

# JIGGLE: An Active Sensing Framework for Boundary Parameters Estimation in Deformable Surgical Environments

Nikhil Uday Shinde\*, Xiao Liang\*, Fei Liu , Yutong Zhang, Florian Richter, Sylvia Herbert and Michael C. Yip

\* These authors contributed equally to this work

University of California San Diego

Email: {nshinde, x5liang, f4liu, yuz049, frichter, sherbert, yip}@ucsd.edu

**Abstract**—Surgical automation can improve the accessibility and consistency of life-saving procedures. Most surgeries require separating layers of tissue to access the surgical site, and suturing to re-attach incisions. These tasks involve deformable manipulation to safely identify and alter tissue attachment (boundary) topology. Due to poor visual acuity and frequent occlusions, surgeons tend to carefully manipulate the tissue in ways that enable inference of the tissue’s attachment points without causing unsafe tearing. In a similar fashion, we propose JIGGLE, a framework for estimation and interactive sensing of unknown boundary parameters in deformable surgical environments. This framework has two key components: (1) a probabilistic estimation to identify the current attachment points, achieved by integrating a differentiable soft-body simulator with an extended Kalman filter (EKF), and (2) an optimization-based active control pipeline that generates actions to maximize information gain of the tissue attachments, while simultaneously minimizing safety costs. The robustness of our estimation approach is demonstrated through experiments with real animal tissue, where we infer sutured attachment points using stereo endoscope observations. We also demonstrate the capabilities of our method in handling complex topological changes such as cutting and suturing.

## I. INTRODUCTION

Surgical automation has the potential to improve the accessibility of life-saving procedures in under-served communities. Automation can help increase consistency between patient outcomes and alleviate the load of routine procedures from a burdened healthcare system. In recent years, the robotics community has made large strides in surgical automation through the development of dVRK [15], works like STAR [40], and improvements in the automation of tasks such as deformable objects manipulation [13, 32, 46], suturing [4, 36], debridement [17], blood suction [34], cutting [47] and dissection [8].

Deformable manipulation is a significant component of all these surgical tasks. In fact, large portions of a surgical procedures (and occasionally, its majority) involve safely cutting tissue in order to detach regions and access the site of the surgery, as well as suturing tissue back together. These actions actively deform and change the structure of the surgical scene. A key step towards realizing surgical autonomy is enabling robots to understand and track these changing structures. Many

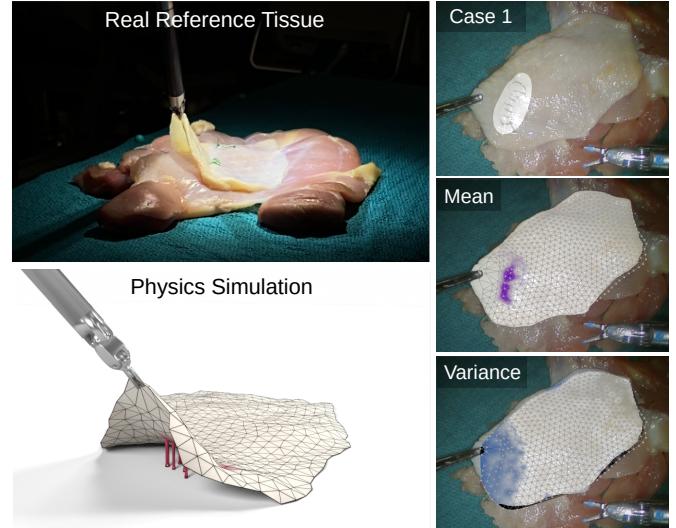


Fig. 1. JIGGLE conducts probabilistic estimation of soft tissue attachment points from image data and manipulation of the tissue. In our real-world ex-vivo experiments, a surgical tool manipulates chicken skin that is attached to its environment with sutures, and we use a stereo endoscope as the observation to estimate the boundary parameters in a probabilistic fashion. The estimated boundary, i.e. the suture locations, is shown in purple. Since the boundary parameters are estimated probabilistically, a corresponding confidence metric can be found and is shown in blue. Note that the confidence is only high near the grasping location because that is where the deformation is applied.

previous works have aimed to take this step solely through 3D scene reconstruction [20, 23, 27, 48]. However, these works are insufficient to explore safe interaction with the environment as they do not provide an estimate of how underlying tissue structures are being deformed or provide an understanding of how aggressively the tissue is being manipulated.

In order to move towards safer surgical autonomy, an interactive approach to track the tissue and its structure is desirable because it can lead to more intelligent tissue control. Previous approaches like [2] that do consider the joint problem of estimation and active sensing are limited in the resolution of attachment regions they can detect. Safety is not considered, and behaviors are limited to hand-tuned motion primitives. Additionally, previous works do not show their results on real tissue data and fail to consider topological

changes like cutting and suturing.

In this work, our goal is to realize an interactive approach to estimating, manipulation, and tracking of a deformable 3D thin-shell tissue in such a way that we can reason about its structure and apply more intelligent, safer tissue control. Specifically, we propose the Joint Interactive Guided Gaussian Likelihood Estimation (JIGGLE) method: a novel active sensing framework for estimating boundary attachment points in deformable surgical environments. We leverage a differentiable physics simulator to develop an efficient probabilistic estimation framework for deformable environments with a high degree of freedom (DOF). Our estimation framework relies solely on stereo camera observations which is typically the only available feedback modality for interactions with the environment. We demonstrate the robustness of this estimation in real tissue manipulation experiments, as shown in Fig. 1. We take advantage of the probabilistic nature of the estimation framework to build an active sensing pipeline that selects actions to maximize information gain while minimizing a safety cost to avoid tissue tearing. We showcase the complete JIGGLE framework in simulations involving topological changes like cutting and suturing that actively change the underlying attachment points. We summarize our contributions as the following:

- An online probabilistic estimation framework to track attachment points in high DOF, deformable environments
- An active sensing algorithm with fine control to maximize information gain while respecting safety constraints
- Demonstrating our framework with topological changes like suturing and cutting.
- Real-world experiments on animal tissue showcasing our robust estimation framework and simulation experiments demonstrating and analyzing the performance of our estimation and active sensing methods.

## II. RELATED WORKS

**Active Sensing.** In robotics, active sensing involves selecting appropriate actions for a robot to maximize information gain while minimizing costs [29]. An active sensing framework contains an estimator (e.g. a Bayesian filter) that can estimate the system’s state under uncertainty. Using this state estimation, an objective relating to information gain is maximized. Information theoretic objectives can be specified in terms of entropy [9], conditional entropy [35], and mutual information [14, 1]. To optimize information gain, frameworks such as trajectory optimization [30] or partially observable Markov decision processes (POMDPs) [42] can be applied to determine the next robot actions. Previously, these techniques have been applied extensively to simultaneous localization and mapping (SLAM) [19, 1], multi-agent tasks [38], and robot manipulation [39]. Another application of active sensing is tissue stiffness discovery for tumor localization using palpation. Previous work on this application focused on guiding the sampling process of Gaussian processes (GPs) to represent variable tissue stiffness [50, 7, 37].

**Differentiable Physics Simulator.** Differentiable simulators enable easy calculation of multiple orders of derivatives of outputs. This characteristic is especially useful in robotic applications with deformable objects as it enables efficient optimization for manipulation and control. The authors of [5, 21] built a differentiable simulation based on projective dynamics (PD) and showed successes in parameter identification of soft bodies and optimal control tasks involving cloth-like objects. Position-based dynamics (PBD) [28, 31] is another favored technique, given its speed and flexibility. It has been used for scenarios such as the manipulation of deformable linear objects (DLOs) [26], safe manipulation of cloth [51], and material parameter optimization [41, 3]. Other popular simulation frameworks, such as the finite element method (FEM) and the material point method (MPM), have also been applied widely in robotic tasks [10, 6, 12]. The surgical robotics community has recently begun researching the use of differentiable simulators in complex surgical scenarios. Specifically, [25, 22] identifies and reduces the gap between real and simulated tissue via online parameter estimation.

In this work, we leverage a differentiable extended PBD (XPBD) simulator. We formulate a new type of geometric constraint that represents the attachment of a tissue to its environment. We rely on the simulator’s differentiability to perform gradient-based updates on the constraints’ parameters.

