system. We add these visually tracked fiducial points to the system state for Deep RL training to represent the environment.

Summary of Contributions:

1) We propose an algorithm for learning tensioning policies using Deep RL, specifically for Multilateral Surgical Pattern Cutting in 2D Orthotropic Gauze.
2) We describe an implementation of the algorithm and present simulated and physical experimental accuracies and sensitivity results of the proposed procedure using a) an FEM fabric simulator and b) a working dVRK surgical robot.

II. Related Work

A. Deformable Manipulation in Robotic Surgery

Manipulation of deformable materials, particularly cutting, is a challenging area of research interest in robotic surgery [21, 22] as well as in computer graphics and computational geometry [7, 33]. The use of expert demonstrations has been considered in prior work as an alternative to explicit models and simulations when studying and handling deformations in the environment. For example, Van den Berg et al. [32], Osa et al. [23], and Schulman et al. [26] all approached manipulation of suture material using Learning From Demonstrations (LfD), an approach that uses demonstration trajectories to learn how to execute specific tasks. In this paper, we explore the feasibility of a self-learning approach using Deep RL and a simulator to improve control in surgical pattern cutting.

B. Reinforcement Learning for Deformable Manipulation

RL has been a popular control method in robotics when dynamics are unknown or uncertain [16]. There are a few examples of RL applied to deformable object manipulation, e.g. folding [3] and making pancakes [5]. The recent marriage between RL and Neural Networks (Deep RL) opened a number of new opportunities for control in non-linear dynamical systems [18]. An obstacle to applying Deep RL to physical robots is that large amounts of data are required for training, which makes sufficient collection difficult if not impossible using physical robotic systems–leading to our study of simulation-based learning.

C. Simulation-based Training

There has been some recent work using simulators to train policies [1, 8, 9] before applying the policies to the physical system. For example, Frisken et al. [11] discuss a linked volume representation that enables physically realistic modeling of object interactions including collision detection and response, 3D object deformation, and interactive object modification. In this work, we explore a finite-difference model for simulating the deformation and cutting of the gauze [29]. We choose the finite-difference model for its computational efficiency, which is important since we are running $10,000$ trials with RL.

III. Problem Statement and Overview

This section overviews the tensioning problem, and our application of tensioning to a generalization of the pattern cutting task from Fundamentals of Laparoscopic Surgery.

A. Deformable Sheet Cutting

We are given a rectangular planar sheet and a simple algebraic desired cutting contour of bounded curvature, which can be either closed or open. While the contour can intersect the edge of the sheet, it cannot intersect itself. This sheet is fixtured at the four corners. The objective is to cut along this contour and minimize damage to the material not on the contour. A motivating instance of this problem is a

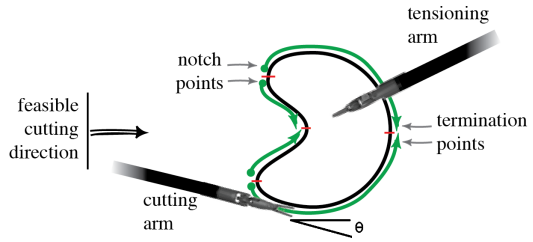

Fig. 3: The dVRK arms have limited approach angles, and cutting can only happen along a vector where the angle θ from the forward direction from the base of the arm is less than or equal to 90 degrees in magnitude, with equality only permitted for isolated points. This requires segmenting contours into multiple sections for cutting. Notch and termination points are locations at the limits of the cutting arm's range of motions, or those orthogonal to the main approach direction of the arm, and the direction of cutting is always in the main approach direction.

generalization of the pattern cutting task from Fundamentals of Laparoscopic Surgery (FLS), which is a standard training regimen for medical students in laparoscopic surgery [25]. A typical pattern cutting task features a 50 mm diameter, 2 mm thick circular pattern marked on a 10×10 cm square of standard surgical gauze suspended by clips as shown in Figure 1.

1) Kinematics of Scissor Cutting: Robotic scissors are a popular tool for cutting thin sheets. However, scissor-based cutting is kinematically constrained. Cutting can only happen in the direction in which the scissors are pointing. If the scissors cannot be rotated in all 360 degrees, it may not be possible to cut a complex contour that cannot be completed in a single, smooth trajectory. For many practical robots this is often the case, e.g., the physical constraints of the da Vinci wrist restrict motion to $\pm90^\circ$ of articulation [14] from a fixed position, yaw, and roll. Therefore, a countour has to be broken into smaller segments, where each segment is cut from a different starting position (Figure 3).

2) Scissor Cutting in Deformable Sheets: Mahvash et al. describe the dynamics of scissor-based cutting of deformable materials in [19]. Essentially, scissors apply a strong local deformation around the blades stretching the material until it tears. The elasticity of the material determines how much of such a force it can resist before tearing. Therefore, such materials are best cut when held in tension–i.e., the material is stretched by a secondary tool. This means that the material's ability to resist the cut is reduced because some of that energy goes into resisting the stretching. In addition to the elasticity effects, tensioning also improves the robustness to deformation when cutting complex contours. As more of the sheet is cut, parts of the contour may change position due to deformation. Tensioning adds an additional degree of freedom to prevent such deformations from drastically changing the position of the contour during execution.

B. Definitions and Notation

Let $G_{x,y,z} \subset \mathbf{R}^3$ be a three-dimensional global coordinate frame. We assume an infinitely thin 2-D sheet that lies on a 2D manifold in $G_{x,y,z}$. We denote the set of points on this sheet as Σ, and $\Sigma^{(G)}$ is the locations of these points in the global frame.

1) Cutting: We assume that one arm of the robot is designated as the cutting arm and the other as the tensioning arm. A cutting contour is a sequence of points C on the surface of the sheet. The cutting arm operates in an open-loop trajectory that attempts to cut along C_0, the position of the cutting contour in the global frame at time zero. Error is measured using the symmetric difference between the desired contour on the sheet and the achieved contour cut. These will be different due to deformation of the sheet during cutting. Let X be the set of points inside a closed intended trajectory and let Y be the set of points inside a closed simulated trajectory. The symmetric difference is then the exclusive-or $A \oplus B$ of the two sets. For open contours, the contours are closed by forming boundaries with the edges of the sheet.

Problem 1. Cut Planning Problem: Assuming that the cutting arm cannot rotate over 90 degrees with respect to the horizontal axis in the direction of entry, problem 1 is to segment the cutting contour into K segments, and identify *notch points* (where the cutting arm should enter the sheet), *termination points* (where the segment is complete), and *order* in which the segments should be cut. We assume the number of segments of the input contour to be less than five and make no special assumptions for segments that require the cutting of a notch for the scissors to enter.

2) Tensioning: Since the cutting is open-loop it cannot account for deformation, and this is why we need tensioning to apply feedback based on the state of the sheet.

Definition 1 (Tensioning): Let $s \in \Sigma$ be called a pinch point. Tensioning is defined as constraining the position of this pinch point to a specific location $u \in G_{x,y,z}$:

$$T = \langle s, u \rangle$$

In this paper, we consider a single pinch point for an entire cutting trajectory. This allows us to define a tensioning policy:

Definition 2 (Tensioning Policy): Let $\Sigma_t^{(G)}$ be the locations of all of the points on the sheet in the global coordinate frame at time t. For a fixed pinch point s, a tensioning policy π_s is a function where $\Delta_u = u_{t+1} - u_t$:

$$\pi : \Sigma^{(G)}(t) \mapsto \Delta_u$$

This naturally leads to two complementary problems.

Problem 2a. Pinch Point Selection Problem: For a fixed tensioning policy π, problem 2a is to find the pinch point s that minimizes the symmetric difference of the desired vs. actual contour subject to kinematic and reachability constraints.

Problem 2b. Tensioning Policy: For a fixed pinch point u, problem 2b is to find a tensioning policy that minimizes the symmetric difference of the desired vs. actual contour. We synchronized the arms' actions by waiting for the tensioning action at fixed time intervals.

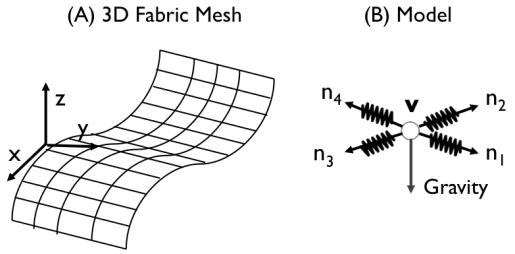

Fig. 4: (A) We simulate the sheet with a finite difference method with a rectangular mesh of vertices v coupled with ideal springs, n_i. (B) The parameters of the simulated sheet are elasticity τ, a $-z$ external gravitational force $f(t)$, and time-constants α, δ that determine the rate at which the sheet reacts to $f(t)$

C. Assumptions

Our planner makes several important assumptions about the nature of the environment and task.