**Boundary Condition Identification.** Besides stiffness estimation, understanding attachment points and boundary parameters has also been explored. In [33], the authors use deformable registration of a simulation to CT images to determine boundary parameters from the forces calculated on the simulated mesh. The authors in [49] formulates boundary conditions as virtual springs and estimate them using an interior point solver, but do not consider active sensing. Other approaches take a fully perception-based approach to understanding boundaries in surgical environments. Semantic segmentation based approaches like [23] estimate visible boundaries directly from endoscopic data. For attachments that are not directly visible, [44, 45], developed a deep learning based method to directly predict attachment points of soft tissues. However, they fail to consider active sensing; instead, they take a deterministic single-shot approach that fails to consider probabilistic estimation. This confidence information is crucial for tracking boundary estimates and understanding information gain, which is vital to active sensing. [2] takes initial steps in solving the joint problem of estimation and active sensing. Their estimation approach relies on a coarsely discretized set of possible attachment points with binary connections to the environment. This assumption inhibits their method from scaling to higher DOF models and capturing complex attachment regions. Their active sensing approach relies on an exhaustive evaluation of pre-defined motion primitives. Hand-designing motion primitives is not generalizable and incapable of generating new complex deformations required for identification. Additionally, they do not account for safety in their approach. We compare against their active sensing approach in our evaluations. Finally, that work fails to consider

topological changes like cutting and suturing.

### III. PROBLEM STATEMENT

In this paper, we consider a deformable thin shell layer  $\mathcal{T}$  in an environment in  $\mathbb{R}^3$ .  $\mathcal{T}$  represents a tissue in the context of surgery such as skins and membranes or may be used to represent deformable objects like cloth for household tasks. In this paper, we will refer to the thin shell as a tissue. We assume the tissue  $\mathcal{T}$  has non-overlapping subsets  $\mathcal{T}_f$  and  $\mathcal{T}_b$  such that  $\mathcal{T} = \mathcal{T}_f \cup \mathcal{T}_b$ .  $\mathcal{T}_f$  represents the regions of the tissue that are free to move.  $\mathcal{T}_b$  represents regions whose movement is constrained by the environment (i.e. boundary constraints). In a surgical environment, these constraints may be the result of natural tissue connections, such as fascia or wound closures. We refer to these boundary constraints on the tissue as *attachment points*, and we assume their strength can vary spatially across the tissue. In addition, the attachment points can change over time due to actions such as cutting and suturing, that actively alter the tissue's attachments to the environment, during surgery. We assume that the tissue is manipulated using a position-controlled robot end-effector. The robot grasps and directly moves a sub-region of the free tissue surface  $\mathcal{T}_g \subset \mathcal{T}_f$ .

Without loss of generality, we use a thin shell mesh to discretely represent the tissue,  $\mathcal{T}$ . This mesh is composed of particles that are locally connected by edges. The state of the tissue at time  $t$  is specified by its  $n$  particle positions  $\mathbf{x}_t = [\mathbf{x}_t^1, \dots, \mathbf{x}_t^n]^\top$ , where  $\mathbf{x}_t^i \in \mathbb{R}^3, \forall i \in [1, \dots, n]$  specifies the position of particle  $i$ . We denote the position-based grasp control of  $\mathcal{T}$  as  $u_t \in \mathbb{R}^3$ .  $\mathbf{b}_t = [b_t^1, b_t^2, \dots, b_t^n]^\top \in \mathbb{R}^{n \times 1}$  is a parametric representation of the boundary constraints on tissue  $\mathcal{T}$ . In our work, we select each  $b^i$  to signify the strength of the environmental attachment constraint for particle  $i$ . The exact form of these attachment constraints can be tailored to the method chosen to solve the problem statement and hence is discussed later in section IV-A. By parameterizing the attachment constraints for the entire tissue,  $\mathbf{b}_t$  fully specifies the topology and structure of  $\mathcal{T}_f$  and  $\mathcal{T}_b$ . We will refer to this parameter  $\mathbf{b}$  as the boundary parameter.

In this paper, we consider the following estimation task. We assume there exists a real tissue  $\mathcal{T}^{\text{ref}}$  with its attachment constraints specified by true parameters  $\mathbf{b}_t^*$ . We have access to observations on the state of the reference tissue:  $\mathbf{x}_t^{\text{ref}}$ . Our goal is to estimate the true parameters,  $\mathbf{b}_t^*$ , using the observed state of the tissue,  $\mathbf{x}_t^{\text{ref}}$ , and  $\mathbf{b}_t$  denotes our estimated belief.

In the active sensing problem, we aim to find the best actions to estimate  $\mathbf{b}_t$ . The robot must first solve the estimation problem using observations from its current action to update its estimate  $\mathbf{b}_t$ . Then, using this estimate, it must decide on the next action,  $u_{t+1}$ , to maximize future information gain on parameter  $\mathbf{b}_t$ . To formalize this, we consider our estimate  $\mathbf{b}_t$  of the true parameters  $\mathbf{b}_t^*$  as a random variable and seek a controller that minimizes the entropy of our predicted belief,  $\mathcal{H}(\mathbf{b}_{t+1})$ .

---

### Algorithm 1: JIGGLE: online active sensing framework for estimating boundary constraints.

---

```

Input : Initial particles  $\mathbf{x}_0$ , Initial boundary estimation  $\mathbf{b}_0$ , initial covariance  $\Sigma_0$ , real-world tissue  $\mathcal{T}^{\text{ref}}$  or ground truth parameters  $\mathbf{b}^{\text{ref}}$ , initial control point  $u_0$ , motion noise covariance  $W$ , horizon  $T$ 

1  $\hat{\mathbf{b}}_0, \Sigma_0 \leftarrow \mathbf{b}_0, \Sigma_0$ 
2  $t_c \leftarrow 0$ 
3 for  $t$  in  $[0, \dots, T]$  do
    // Incorporate topological changes
    4 if topological change then
        5      $t_c = t$ 
        6     if in simulation then
            7          $\mathbf{b}^{\text{ref}}, \delta\mathbf{b}_t, W_t \leftarrow \text{PerformCutOrSuture}(\mathbf{b}^{\text{ref}})$ 
        8     else
            9          $\mathcal{T}^{\text{ref}}, \delta\mathbf{b}_t, W_t \leftarrow \text{PerformCutOrSuture}(\mathcal{T}^{\text{ref}})$ 
    10     else
    11          $\delta\mathbf{b}_t, W_t \leftarrow \mathbf{0}, W$ 
    12      $\hat{\mathbf{b}}_{t+1|t}, \Sigma_{t+1|t} \leftarrow \text{EKFPredict}(\hat{\mathbf{b}}_{t|t}, \Sigma_{t|t}, \delta\mathbf{b}_t, W_t)$ 
    // Estimate parameters
    13      $M \leftarrow \text{SampleTimeStep}(t_c, t)$ 
    14     if in simulation then
        15          $\mathbf{x}_t^{\text{ref}} \leftarrow f(\mathbf{x}_t^{\text{ref}}, u_t, \mathbf{b}^{\text{ref}})$ 
        16          $\hat{\mathbf{b}}_{t+1}, \Sigma_{t+1} \leftarrow$ 
            JointEst( $\hat{\mathbf{b}}_{t+1|t}, \Sigma_{t+1|t}, \{\mathbf{x}_{t'}^{\text{ref}}\}_{t'=t_c}^t, \{Q_{t'}\}_{t'=t_c}^t, M$ )
    17     else
        18          $\mathcal{T}^{\text{ref}} \leftarrow \text{ApplyControl}(\mathcal{T}^{\text{ref}}, u_t)$ 
        19          $Q_t \leftarrow \text{Perception}(\mathcal{T}^{\text{ref}})$ 
        20          $\hat{\mathbf{b}}_{t+1}, \Sigma_{t+1}, \mathbf{x}_t^{\text{ref}} \leftarrow$ 
            JointEst( $\hat{\mathbf{b}}_{t+1|t}, \Sigma_{t+1|t}, \{\mathbf{x}_{t'}^{\text{ref}}\}_{t'=t_c}^t, \{Q_{t'}\}_{t'=t_c}^t, M$ )
    // Decide the next action
    21     if active sensing then
        22          $\delta u \leftarrow \text{ActiveSensing}(\mathbf{x}_t^{\text{ref}}, u_t, \hat{\mathbf{b}}_{t+1}, \Sigma_{t+1})$ 
    23     else
        24          $\delta u \leftarrow \text{UserDefine}()$ 
    25      $u_{t+1} = u_t + \delta u$ 

```

---

### IV. METHODS

This section covers our approach to 1) estimation, and 2) active sensing of boundary parameters. We first describe our simulation environment in IV-A, and how it is used for estimation in IV-B. We discuss extensions of this framework, necessary for use with real world data in IV-C. IV-D describes the objectives and constraints used for active sensing, followed by the algorithms used to solve for control in V-C.

#### A. Deformable Softbody Simulator

**Extended Position-based Dynamics (XPBD).** In this work, we model the problem using a *quasi-static* extended position-based dynamics (XPBD) simulator. XPBD was chosen for its computational efficiency and ease of specifying new geometric constraints. The following paragraph is a brief summary of the XPBD method and update rule [28].