A1. Ideal Spring Deformations: We assume that the physics of the sheet can be approximated using the finite element method. We model the sheet as a discretized two-dimensional array of point masses connected to their neighbors by springs with a fixed spring constant.

A2. State Estimation: We randomly choose fiducial markers to track the state of the sheet in the global frame. We assume that this low-resolution information is sufficient to infer the overall deformation of the cloth.

A3. Quasi-Static Planning: We assume that at each time-step the sheet is in an equilibrium state and therefore do not consider estimation of higher derivatives of the state.

A4. Bounded Cardinal Actions: We assume that controls are bounded $1mm$ movements of the tensioning arm in the $G_{x,y}$ plane.

A5. Zero-Force Cuts: We assume that the cutting arm does not impart forces on the sheet other than what is needed to tear the material.

IV. Simulator Model

We use a finite-element simulator to plan the cutting trajectories and optimize the tensioning policy.

A. Simulating a Deformable Sheet

A planar deformable sheet can be modeled as a rectangular mesh of point masses connected by ideal springs of equal length [6, 12] (Figure 4). Each point mass, a vertex of the mesh, can translate in three dimensions. A constant gravitational force is applied to each vertex, and tensioning is modeled as displacement of a single vertex. These mesh points represent a discretization of Σ and the simulator estimates $\Sigma_t^{(G)}$.

The simulator is initialized with equally spaced mesh points in the $\Sigma_0^{(G)} \in G_{x,y}$ plane. The state is then iteratively updated:

$$\Sigma_{t+1}^{(G)} \leftarrow \mathsf{sim}(\Sigma_t^{(G)})$$

For each $p \in \Sigma$, the updates have the following form:

$$p_{t+1} = \alpha p_t + \delta(p_t - p_{t-1}) - \sum_{(p' \in \text{neighbors} \wedge \neg \text{cut})} \tau(p'_t - p_t) + F_t \quad (1)$$

where τ is a spring constant, $\tau(p'_t - p_t)$ is an idealized spring model of the interaction between vertices, and F_t is an external force applied to the system (e.g. gravity). α and δ are parameters that govern the time-constant of the system and how quickly the sheet reacts to applied forces. Cutting is simulated as removing vertices from the mesh (instantaneous and quasi-static) along the cutting trajectory, i.e., the vertex no longer has an effect on its neighbors. Tensioning is simulated as a position constraint for a chosen pinch point s:

$$s_t = u_t$$

To model standard surgical gauze, we use the following parameters: $\delta = 0.008$, $\alpha = 0.99$, $\tau = 1.0$ and $f(t)$ is gravity. We fit these parameters through trial and error, observing how cutting trajectories deformed in real gauze vs. the simulator. We tuned parameters in the simulator, until the failure modes of the simulated rollouts appeared with similar frequency to the corresponding physical failures. We use 625 vertices (25 x 25) in our experiments to approximate the physical gauze. We found that increasing the fidelity of the simulator beyond 625 vertices quadratically increased the runtime of the simulator but only provided marginal gains, which was experimentally validated in Section VII-B. The simulator was implemented in Python using Cython to write C extensions. This sped up each run of the simulation by a factor of six. A single simulation for cutting a prototypical closed contour takes approximately 4 seconds on an Intel Core i7 desktop computer.

V. Planning and Ordering Cut Segments

Before evaluating tensioning policies, we need to select pinch points for tensioning and plan a cutting trajectory.

A. Pinch Point Planning

To address Problem 2a, we uniformly sample $N = 30$ mesh points randomly from our feasible region of points in Σ to generate a candidate set of pinch points P. This is referred to as **`sample_pinch_points`**() in our algorithm. We define the feasible region of vertices to be the region of Σ that does not result in a collision of the tensioning arm with the cutting arm. We use dynamic programming to generate this set of vertices, and we model the cutting arm as a cylinder with scissors extending in the direction of travel. N is chosen to compromise between the runtime of the full algorithm and the probability of selecting an optimal point. We also include the centroid of the contour $c(t)$ in the set P as a pinch point in addition to the sampled points. We train policies, running the entire pipeline, for each of the candidate pinch points, and select the point that results in the highest overall cutting accuracy.