At time  $t$ , the robot end effector chooses its next position as its action. This is denoted as  $u_t \in \mathbb{R}^3$ . Let  $\mathbf{x}_t$  be the tissue

mesh particle states at time  $t$ . XPBD solves for the next state,  $\mathbf{x}_{t+1}$ , by minimizing the total potential energy of the system:

$$\mathbf{x}_{t+1} = \arg \min_{\mathbf{x}_{t+1}} \mathcal{U}(\mathbf{x}_{t+1}, u_t) \quad (1)$$

The potential energy is defined as:

$$\mathcal{U}(\mathbf{x}_{t+1}, u_t) = 1/2 \mathbf{C}(\mathbf{x}_{t+1}, u_t)^\top \text{diag}(\mathbf{k}) \mathbf{C}(\mathbf{x}_{t+1}, u_t) \quad (2)$$

Here,  $\mathbf{C}(\mathbf{x}_{t+1}, u_t)$  is the union ( $\cup$ ) of all the position-based constraints that define the system. The simulator aims to satisfy these constraints through energy minimization.  $\mathbf{k}$  is a vector that defines the relative weighting on the constraints in  $\mathbf{C}(\mathbf{x}_{t+1}, u_t)$ . Each entry in  $\mathbf{k}$  can be interpreted as the relative stiffness of the spring associated with a particular constraint.

The constraints we use are:  $\mathbf{C}(\mathbf{x}_{t+1}, u_t) = \mathbf{C}_m(\mathbf{x}_{t+1}) \cup \mathbf{C}_a(\mathbf{x}_{t+1}, u_t) \cup \mathbf{C}_b(\mathbf{x}_{t+1})$ . The mesh deformation constraints,  $\mathbf{C}_m$ , include distance constraints defined on every mesh edge and bending constraints between neighboring mesh faces. We extend the XPBD framework by incorporating two new geometric constraints to enable robot end-effector control and environment attachment points. These robot grasp-related constraints are formulated as  $\mathbf{C}_a$ , and  $\mathbf{C}_b$  describes attachment-related constraints. These new constraints are discussed below.

We solve the minimization of  $\mathcal{U}(\mathbf{x}_{t+1}, u_t)$  in Eq. 1 by iteratively updating particle positions, initialized at  $\mathbf{x}_t$ , to satisfy the constraints using the following update rule:

$$\begin{aligned} \Delta \mathbf{x} &= \mathbf{M}^{-1} \nabla \mathbf{C}^\top \Delta \lambda \\ \Delta \lambda &= - \left( \nabla \mathbf{C} \mathbf{M}^{-1} \nabla^\top \mathbf{C} + \tilde{\alpha} \right)^{-1} \left( \mathbf{C} + \tilde{\alpha} \lambda \right) \quad (3) \\ \tilde{\alpha} &= \text{diag}(\mathbf{k})^{-1} / \Delta t^2 \end{aligned}$$

in which  $\mathbf{M}$  is a diagonal mass matrix, and  $\lambda$  is a Lagrange multiplier. For more details on this update rule, see [28].

We summarize the XPBD simulator as follows:

$$\begin{aligned} \mathbf{x}_{t+1} &= f(\mathbf{x}_t, u_t, \mathbf{b}), \\ \text{s.t., } \mathbf{C}_m(\mathbf{x}_{t+1}) &= \mathbf{0}, \mathbf{C}_a(\mathbf{x}_{t+1}, u_t) = \mathbf{0}, \mathbf{C}_b(\mathbf{x}_{t+1}) = \mathbf{0} \quad (4) \end{aligned}$$

where  $\mathbf{b} \subset \mathbf{k}$  are constraint strength parameters for the  $\mathbf{C}_b$  that we will elaborate on below.

**End-effector Grasp Constraints  $\mathbf{C}_a$ .** To enable control, the tissue mesh must be constrained by the current grasp point of the robot. Note that at time  $t-1$ , the robot selects as its action  $u_{t-1}$  the next location of the end effector. Therefore, the current grasp location at time  $t$  is  $u_{t-1}$ . This grasping point is treated as an infinite-mass virtual particle that is connected to points on the mesh within a local neighborhood  $\mathbf{x}_t^g \subset \mathbf{x}_t$ . The  $i$ -th particle is included in  $\mathbf{x}^g$  if  $\|x_1^i - u_0\|_2$ , the initial distance between the  $i^{th}$  particle and chosen grasp point at time  $t=1$ , is less than radius  $r$ . The robot can grasp different parts of the mesh by choosing a different initial grasp point  $u_0$ . In our implementation,  $\mathbf{x}^g$  is constrained by linear springs connecting each entry of  $\mathbf{x}^g$  s.t.  $u_t$ :  $C_a^j(x_{t+1}, u_t) = \|x_{t+1}^j - u_t\|_2 - \|x_1^j - u_0\|_2, x^j \in \mathbf{x}^g, C_a^j \in \mathbf{C}_a$ . We assume knowledge of the spring constants  $\mathbf{k}_a \subset \mathbf{k}$  associated with  $\mathbf{C}_a$ .

**Spring Boundary Constraints  $\mathbf{C}_b$ .** We formulate attachments between a tissue and its surrounding environment with

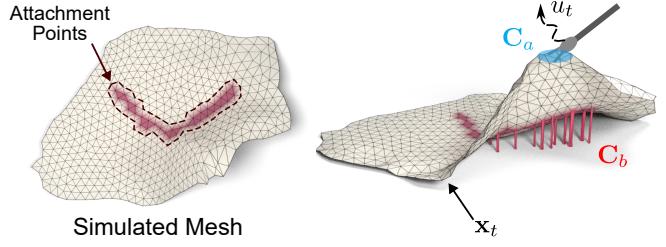


Fig. 2. An illustration of the spring boundary and grasping constraints discussed in section IV-A. Attachment points on the tissue are depicted in red and are modeled using springs of varying stiffness that connect the tissue to the environment. Meanwhile, the grasp is modeled as an infinite-mass virtual particle that is connected via springs to a small neighborhood of particles highlighted in blue.

spring boundary constraints similar to [49]. The idea, illustrated in Fig. 2, assumes that a zero resting-length spring with unknown stiffness exists between every simulation particle  $x_t^i$  and its corresponding infinite-mass *virtual particle* whose position is fixed at  $x_0^i$ . Each attachment can be expressed with a geometric constraint  $C_b^i(\mathbf{x}) = \|x_t^i - x_0^i\|_2, x^i \in \mathbf{x}$ , and the spring constant or the strength of the corresponding spring boundary constraint on particle  $x^i$ , denoted with  $b^i$ .  $\mathbf{b} = [b^1 \dots b^n]^\top \subset \mathbf{k}$  denotes all spring constants between the particles and the environment. Particles attached to the environment have a high stiffness,  $b^i > 0$ . Particles in free regions have  $b^i = 0$ .

We aim to estimate unknown spring boundary parameters,  $\mathbf{b}_t^*$ , from observations of mesh deformations. The proposed simulation is implemented in PyTorch and is fully differentiable using PyTorch’s automatic differentiation package.

### B. EKF-Based Boundary Estimation

We propose a probabilistic estimation framework to track distributions,  $\mathbf{b}_t$ , over true boundary parameters  $\mathbf{b}_t^*$ , in a changing environment. We represent the distribution over our boundary parameters using a multivariate Gaussian distribution. This enables efficient tracking of our boundary estimates using a Kalman filter-based approach. The differentiable XPBD simulation allows us to locally linearize the tissue model to fit an extended Kalman filter (EKF) formulation. The approach enables us to treat the mesh particle states as indirect observations of the boundary parameters, which are our latent random variable states.

**Motion Model.** At any given timestep  $t$ , our belief of the current boundary parameters is represented by the random variable  $\mathbf{b}_t \sim \mathcal{N}(\hat{\mathbf{b}}_t, \Sigma_t)$ . The motion model describes the change in the underlying boundary between timesteps. This captures topology-changing actions that directly modify the boundary parameters. We formulate a boundary-changing action as  $\delta \mathbf{b}_t \in \mathbb{R}^n$ . We assume such changes can only be the result of special actions such as cutting and suturing and not of robot grasping. As a result, we control the value of  $\delta \mathbf{b}_t$ . For later convenience, we denote the last topology-changing

action's time step as  $t_c$ . Our motion model is defined as:

$$\begin{aligned}\mathbf{b}_{t+1} &= m(\mathbf{b}_t, \delta\mathbf{b}_t, w_t) \\ &= \mathbf{b}_t + \delta\mathbf{b}_t + w_t \\ w_t &\sim \mathcal{N}(0, W_t)\end{aligned}\quad (5)$$

In the absence of topological changes,  $\delta\mathbf{b}_t = 0$ . During active changes, elements in  $\delta\mathbf{b}_t$  will be assigned non-zero values, especially in regions closer to the topological change. For example,  $\delta\mathbf{b}_t$  will be given negative values to indicate cutting or positive values to indicate the addition of attachment points through actions like suturing.  $w_t$  is an additive Gaussian motion model noise. This noise has covariance  $W_t$  with increased variance near the modified region, indicating uncertainty about the outcomes of the topology-changing event.

**Observation Model.** Given that the boundary parameters are not directly observable, we observe them indirectly through the state of the tissue,  $\mathbf{x}_t^{\text{ref}}$ , which depends on  $\mathbf{b}_t$ . The observation model  $h$  is defined as:

$$\begin{aligned}\mathbf{x}_{t+1} &= h(\mathbf{b}_{t+1}, \mathbf{x}_t^{\text{ref}}, u_t, v_t) \\ &= f(\mathbf{x}_t^{\text{ref}}, u_t, \mathbf{b}_{t+1}) + v_{t+1} \\ v_{t+1} &\sim \mathcal{N}(0, V_{t+1})\end{aligned}\quad (6)$$

Here, we initialize our XPBD simulation,  $f$ , using the last observed tissue surface  $\mathbf{x}_t^{\text{ref}}$  and simulate one time step forward with the updated belief of the boundary parameters  $\mathbf{b}_{t+1}$ , updated using the motion model  $m$ , and the control action  $u_t$ . The covariance matrix  $V_{t+1}$  directly characterizes the noise in observations of the tissue surface. We use these elements to establish the following EKF equations to track the changes in our boundary parameter belief, represented by  $\hat{\mathbf{b}}_t$  and  $\Sigma_t$ .

**Prediction :**

$$\begin{aligned}\hat{\mathbf{b}}_{t+1|t} &= \hat{\mathbf{b}}_{t|t} + \delta\mathbf{b}_t \\ \Sigma_{t+1|t} &= \Sigma_{t|t} + W_t\end{aligned}\quad (7)$$

**Update :**

$$\begin{aligned}\hat{\mathbf{b}}_{t+1|t+1} &= \hat{\mathbf{b}}_{t+1|t} + K_{t+1} \tilde{\mathbf{y}}_{t+1} \\ \Sigma_{t+1|t+1} &= (I - K_{t+1} J_{t+1}) \Sigma_{t+1|t}\end{aligned}\quad (8)$$

where the observation residual  $\tilde{\mathbf{y}}_{t+1}$ , and the Kalman gain  $K_{t+1}$  are given by:

$$\begin{aligned}\tilde{\mathbf{y}}_{t+1} &= \mathbf{x}_{t+1}^{\text{ref}} - h(\hat{\mathbf{b}}_{t+1|t}, \mathbf{x}_t^{\text{ref}}, u_t, v_t) \\ K_{t+1} &= \Sigma_{t+1|t} J_{t+1}^{\top} S_{t+1}^{-1}\end{aligned}\quad (9)$$

with the residual covariance  $S_{t+1}$ , the observation Jacobians  $J_{t+1}$ , and covariance Jacobians  $R_{t+1}$  as:

$$\begin{aligned}S_{t+1} &= J_{t+1} \Sigma_{t+1|t} J_{t+1}^{\top} + R_{t+1} V_{t+1} R_{t+1}^{\top} \\ R_{t+1} &= \frac{\partial h}{\partial v} \Big|_{\hat{\mathbf{b}}_{t+1|t}} = I \\ J_{t+1} &= \frac{\partial h}{\partial \mathbf{b}} \Big|_{\hat{\mathbf{b}}_{t+1|t}} = \frac{\partial f(\mathbf{x}_t^{\text{ref}}, u_t, \mathbf{b}_{t+1})}{\partial \mathbf{b}} \Big|_{\hat{\mathbf{b}}_{t+1|t}}\end{aligned}\quad (10)$$

Here,  $f(\mathbf{x}_t^{\text{ref}}, \hat{\mathbf{b}}_{t+1|t}, u_t)$  outputs the expected tissue surface after using the XPBD simulator to forward simulate one

timestep with action  $u_t$ , using the mean of the current estimated boundary belief,  $\hat{\mathbf{b}}_{t+1|t}$ .

**Multiple Shooting Metric.** To enhance the robustness of the estimation, we modify the EKF to use previous observations within a specified horizon. This is possible with the assumption that no topology changing events occur within the considered observation window and that boundary parameters are constant unless changed by these controlled topology changing events. We use the multiple shooting method to encode previous observations [11]; this approach samples from previously observed trajectories, forward simulates from those samples using the current belief and penalizes the deviation of predicted particle states from the reference observations. For simplicity, we set the prediction horizon to 1. The new EKF update step is defined as

$$\begin{aligned}\mathcal{X}_{M+1}^{\text{ref}} &= \begin{bmatrix} \mathbf{x}_{m1+1}^{\text{ref}} \\ \vdots \\ \mathbf{x}_{mk+1}^{\text{ref}} \end{bmatrix}, \quad \mathcal{X}_{M+1}^{\text{pred}} = \begin{bmatrix} f(\mathbf{x}_{m1}^{\text{ref}}, u_{m1}, \mathbf{b}_{t+1|t}) \\ \vdots \\ f(\mathbf{x}_{mk}^{\text{ref}}, u_{mk}, \mathbf{b}_{t+1|t}) \end{bmatrix} \\ J_{M+1} &= \begin{bmatrix} \frac{\partial f(\mathbf{x}_{m1}^{\text{ref}}, u_{m1}, \mathbf{b}_{t+1|t})}{\partial \mathbf{b}} \Big|_{\hat{\mathbf{b}}_{t+1|t}} \\ \vdots \\ \frac{\partial f(\mathbf{x}_{mk}^{\text{ref}}, u_{mk}, \mathbf{b}_{t+1|t})}{\partial \mathbf{b}} \Big|_{\hat{\mathbf{b}}_{t+1|t}} \end{bmatrix}, \quad R_{M+1} = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} \\ \hat{\mathbf{b}}_{t+1|t+1} &= \hat{\mathbf{b}}_{t+1|t} + K_{t+1|t} (\mathcal{X}_{M+1}^{\text{ref}} - \mathcal{X}_{M+1}^{\text{pred}}) \\ \Sigma_{t+1|t+1} &= (I - K_{t+1|t} J_{M+1}) \Sigma_{t+1|t} \\ K_{t+1|t} &= \Sigma_{t+1|t} J_{M+1}^{\top} (J_{M+1} \Sigma_{t+1|t} J_{M+1}^{\top} + R_{M+1} V_{t+1} R_{M+1}^{\top})^{-1}\end{aligned}\quad (11)$$

Here,  $M = \{m_1, \dots, m_k\}$  are  $k$  samples drawn by first selecting  $m_k = t$  and uniformly sampling  $k-1$  in the range  $[t_c, t-1]$ .  $t_c$  is the time step of the last topology-changing event. We do not include time steps before this as the previous topology has been altered and is therefore irrelevant.

#### C. Joint Boundary Parameter and State Estimation for Real-world Scenarios

Our estimation framework requires the knowledge of reference particle positions  $\{\mathbf{x}_t^{\text{ref}}\}$  to update spring boundary parameters. This information is easily accessible in simulation but requires additional effort in the real world. Let  $\mathcal{T}^{\text{ref}}$  denote the real tissue. Via robot perception, we can access dense surface point clouds  $Q_t$  of the tissue.  $\mathbf{x}_t^{\text{ref}}$  can be estimated by finding a configuration of mesh particles that matches  $Q_t$  and respects geometric constraints.