B. Segment Planning

The kinematic constraints of the da Vinci arms, such as joint limits at the wrist and self collisions, preclude us from completing the cutting task in one smooth trajectory. The physical constraints of the da Vinci wrist restrict motion to $\pm 90^{\circ}$ of articulation [14] from a fixed pitch, yaw, and roll in the forward direction along the base of the arm (Figure 3). For our experiments, we chose the right da Vinci arm to serve as the cutting arm. We segment the contours and identify *notch points* (where the cutting arm enters the sheet) and *termination points* for cutting. We use the methods **gen_segments**(c) and **exhaustive_search**(g,K) for segmentation of a parametric contour and creating an ordering over the segments for executing the cut.

gen_segments(c) – To address problem 1, we find local minima and maxima of the contours to locate a set of candidate segments. We calculate an estimated directional derivative along the trajectory towards the cutting arm. A derivative changing from positive to negative signifies the location of a notch. A negative to positive change marks the end point of a segment. A conceptual illustration of this procedure is shown in Figure 3. Our method assumes no exactly vertical segments exist in the contour.

exhaustive_search(g,K) – Given a complete set of executable segments, the sequence of cuts needs to be specified. We posit that for any given set of segments, there is an optimal ordering such that the simulated cutting error is minimized. For experiments in this paper, we iterate through all possible permutations of cutting ordering for each of the candidate pinch points P, choosing the ordering and subsequently trajectory plan that maximizes the simulation score using a fixed tensioning policy. Although this means $K!$ simulations are required for K registered segments, we assume that K is small for the majority of relevant cases.

VI. Learning the Tensioning Policy

A. Trust Region Policy Optimization

The goal is to learn a policy for the tensioning arm such that the error from the cutting trajectory to the marked contour is minimized. We model the tensioning problem as a Markov Decision Process (MDP) $\mathcal{M}$:

$$\mathcal{M} = \langle S, A, \xi(\cdot,\cdot), R(\cdot,\cdot), T \rangle.$$

where the actions A are $1mm$ movements of the tensioning arm in the x and y directions, and the states S are described below. The action space is tuned so the policy can generate sufficient tension to manipulate the cloth significantly over a few timesteps. The dynamics model ξ is unknown and the time horizon T is fixed. Reward is measured using the symmetric difference between the desired contour and the achieved contour cut. The robot receives 0 reward at all time-steps prior to the last step, and at the last time-step $T-1$ receives the symmetric difference. We do not shape the reward, as symmetric difference is exactly the error metric used for evaluation as well.

The Reinforcement Learning objective is to find a function $\pi_\theta : S \mapsto A$ that optimizes the expected reward:

$$R(\theta) = \mathbf{E}_{(s_t,a_t)\sim\pi_\theta}[\sum_{t=0}^{T} R(s,a)] \tag{2}$$

To optimize θ, we leverage the TRPO implementation [27] in Rllab [10]. TRPO is a policy gradient method that can effectively optimize neural networks despite noisy estimates of the gradient. Since $R(\theta)$ is a stochastic quantity, TRPO prevents excessive oscillation during optimization. We use a neural network to parametrize the policy π_θ, which maps an observation vector to a discrete action. A two 32x32 Hidden Layer Multi-Layer Perceptron is used to represent this mapping. Since neural networks are differentiable, we can optimize the quantity $R(\theta)$. The convergence of the policy takes about 5 minutes on an Intel core i7 desktop and for computational reasons, the experiments presented here are trained up to 25 iterations.

B. Choice of State Space

Our experiments require a state space that is observable for RL in both the simulator and in the real robot. The state space is a tuple consisting of the time index of the trajectory t, the displacement vector from the original pinch point u_t, and the location $x_i \in \mathbb{R}^3$ of fiducial points chosen randomly on the surface of the sheet. In all simulation-only experiments we use 12 fiducial points and for robot experiments we use between 8-12 fiducial points. This is a sample-based approximation of tracking $\Sigma_t^{(G)}$. When a point is occluded by the robot arm or cloth deformation, its last visible position is used until it is visible again. The policy gradient algorithm learns a state-feedback tensioning policy as a function of these states.

VII. Simulation Experiments

In the first set of experiments, we compare Deep RL to alternative tensioning approaches. We first evaluate the techniques in terms of performance by calculating the symmetric difference between the target pattern and actual cut. Then, we compare the techniques in terms of robustness by tuning the techniques for one simulated parameter setting and then applying them to perturbations.