Similar to [22], we achieve this by introducing an observation matching condition into the XPBD's simulation's iterative constraint solving procedure. Specifically, the first row of Eq. 3 is modified to

$$\Delta \mathbf{x} = \mathbf{M}^{-1} \nabla \mathbf{C}^{\top} \Delta \lambda - \alpha \nabla_{\mathbf{x}} \mathcal{P}({}^i \mathbf{x}, Q_t) \quad (12)$$

where  ${}^i \mathbf{x}$  represents intermediate simulation particles in the iterative process. Additionally,  $\alpha$  is a step size, and  $\mathcal{P}(\cdot)$  is

a loss metric that measures the difference between  ${}^i\mathbf{x}$  and  $Q_t$ . To compute the metric, another dense point cloud  $P_t$  is sampled from the simulation mesh using differentiable point cloud sampling [43]. The loss metric is then defined by the chamfer distance between two point clouds

$$\begin{aligned}\mathcal{P}({}^i\mathbf{x}, Q_t) &= \mathcal{CD}({}^iP, Q_t) \\ &= \sum_{p \in {}^iP} \min_{q \in Q_t} \|p - q\|_2^2 + \sum_{q \in Q_t} \min_{p \in {}^iP} \|q - p\|_2^2.\end{aligned}$$

Note that in Eq. 12 for estimating  $\mathbf{x}_t^{\text{ref}}$ ,  $\Delta\lambda$  depends on the current parameter estimate  $\hat{\mathbf{b}}_t$ . Therefore, the two variables,  $\mathbf{x}_t^{\text{ref}}$  and  $\hat{\mathbf{b}}_t$ , are coupled through the registration of the point cloud data. This motivates us to combine the estimation of  $\mathbf{x}_t^{\text{ref}}$  and the update step of EKF into one joint estimation problem, where both sub-problems are iteratively estimated:

$$\begin{aligned}\hat{\mathbf{b}}_{t+1}, \Sigma_{t+1} &= \text{EKF Update}(\hat{\mathbf{b}}_t, \Sigma_t, \{\mathbf{x}_t^{\text{ref}}\}_{t=t_c}^t, M) \\ \text{s.t. } \mathbf{x}_{m+1}^{\text{ref}} &= f'(\mathbf{x}_m^{\text{ref}}, u_m, \hat{\mathbf{b}}_t, Q_m), m \in M \\ \mathbf{C}(\mathbf{x}_{m+1}^{\text{ref}}) &= \mathbf{0}, \mathbf{C}_a(\mathbf{x}_{m+1}^{\text{ref}}, u_m), \mathbf{C}_b(\mathbf{x}_{m+1}^{\text{ref}}) = \mathbf{0}\end{aligned}\quad (13)$$

where  $f'(\cdot)$  is a modified XPBD equation that uses Eq. 12.

#### D. Objectives and Constraints for Active Sensing Control

This section delves into active sensing methodologies that we build on top of our estimation framework. We leverage our current belief of the tissue boundary parameters to inform the selection of next step actions and encourage the discovery of additional tissue attachment points. We investigate four unique active control objectives. These objectives are **Boundary Entropy Minimization**, **Uncertainty-Weighted Displacement (UWD)**, **Boundary Energy Minimization**, and **In-Workspace Constraints**. Each objective offers a unique perspective on guiding actions for exploration.

**Boundary Entropy Minimization.** Entropy is a measure of uncertainty of a probability distribution, where a higher entropy indicates higher uncertainty. We therefore seek to take actions that reduce the entropy  $\mathcal{H}$  of our belief distribution  $\mathbf{b}_t$ :

$$\mathcal{H}(\hat{\mathbf{b}}_{t+1}) = - \int p(\hat{\mathbf{b}}_{t+1}) \ln p(\hat{\mathbf{b}}_{t+1}) d\mathbf{b} \quad (14)$$

As a result, we represent the entropy's evolution as a function of the estimated boundary  $\mathbf{b}_{t+1}$  at time  $t+1$  by employing the update step equations from the EKF as outlined in Eq. 8.

$$\mathcal{H}(\hat{\mathbf{b}}_{t+1}) = \frac{D}{2}(1 + \ln(2\pi)) + \ln(|\Sigma_{t+1}|) \quad (15)$$

$$\approx \ln(|\Sigma_{t+1}|) \quad (16)$$

where  $D$  is the dimensionality of the multivariate Gaussian distribution. As the first term in Eq. 15 is a constant  $c = \frac{D}{2}(1 + \ln(2\pi))$ , it can be disregarded during optimization. We employ this objective to select the control action that minimizes the entropy through optimization in the following section.

**Uncertainty-Weighted Displacement (UWD) Maximization.** The strategy of minimizing entropy is heavily dependent on the accuracy of the current boundary parameter estimate. In a complex deformable environment, the greedy minimization

of this strategy with imperfect boundary estimates is prone to local minima. Moreover, the computational complexity associated with entropy propagation, i.e.,  $\nabla_{u_t} \Sigma_{t+1|t+1}$ , limits the applicability of this heuristic in sampling-based and multi-step trajectory optimization approaches.

By introducing a heuristic function for uncertainty-weighted displacement (UWD), denoted as  $\mathcal{D}$ , our new metric relies exclusively on the covariance matrix  $\Sigma_t$  updated from the preceding step. Notably, this matrix is independent of the current step motion, as evidenced by  $\nabla_{u_t} \Sigma_t = 0$ . To address challenges in the previous metric, particularly in computing intense gradients, our proposed function directly calculates deformation displacement. The objective is expressed as:

$$\mathcal{D}(\mathbf{x}_t^{\text{ref}}, u_t, \hat{\mathbf{b}}_t) = \left\| \widehat{\Delta \mathbf{x}_{t+1}} \cdot \Sigma_t \right\| \quad (17)$$

$$\begin{aligned}\mathbf{x}_{t+1} &= f(\mathbf{x}_t^{\text{ref}}, u_t, \hat{\mathbf{b}}_{t+1|t}) \\ \Delta \mathbf{x}_{t+1} &= \mathbf{x}_{t+1} - \mathbf{x}_0 \\ \widehat{\Delta \mathbf{x}_{t+1}} &= \sum_i^n \mathbf{e}_i \mathbf{e}_i^\top \otimes [\Delta \mathbf{x}_{t+1}]_i.\end{aligned}\quad (18)$$

Here,  $\Delta \mathbf{x}_{t+1}$  represents the simulated displacement of the tissue mesh particles at time  $t+1$  from their original positions at  $t = 0$ , resulting from the forward XPBD simulation  $f(\mathbf{x}_t^{\text{ref}}, u_t, \hat{\mathbf{b}}_{t+1|t})$  under the action  $u_t$ , and the current updated boundary belief  $\hat{\mathbf{b}}_{t+1|t}$ . A concatenated-matrix form  $\widehat{\Delta \mathbf{x}_{t+1}}$  of this displacement is computed by letting  $\mathbf{e}_i$  be the  $i$ -th column vector of the standard basis of  $\mathbb{R}^n$ , and  $[\Delta \mathbf{x}_{t+1}]_i \in \mathbb{R}^{3 \times 1}$  be the deformation displacement of the  $i$ -th particle in  $\mathbf{x}_{t+1}$ . Then, a deformation displacement matrix  $\widehat{\Delta \mathbf{x}_{t+1}}$  can be derived with  $\otimes$  denoted as Kronecker product operation, facilitating the proper weighting of particle displacement by the covariance of our current boundary belief  $\Sigma_t$ .

The objective of maximizing the  $\mathcal{D}$  in Eq. 17 is akin to maximizing tissue displacement, with each region weighted by the uncertainty of the current belief in that location. This heuristic consequently emphasizes the enhancement of displacement in regions of the tissue surface characterized by higher uncertainty. In the Appendix VIII-A, we have presented a proposition along with an explanation demonstrating that this objective can still guide efficient entropy minimization without directly relying on the future prediction of the covariance matrix, i.e.,  $\Sigma_{t+1|t+1}$ .

**Boundary Constraint Energy Minimization.** During active exploration, we want to pose safety constraints that avoid exerting excessive force at connected regions. In surgery, this is critical to prevent unwanted tearing and bleeding. We formulate an energy-based safety metric to penalize high elastic potential energy at the boundary. The elastic potential energy of boundary constraints is computed similarly to Eq. 2:

$$U_b(\mathbf{x}_t^{\text{ref}}, u_t, \hat{\mathbf{b}}_t) = \frac{1}{2} \mathbf{C}_b(\mathbf{x}_{t+1})^\top \text{diag}(\hat{\mathbf{b}}_t) \mathbf{C}_b(\mathbf{x}_{t+1}) \quad (19)$$

**In-Workspace Constraint.** In surgery, the workspace of the robot is often constrained due to the ports used to insert the laparoscopic tools and obstructions created by organs

and other tissues. We formulate a position-based workspace constraint to limit the range of the next grasp point. This penalizes actions that exit a constraint region. For example, in our work we consider the constraint where the robot should not pull the tissue downward through the ground plane, which represents the anchoring tissue. We represent this constraint function as:

$$\Omega(u_t) = e^{-\min([u_t]_z, 0)} \quad (20)$$

where  $[u_t]_z$  is the height of the grasp point from action  $u_t$ . While in surgery the environment can also deform, here we assume the ground plane doesn't change. Though we use the constraint in Eq. 20, this constraint can be tailored to restrict the choice of actions for the scenario at hand.

### E. Optimization-based Controller Design

Based on our choices of active sensing objectives, we present two variants of controllers that both take advantage of the differentiability of the PBD simulator. They are 1) a local gradient-based action controller and 2) a sampling-based large-step action controller.

**Local Gradient-based Action Control.** We locally solve for the action that minimizes the entropy of our belief distribution on the boundary parameters without damaging the tissue at the boundary or violating in-workspace constraints:

$$\begin{aligned} & \arg \min_{u_t} \mathcal{H} + \mathcal{U}_b \\ \text{s.t. } & \Omega(u_t) \geq \Omega_{\min} \end{aligned} \quad (21)$$

Nevertheless, dealing with this high-dimensional optimization challenges traditional solvers. To mitigate this, we relaxed the constraints using a penalty-based method, addressing Eq. 21 as an unconstrained optimization problem, solved using gradient descent. This approach capitalizes on the differentiability of our XPBD simulator.

$$\min_{u_t} \mathcal{L}_{\mathcal{H}} = \min_{u_t} \mathcal{H} + \mathcal{U}_b + \sigma \|\max \{0, \Omega_{\min} - \Omega\}\| \quad (22)$$

where  $\sigma$  denotes a weight. Gradient descent iteratively updates the best local action to take:

$$u_t = u_{t-1} - \alpha \nabla_{u_t} \mathcal{L} \quad (23)$$

To simplify the gradient calculation, we replace entropy ( $\mathcal{H}$ ) in Eq. 22 with uncertainty-weighted displacement (UWD),  $\mathcal{D}$ , from Eq. 17. This substitution is made possible given the proposition VIII-A, in the appendix, which illustrates the relationship between entropy,  $\mathcal{H}$ , and UWD,  $\mathcal{D}$ . With this we have the new loss function:

$$\mathcal{L}_{\mathcal{D}} = -\mathcal{D} + \mathcal{U}_b + \sigma \|\max \{0, \Omega_{\min} - \Omega\}\| \quad (24)$$

Once optimized, the robot executes action  $u_t$ , obtains an observation  $\mathbf{x}_{t+1}^{\text{ref}}$ , updates the belief to be  $\hat{\mathbf{b}}_{t+1}$ , and then repeats this process.

In practice, we noticed that this local approach stopped deforming the tissue after a few iterations. This is attributed to vanishing gradients, signifying that the optimization has

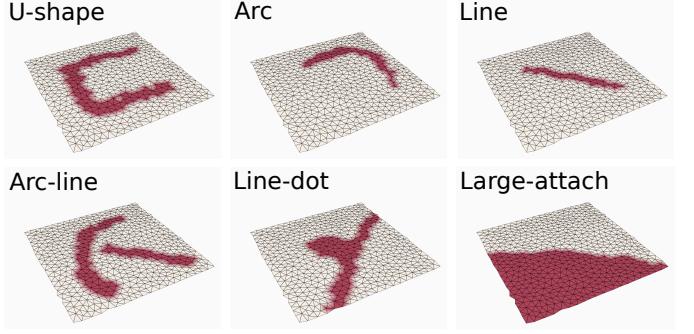


Fig. 3. The images show the ground truth attachment points in red using spring boundary constraints for the simulation test cases in our experiments.

reached a point where local-step control actions have a negligible impact on reducing the loss term.

**Sampling-based Large Step Action Control.** To address the challenges posed by local minima and gradient vanishing issues, we explore longer-horizon action sequences for more information acquisition. Although trajectory optimization can mitigate these issues, it becomes computationally demanding, especially for high degree-of-freedom deformable objects. Therefore, we introduce a sampling-based large step controller to overcome the limitations associated with gradient locality.

Instead of considering a long horizon with multiple control steps, we optimize a set of large-step control samples  $\mathbb{S}$  from the workspace in parallel. For each sample, we take a single large control step from the current grasp point. We reformulate the following minimization problem:

$$\begin{aligned} s_k^* &= \arg \min_{s_k \in \mathbb{S}} \mathcal{L}_{\mathcal{D}}(\mathbb{S}) \\ \text{s.t. } & \mathbb{S} = \{s_1, s_1, \dots, s_k\} \\ & s_k = s_k - \alpha \nabla_u \mathcal{L}_{\mathcal{D}} \end{aligned} \quad (25)$$

After finding the best large step action  $s^*$  from the set  $\mathbb{S}$ , we move the current grasp point  $u_{t-1}$  towards it using truncated control length  $\gamma$ .

$$u_t^* = u_{t-1} + \gamma \frac{s_k^* - u_{t-1}}{\|s_k^* - u_{t-1}\|} \quad (26)$$

We re-sample the large-step control samples  $\mathbb{S}$  between each action by uniformly sampling the workspace. To encourage smooth trajectories and a balance between exploration and exploitation, we retain the top 10% of samples in  $\mathbb{S}$  (samples with the lowest loss values,  $\mathcal{L}_{\mathcal{D}}$ ) from the previous iteration, when re-sampling  $\mathbb{S}$ . It is worth noting that the optimization of sample sets can be computed concurrently on the GPU for accelerated processing.

Our full algorithm is summarized in Alg. 1. In line 4-12, we handle topological changes using the prediction step of our probabilistic estimation framework (section IV-B). Following that, visual feedback is used to update our boundary parameter belief in a joint estimation manner in line 13-20 (section IV-B, IV-C). Finally, the proposed active sensing objectives and controllers section IV-D, IV-E can be used to determine the next action in line 21-25.

TABLE I

COMPARISON OF BOUNDARY ESTIMATION ACCURACY AFTER 4 GRASP SEQUENCES. RESULTS ARE COMPARED BETWEEN OUR METHODS AND THE ADAM OPTIMIZER. THE PROPOSED OUTPERFORMS BETTER THAN ADAM IN ALL TEST CASES.

Cases	PCD ↑		PUG ↑	
	Adam	Ours	Adam	Ours
Arc	82.1	100	59.3	87.1
Line	85.7	100	40.0	43.3
Line-dot	68.0	96.0	34.7	48.9
Arc-line	83.8	97.3	51.8	65.5
U-shape	87.3	97.5	52.9	76.0

## V. EVALUATIONS

In our experiments, we seek to evaluate our framework’s capabilities in 1) estimation accuracy in both simulated and real scenarios, 2) actively maximizing information gain while considering safety, and 3) intelligently adapting to topological changes. The full capabilities of our approach are evaluated in simulated environments in Sections V-B, V-C, and V-D. In Section V-E, we analyze only our estimation performance in real-world tissue scenarios.

### A. Evaluation Metrics

**Accuracy.** Estimation accuracy is measured in two terms. They are 1) the percentage of correct detection (PCD): the number of correct attachment point detections over the total number of detections, and 2) the percentage of uncovered ground truth (PUG): the number of uncovered attachment points divided by the total number of connections. The  $i$ th mesh point  $\mathbf{x}_t^i$  is considered attached at time  $t$  if  $b_t^i > b_{\text{thresh}}$ . In our experiments  $b_{\text{thresh}} = 0.05$ .

Due to observational noise and limitations posed by discretization, estimated attachment points can be spatially close to ground truth but not perfectly overlap. Therefore, we report results from performing a dilation operation to expand the ground truth attachment region spatially and estimated connections for computing PCD and PUG, respectively. Dilation is implemented with graph convolution on simulation meshes.

**Active Sensing Performance.** Different active sensing methods are compared in terms of the amount of entropy reduction within a fixed number of timesteps. All comparison active sensing methods are built on top of the proposed EKF-based estimation framework. Their entropy is calculated using Eq. 14. We also evaluate the safety performance of comparison methods in terms of the mean boundary energy calculated using ground truth boundary stiffness and Eq. 19. In this term, a larger value means a higher chance of unsafe tearing.

### B. Estimation Capabilities in Simulation

We first evaluate the estimation accuracy of the proposed method in a simulated environment with a pre-defined set of robot actions (i.e. no active sensing). We manually define different shapes of connected regions that are visualized in Fig. 3. The boundary parameters at the attached regions are set with a ground truth stiffness  $\mathbf{b}^* = 0.1$ . We design control

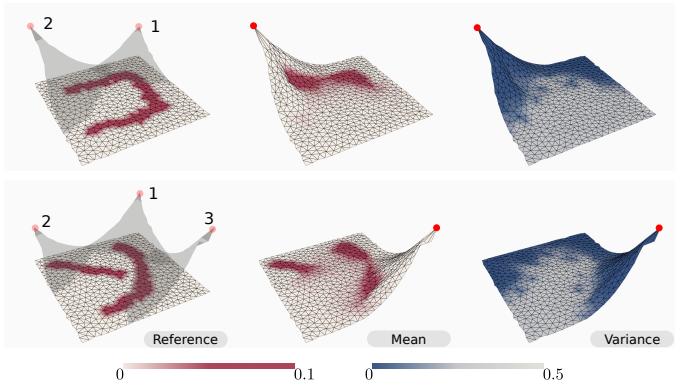


Fig. 4. Example results from our estimation framework on simulated environments where the order of the grasps is numbered, and the boundary is highlighted in red on the left-most column. The final result from our proposed method is shown in red in the middle column. Finally, the confidence (inverse of variance) of our estimation is shown in blue in the rightmost column. We can see how the variance has decreased in the regions where the trajectories have displaced the tissue from its original state, and the mean estimate has converged close to the reference values.