A. Cutting Accuracy

We manually drew 17 different closed and open contour to evaluation in simulation, as illustrated in Table I. For each of the contours, we evaluated four different tensioning methods:

- *Not Tensioned:* Single-arm cutting is performed with no assisted tensioning other than the stationary corner clips that fix the sheet in place.
- *Fixed Tensioning:* The material is pinched at a single point with the gripper arm (with corner points still fixed in place), but no directional tensioning is applied. We simulate area contact by pinching a circular disc.
- *Analytic Tensioning:* Tensioning is proportional to the direction and magnitude of the error in the 3D position of the cutting tool and the closest point on the desired

TABLE I: **Evaluation of Deep RL**: For the 17 contours shown, we evaluate the three tensioning policies described in Section VII-A. We measure and report performance in terms of relative percentage improvement in symmetric difference over a baseline of no tensioning for the tensioning trials. The 95% confidence interval for 10 simulated trials is shown for fixed, analytic, and Deep RL tensioning, while the mean absolute symmetric difference error is reported for the no-tensioning baseline experiments. The data suggest that Deep RL performs significantly better in comparison to the fixed and analytic baseline. The corresponding pinch points used for fixed and Deep RL are indicated in red. The analytic pinch point is the centroid of the shape.

Shape	Tensioning Method			
	No-Tension	**Fixed**	**Analytic**	**Deep RL**
1	17.4	-20.69±4.44	-149.43±10.24	**64.37±5.77**
2	22.5	32.44±1.74	-117.78±0.00	**55.11±7.40**
3	23.2	-21.12±5.41	18.10±1.78	**38.36±9.43**
4	102.9	7.48±0.62	30.42±1.45	**36.15±3.97**
5	41.1	0.49±7.99	9.98±0.00	**52.31±7.26**
6	42.0	**55.00±1.77**	11.43±1.52	45.95±6.60
7	40.2	22.64±1.14	3.73±1.46	**33.83±3.99**
8	40.0	-1.25±0.82	1.75±2.83	**35.50±4.67**
9	66.6	2.85±2.60	**34.68±2.25**	28.38±4.98
10	63.6	25.63±2.98	20.60±3.60	**41.35±9.90**
11	73.6	2.31±1.66	24.32±0.98	**56.11±8.99**
12	79.3	22.82±4.51	55.49±0.38	**63.56±3.61**
13	94.3	3.29±2.05	27.15±0.32	**34.04±5.87**
14	71.7	3.07±7.15	-2.51±0.61	**39.89±8.20**
15	178.7	74.71±1.22	80.75±0.88	**81.25±1.38**
16	114.6	-8.03±2.16	**31.06±0.62**	29.06±8.81
17	74.8	10.29±2.08	**28.34±2.07**	0.80±7.03
Mean (%)		13.54±9.84	9.70±21.96	**43.30±8.61**

contour. The gain was hand-tuned on randomly-chosen contours and is fixed to 0.01.

- *Deep RL* This policy is described in detail in Section VI. A separate policy is trained for each shape, this requires 20 iterations of TRPO in the simulator.

For the Deep RL, the pinch point used was chosen by the algorithm described in Section V and Section VI. For the analytic tensioning model, the centroid of the contour is used as the pinch point. For the fixed policy, the point was chosen by random search over the feasible set of points. Performance is measured using the symmetric difference between the desired contour and the actual cut.

The averaged symmetric difference scores for each instance and tensioning method are reported in Table I. The success of the different tensioning algorithms are presented as the percentage of improvement in symmetric difference score over the non-tensioned baseline. The average of the scores over all 17 contours are also included in the table. For the selected set of contours, Deep RL achieves the best average relative improvement of 43.30%, the analytical method 9.70%, and the fixed approach 13.54%.

B. Robustness

One concern with Deep RL is that it could overfit to the simulation environment and would not be as robust as the alternatives. We evaluate the robustness of Deep RL by training it with particular simulation parameters and then testing it in an alternative parameter setting.

1. Robustness to Resolution We first evaluate the learned policy's sensitivity to the resolution of the simulator. We trained four different policies, each on a mesh composed of different number of vertices (100, 400, 625, 2500). These experiments were performed on shape 1, as illustrated in Table I. We chose resolution parameters to cover a large range of fidelity while maintaining the geometry of the cloth. We used each of the four policies to cut meshes with six different resolutions for ten rollouts each. Results are presented in Figure 5 as the absolute symmetric difference error (in number of vertices) per each trained policy when simulated on a mesh of x points. Our results indicate that policies trained on lower resolution simulation models still result in comparable performance up to a very low resolution (400 vertices). This suggests that increasing the fidelity of the simulation environment provides diminishing returns for a quadratically increasing runtime. We use this as justification to limit the fidelity of our simulator for other experiments.