sequences that can potentially reveal the entire shape of the attached region. For each test case (case *Large-attach* excluded in this experiment), four corners of the squared mesh are sequentially grasped and lifted up to a height that is roughly equal to half of the mesh width. In each test case, we initialize our belief on the boundary parameters with the assumption that there are no attachment points by setting the mean of our boundary belief to be very small at  $\hat{\mathbf{b}}_0 = 1e - 4$ . For our baseline, we use a stochastic gradient descent (Adam) algorithm that optimizes a similar multiple shooting metric to directly update a deterministic estimate of the boundary parameters by differentiating through the XPBD simulator. To update the boundary parameters using Adam, we first compute a multiple shooting based loss:

$$\mathcal{L}_{\text{Adam}} = \frac{1}{k} \|\mathcal{X}_{M+1}^{\text{ref}} - \mathcal{X}_{M+1}^{\text{pred}}\|_2^2$$

Where  $\mathcal{X}_{M+1}^{\text{ref}}$  and  $\mathcal{X}_{M+1}^{\text{pred}}$ , as defined in Eq. 11, consist of  $k$  ground truth observations and predicted samples respectively. Leveraging the differentiability of our XPBD simulation, we compute a backward pass to get the gradients with respect to the boundary parameters  $\hat{\mathbf{b}}_t$  that minimize this loss. We use these gradients with pytorch’s Adam optimizer to directly step and update the estimated boundary parameters  $\hat{\mathbf{b}}_t$ .

The complete set of results after the execution of the entire trajectory is shown in Table I. Our proposed approach performs better than the baseline, uncovering more attachment regions with improved accuracy under the same manipulation sequences. Example results from our estimation method are visualized in Fig. 4, which demonstrates that the proposed EKF estimator is able to identify partial attachment regions, as shown by the current mean of the boundary parameter estimate. Regions that have not been influenced by previous control sequences remain uncertain, as shown by the variance.

### C. Effectiveness of Active Exploration

We compare the active sensing performance of four different variants of the proposed framework:

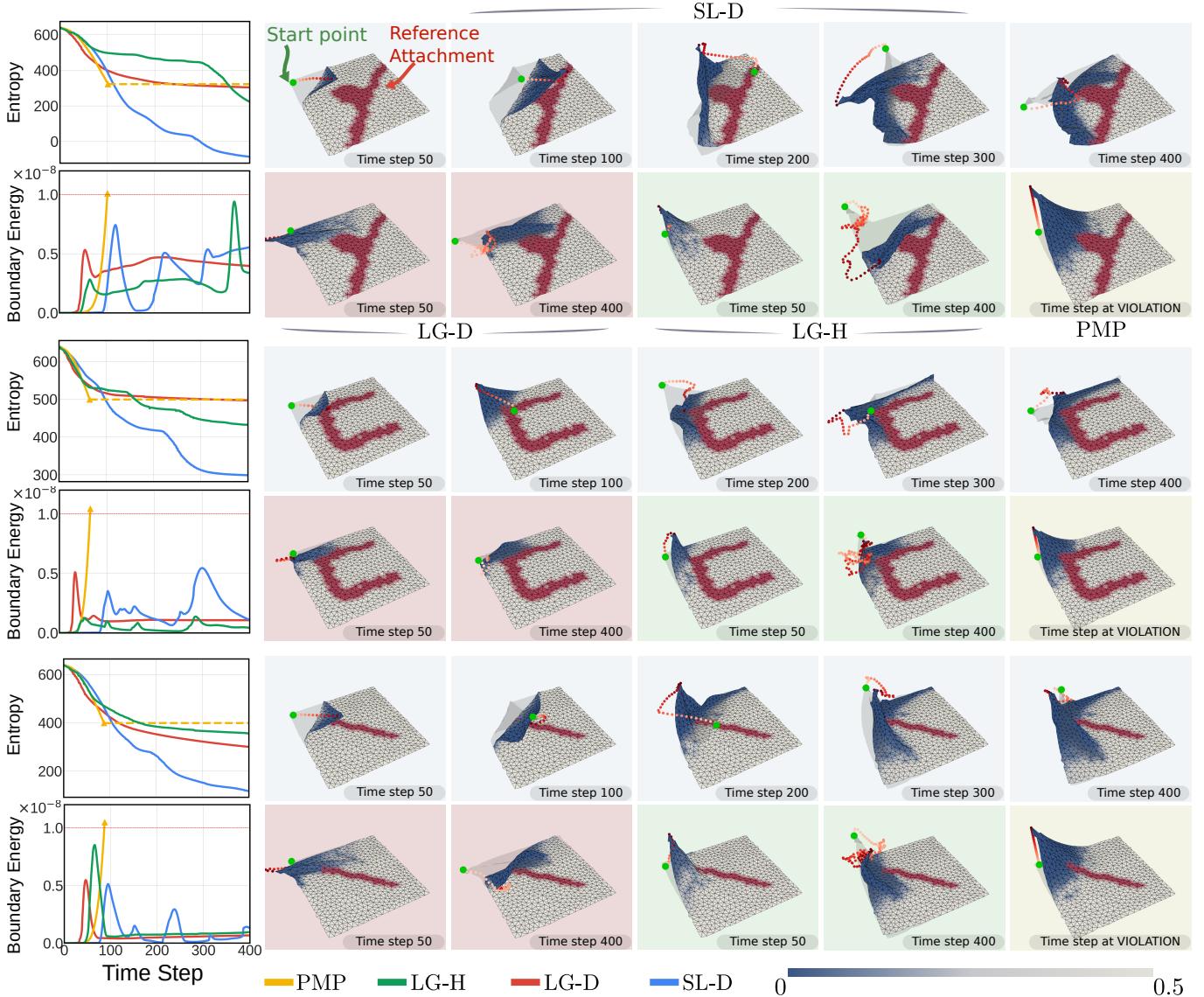


Fig. 5. Results from our active sensing experiments with 4 different strategies: PMP (yellow) is a baseline, and SL-D (blue), LG-H (green), and LG-D (red) are proposed in this work. Every two rows show one experiment with the entropy and energy plotted in the left-most column, and the images on the right show a collage of the control trajectories being applied from the different active sensing strategies. Note that the colored background on each image corresponds to the active sensing strategy (best viewed in color). The red color on the tissue highlights the reference attachment, and the blue shows the confidence, inverse of variance, of the estimated boundary. The goal of the active sensing strategies is to maximize the confidence, which is measured in entropy, while adhering to safety constraints, which are measured in energy. Overall, SL-D achieves more entropy reduction than all other baselines while keeping a safe boundary energy profile. It also produces the most intricate control point trajectories, such as switching directions and folding. In comparison, local controllers LG-H and LG-D get trapped in local minima, resulting in higher entropy. PMP reduces entropy in the beginning but results in quick safety violations.

- 1) **LG-H:** Local gradient-based action, minimizing entropy using Eq. 22 and Eq. 23,
- 2) **LG-D:** Local gradient-based, maximizing displacement using Eq. 24 and Eq. 23,
- 3) **SL-D:** Sampling-based large step action, maximizing displacement using Eq. 24, Eq. 25, and Eq. 26, and
- 4) **PMP:** Predefined motion primitives, a method that conducts an exhaustive search to minimize entropy over motion primitives of  $(\pm x, \pm y, \pm z)$ . [2]

To give the baseline, PMP, the best chance for comparison, we run all motion primitives with our estimation framework and show the result of the motion primitive that gives the most entropy reduction. All active exploration experiments in

this sub-section generate a single continuous control sequence without changing to a new grasp point. This mimics surgical scenarios where re-grasping different regions of the tissue may be limited due to constraints in the robot's workspace.

We desire our framework to output actions that maximize information gain while remaining safe. We measure information gain as entropy reduction of our estimated boundary parameter belief  $\mathcal{H}(\mathbf{b}_t)$ . An action is considered safe if the resulting mean boundary potential energy of the system remains below a specified threshold, which we set as  $\mathcal{U}_{b,\max} = 1e-8$ . This threshold is our safety constraint, and if violated, represents that damage was caused to the tissue (e.g. tearing).

The simulated environments and results are shown in Fig. 5.