2. Robustness to Process Noise Deep RL was trained in a noise-free simulated environment. Then, all of the approaches were applied to shape 9 (see Table I) with increasing levels of Gaussian noise, i.e. $N(0,k)$ for $k \in \sigma[0,1]$, added to the mesh vertex position update (Eq. (1)). We ran three trials for each policy at 10 different values of σ. Figure 5 depicts the median relative percentage scores in terms of symmetric difference for fixed tensioning, analytic tensioning, and deep RL tensioning, over no-tensioning. We noticed that Deep RL is at least as robust as the other approaches.

3. Robustness to Model Error We vary the magnitude of the external force applied to each vertex in the $-z$-direction. Deep RL is trained on $f(t) = 2500$ in (1). All four policies are tested on varying magnitudes of $f(t)$ on shape 9, for 20 trials. We present results of this experiment in Figure 5. When external forces are greater than this reference value, the Deep RL tensioning policy performs consistently better than the other policies. Both the Deep RL policy and the analytic policy perform worse than no tensioning and fixed tensioning for low force values. Intuitively, as the downward vertical force (gravity) tends toward zero, the gauze behaves more like a rigid sheet than a deformable material, so a simpler

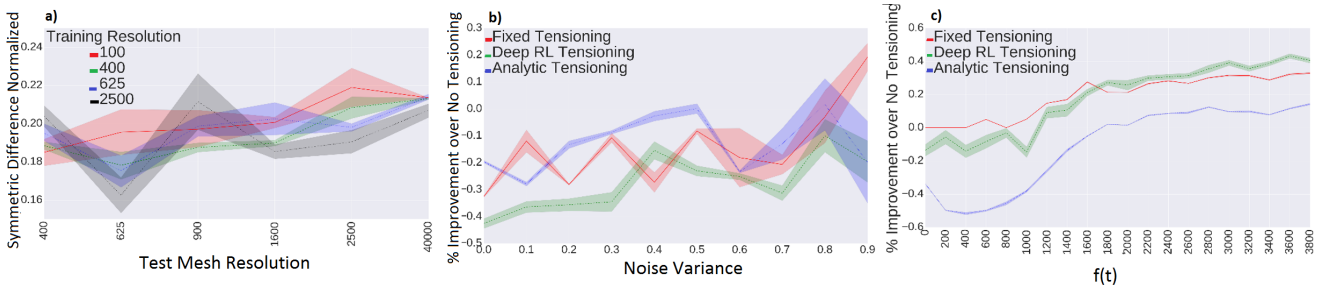

Fig. 5: a) **Robustness to resolution**: Deep RL was trained for four different mesh resolutions, and evaluated on six different resolutions. The symmetric difference error (calculated in units of vertices) is normalized by the test environment resolution. Deep RL is relatively robust to changes in resolution but the performance suffers as the difference between the mesh resolution used for training and testing increases. b) **Robustness to Process Noise**: Deep RL was trained on a noise-free mesh and evaluated with different levels of Gaussian noise added to the process model (Equation (1)). We plot the improvement of the fixed, analytic, and Deep RL tensioning over no tensioning. Deep RL is at least as robust as the alternatives to noise. c) **Robustness to Model Error**: We evaluate the sensitivity of the policies to varying levels of an unmodeled external force in Equation (1). For shape 9, Deep RL is trained using $f(t) = 2500$ indicated by the red dashed line. The mean and variance for twenty trials is represented as a percentage of improvement over no tensioning. The variance for deterministic policies is small in this experiment because the simulator is deterministic as well.

policy will work better, and the analytic and Deep RL policies are overcompensating to error.

VIII. Physical Experiments

A. dVRK: Hardware and Software

We use the Intuitive Surgical da Vinci Research Kit (dVRK) surgical robot assistant, as in [13, 21, 28]. We interface with the dVRK using open-source electronics and software developed by [15]. The software system is integrated with ROS and allows direct robot pose control. We use the standard laparoscopic stereo camera for tracking fiducials on the surgical gauze. We equip the right arm with the curved scissors tooltip for cutting and the left arm with the needle driver tooltip for pinching and tensioning. The arms are located on either side of the surgical gauze to maximize the workspace for each without collision.

B. Physical Evaluation of Deep RL

We show the physical results on 4 contours using Fixed Tensioning and Deep RL using the symmetric difference as the evaluation metric in Table II. The tensioning policy for Deep RL was derived from simulation experiments by registering the physical gauze to a sheet in the simulator environment. In this set of experiments, we used fixed tensioning as the baseline. The no tensioning policy frequently failed in the physical experiment so we excluded that from the results table for a fair baseline comparison. We were unable to evaluate the analytic method in the physical experiments because the state-estimation needed for feedback control was not possible outside of the simulated environment.

Figure 1 also illustrates the results in the case of a circular contour. We do not attempt analytic tensioning since it requires real-time tracking of the pattern to estimate local error. We observed that 3 out of 4 of the Deep RL experiments performed better than the fixed tensioning policy with respect to the symmetric difference of the cut and ideal trajectories. This is the same objective the trained policy was designed to minimize in simulation. A weakness of the policy was failure to address discrete failure modes, including discrete

TABLE II: **dVRK Physical Experiments**: This table compares the relative percentage improvement in terms of symmetric difference to a baseline of fixed tensioning to Deep RL in experiments performed on the dVRK robot. The black dot indicates the pinch point of the gripper arm.

	1	2	3	4
Shape				
Deep RL	**118.63 %**	**36.77 %**	**75.33 %**	-44.59

failure modes induced by the policy itself. Failure as a result of the scissors' entanglement in the gauze occurred in both experimental groups. The active manipulation and tensioning of Deep RL also caused increased deformation of the cloth which occasionally directly resulted in entanglement as well. We did not optimize our policy to minimize these discrete failures, and our simulator does not model the robot with acceptable fidelity to do so. To address this, we would require very accurate models of the robot arms and tools.

IX. Future Work

This paper introduces tensioning policies for Multilateral Surgical Pattern Cutting in 2D Orthotropic Gauze and an algorithm for learning such tensioning policies using Deep Reinforcement Learning. We report accuracy and sensitivity results with an implementation of the algorithm using 1) an FEM fabric simulator and 2) a working dVRK surgical robot.

In future work, we will explore more complex tensioning policies where the pinch point can change during execution and where more than 4 tensioning directions are possible. We will explore 1) alternate contour segmenting methods, 2) segment ordering methods, and 3) methods for selecting pinch points, all of which would reduce runtime complexity. We will also explore methods to avoid collisions between the arms and other obstacles and how this approach can be generalized to cutting contours on more complex or moving surfaces and manifolds for possible application to anastomosis.

ACKNOWLEDGMENT

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the AMP Lab, BAIR, and the CITRIS "People and Robots" (CPAR) Initiative: http://robotics.citris-uc.org in affiliation with UC Berkeley's Center for Automation and Learning for Medical Robotics (Cal-MR). The authors were supported in part by the U.S. National Science Foundation under NRI Award IIS-1227536: Multilateral Manipulation by Human-Robot Collaborative Systems, and by Google, UC Berkeley's Algorithms, Machines, and People Lab, and by a major equipment grant from Intuitive Surgical. We thank Jeff Mahler, Aimee Goncalves, Stephen McKinley, Daniel Seita, Richard Liaw, Lisa Jian, Adam Beckman, and David Tseng for their extensive feedback on this manuscript.

REFERENCES

[1] P. Abbeel, M. Quigley, and A. Y. Ng, "Using inaccurate models in reinforcement learning", in *Proceedings of the 23rd international conference on Machine learning*, ACM, 2006, pp. 1–8 (cit. on p. 2).

[2] R. Alterovitz and K. Goldberg, *Motion Planning in Medicine: Optimization and Simulation Algorithms for Image-guided Procedures*. Springer, 2008 (cit. on p. 1).

[3] B. Balaguer and S. Carpin, "Combining imitation and reinforcement learning to fold deformable planar objects", in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2011, pp. 1405–1412 (cit. on p. 2).

[4] R. A. Beasley, "Medical robots: current systems and research directions", *Journal of Robotics*, 2012 (cit. on p. 1).

[5] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mösenlechner, D. Pangercic, T. Rühr, and M. Tenorth, "Robotic roommates making pancakes", in *Humanoids*, IEEE, 2011, pp. 529–536 (cit. on p. 2).

[6] A. Brooks, *A tearable cloth simulation using vertlet integration*, `https://github.com/Dissimulate/Tearable-Cloth`, 2016 (cit. on p. 4).

[7] N. Chentanez, R. Alterovitz, D. Ritchie, L. Cho, K. Hauser, K. Goldberg, J. R. Shewchuk, and J. F. O'Brien, "Interactive simulation of surgical needle insertion and steering", *ACM Transactions on Graphics*, vol. 28, no. 3, 2009 (cit. on p. 2).

[8] M. Cutler and J. P. How, "Autonomous drifting using simulation-aided reinforcement learning", in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016 (cit. on p. 2).

[9] M. Cutler, T. J. Walsh, and J. P. How, "Real-world reinforcement learning via multifidelity simulators", *IEEE Transactions on Robotics*, 2015 (cit. on p. 2).

[10] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continous control", in *ICML*, 2016 (cit. on p. 5).

[11] S. F. Frisken-Gibson, "Using linked volumes to model object collisions, deformation, cutting, carving, and joining", *IEEE transactions on visualization and computer graphics*, 1999 (cit. on p. 2).

[12] S. Gale and W. J. Lewis, "Patterning of tensile fabric structures with a discrete element model using dynamic relaxation", *Computers & Structures*, 2016 (cit. on p. 4).

[13] A. Garg, S. Sen, R. Kapadia, Y. Jen, S. McKinley, L. Miller, and K. Goldberg, "Tumor localization using automated palpation with gaussian process adaptive sampling", in *CASE*, 2016 (cit. on p. 7).

[14] (2016). EndoWrist Instruments, Intuitive Surgical (cit. on pp. 3, 5).

[15] P. Kazanzides, Z. Chen, A. Deguet, G. Fischer, R. Taylor, and S. DiMaio, "An open-source research kit for the da vinci surgical system", in *ICRA*, 2014 (cit. on p. 7).

[16] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey", *IJRR*, p. 0 278 364 913 495 721, 2013 (cit. on p. 2).

[17] 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.", in *WAFR*, 2016 (cit. on p. 1).

[18] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies", *ArXiv preprint arXiv:1504.00702*, 2015 (cit. on p. 2).

[19] M. Mahvash, L. Voo, D. Kim, K. Jeung, J. Wainer, and A. M. Okamura, "Modeling the forces of cutting with scissors", *IEEE transactions on bio-medical engineering*, vol. 55.3, pp. 848–856, 2008 (cit. on pp. 1, 3).

[20] G. Moustris, S. Hiridis, K. Deliparaschos, and K. Konstantinidis, "Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature", *Int. Journal of Medical Robotics and Computer Assisted Surgery*, vol. 7, no. 4, pp. 375–392, 2011 (cit. on p. 1).

[21] 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 *ICRA*, IEEE, 2015, pp. 1202–1209 (cit. on pp. 1, 2, 7).

[22] H.-W. Nienhuys and A. F. van der Stappen, "A surgery simulation supporting cuts and finite element deformation", in *MICCAI*, 2001 (cit. on p. 2).

[23] T. Osa, N. Sugita, and M. Mamoru, "Online trajectory planning in dynamic environments for surgical task automation", in *RSS*, 2014 (cit. on p. 2).

[24] C. E. Reiley, E. Plaku, and G. D. Hager, "Motion generation of robotic surgical tasks: learning from expert demonstrations", in *EMBC*, IEEE, 2010, pp. 967–970 (cit. on p. 1).

[25] E. Ritter and D. Scott, "Design of a proficiency-based skills training curriculum for the fundamentals of laparoscopic surgery", *Surgical Innovation*, 2007 (cit. on pp. 1, 3).

[26] 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 *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013, pp. 4111–4117 (cit. on p. 2).

[27] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust region policy optimization", *ICML*, 2015 (cit. on pp. 2, 5).

[28] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, and K. Goldberg, "Automating multiple-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization", in *ICRA*, 2016 (cit. on p. 7).

[29] J. E. Shigley, "Mechanical engineering design", 1972 (cit. on p. 2).

[30] A. P. Stegemann, K. Ahmed, J. R. Syed, S. Rehman, K. Ghani, R. Autorino, M. Sharif, A. Rao, Y. Shi, G. E. Wilding, *et al.*, "Fundamental skills of robotic surgery: A multi-institutional randomized controlled trial for validation of a simulation-based curriculum", *Urology*, 2013 (cit. on p. 1).

[31] R. Taylor, A. Menciassi, G. Fichtinger, and P. Dario, "Medical robotics and computer-integrated surgery", *Springer Handbook of Robotics*, pp. 1199–1222, 2008 (cit. on p. 1).

[32] J. Van Den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations", in *ICRA*, 2010 (cit. on p. 2).

[33] H. Zhang, S. Payandeh, and J. Dill, "On cutting and dissection of virtual deformable objects", in *ICRA*, 2004 (cit. on p. 2).