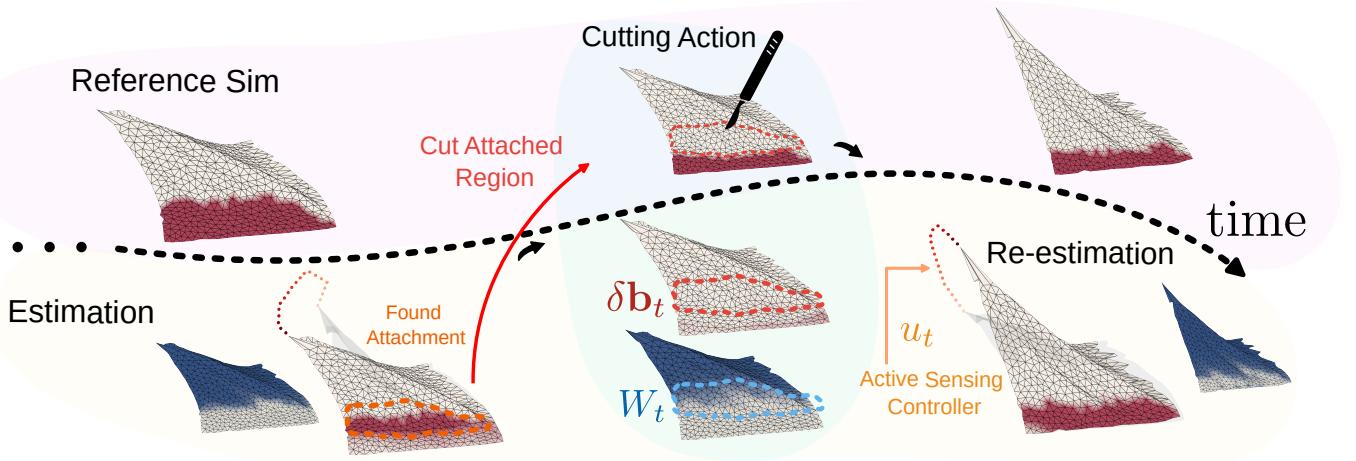


Fig. 6. We applied JIGGLE to a tissue detachment procedure on test case *Large-attach*, which is done by iteratively applying the active sensing approach to find the attachment points, and then a cut is made at that discovered boundary. The sequence of images depicts an iteration of this procedure where the top row shows the ground-truth boundary attachment in red, and the bottom row shows our estimation of the boundary in red and the confidence, inverse of variance, in blue. After each cut action is applied, where all of the discovered boundary is removed, the cutting information,  $\delta b_t$ ,  $W_t$ , is also fed into our estimation algorithm. The strategy is repeated until the tissue is fully detached and it took 7 cycles in this experiment to detach the tissue successfully .

The PMP method successfully reduces the entropy of  $b_t$ . However, in each experiment, it quickly causes the energy to surge and exceed our safety threshold. Although the authors in [2] are able to tune their motion primitives to operate safely for their demonstration, in reality, hand-tailoring unique, safe motion primitives for individual environments is infeasible. The other control rules, LG-H, LG-D, SL-D, can reduce entropy while staying within the safety boundary. While LG-D and LG-H perform comparably, we notice that LG-D is, on average, capable of slightly greater entropy reduction than LG-H. We hypothesize that this is due to LG-H’s heavy reliance on the estimated belief, which leads it to get stuck in more local minima than LG-D, whose heuristic shows greater detachment from the current boundary parameter estimate.

When comparing entropy reduction, we see that the local gradient methods LG-H and LG-D were able to achieve entropy reduction comparable to PMP while maintaining a much lower mean boundary energy. SL-D largely outperformed other methods with improved entropy reduction outcomes which we believe is due to SL-D’s ability to avoid local minima. All our proposed methods, including the local gradient approaches, are capable of complex control strategies. Fig. 5 even shows how the robot selects actions that flip and fold uncertain regions in the cloth to get observations that reduce entropy without excessively pulling at the attached regions.

Despite their more rigorous formulation, the local gradient methods failed to decrease entropy much beyond the baseline PMP method. In Fig. 5, both the entropy plots and the variance graph show that these local methods fail to make much progress after the first 100 time steps. We hypothesize that this is because these methods are susceptible to getting trapped in the local minima of their complex objective functions. This is evidenced by viewing the trajectories in Fig. 5, where in trajectories requiring complex pulling and folding on opposite sides LG-D and LG-H get stuck after reducing entropy by

pulling on one side. SL-D, however, continues to reduce entropy even after the local methods get stuck by overcoming the gradient locality problem via large-step sampling. By sampling and evaluating several long-range trajectories, this method is able to manipulate the cloth in directions that lead to larger entropy reduction in the long term that may not be apparent through short-term local evaluation.

#### D. Adaptation to Topological Changes

We show how our framework can be used with topology-changing actions like cutting and suturing in tasks that require re-estimation of the boundary parameters. We showcase the cutting case in Fig. 6, where the goal is to free the tissue from the environment by iteratively finding and severing all attachment points. In this experiment the tissue itself is not cut, but rather the connection between the tissue and the base environment. The iterative approach first uses our active sensing algorithm to find attachment points on the tissue. A cutting action is then used to detach the found region. Knowledge of the cut is used to update our mean and variance estimations using  $\delta b_t$ ,  $W_t$  and the model  $m$ .  $\delta b_t$  is set to be negative near the points selected to be cut with the cutting action.  $W_t$  is set to inject noise into area close to the cut regions, indicating uncertainty about the success of our action. Being attentive to the increased uncertainty, our active sensing strategy takes actions to safely minimize entropy and re-estimates the boundary parameters. This both validates the effect of our cut action, as well as localizes the remaining attachment points to be cut with the next action.

We also show our framework used for suturing tasks in Fig. 7. We first identify a free region to be attached with a suture (i.e. adding a boundary). After a suturing action, we update our boundary parameter estimate in a similar manner as the cutting case.  $\delta b_t$  is chosen to be greater than 0 at values near the sutured region to indicate the added attachment.

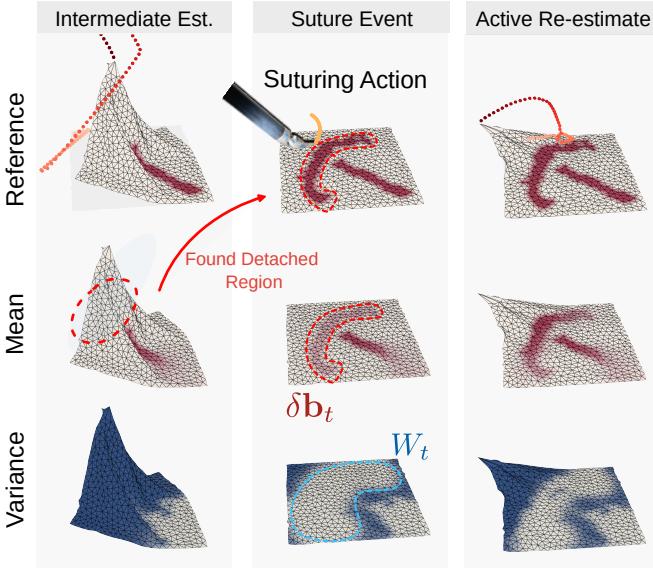


Fig. 7. We applied JIGGLE to a suturing procedure by 1) the left column: applying the active sensing approach to find where the tissue is detached, 2) the middle column: applying a suture action at the desired detached area, and 3) the right column: active sensing again to validate the suture action. The top row of images shows the ground truth attachment in red, and the bottom row shows our estimation of the boundary in red and the confidence, inverse of variance, in blue. After each suture is applied, the suture information is also fed into our estimation algorithm, including a noise injection near the suture region, so the active sensing policy is encouraged to confirm where the new boundary has been added. The second re-estimation, confirming the boundary after the suture, reports an increase in PUG from 20.2 to 75.2. The PCD value remains the same at 100, indicating no false detections.

$W_t$  is set to inject noise near the sutured region, indicating uncertainty in the success of creating new attachments. Our active sensing algorithm takes action to estimate the new boundary parameters, minimizing uncertainty around the modified region and verifying the success of our actions.

#### E. Estimation Capabilities in Real World

Finally, we test our estimation algorithm in a real-world experiment in three cases that mimic surgical tissue manipulation scenarios. A layer of chicken skin, which will be manipulated, is placed on top of a larger chicken thigh. We suture regions of the chicken skin onto the meat to create attachment points. We use an endoscopic stereo RGB camera from the da Vinci Research Kit (dVRK) [16] for sensing and a Tweezer's tooltip for grasping. From the collected images, we extract dense surface point cloud observations of the chicken skin tissue deformation. We use Segment-Anything [18] for semantic segmentation and Raft-stereo [24] for disparity estimation. We then transform 2D image pixels to 3D using an inverse camera projection model. The tweezers' tooltip trajectories are first manually labeled on the images and then transformed to 3D space in a similar manner. Initial simulation meshes are reconstructed from the first point cloud observations of the chicken skin. To reduce the real-to-sim gap that is not caused by boundary connections, we perform system identification on other non-boundary simulation parameters using tools from [22]. In total, 3 test cases are collected on 3 pieces of chicken skins, each with unique attachment shapes. In case 1 and

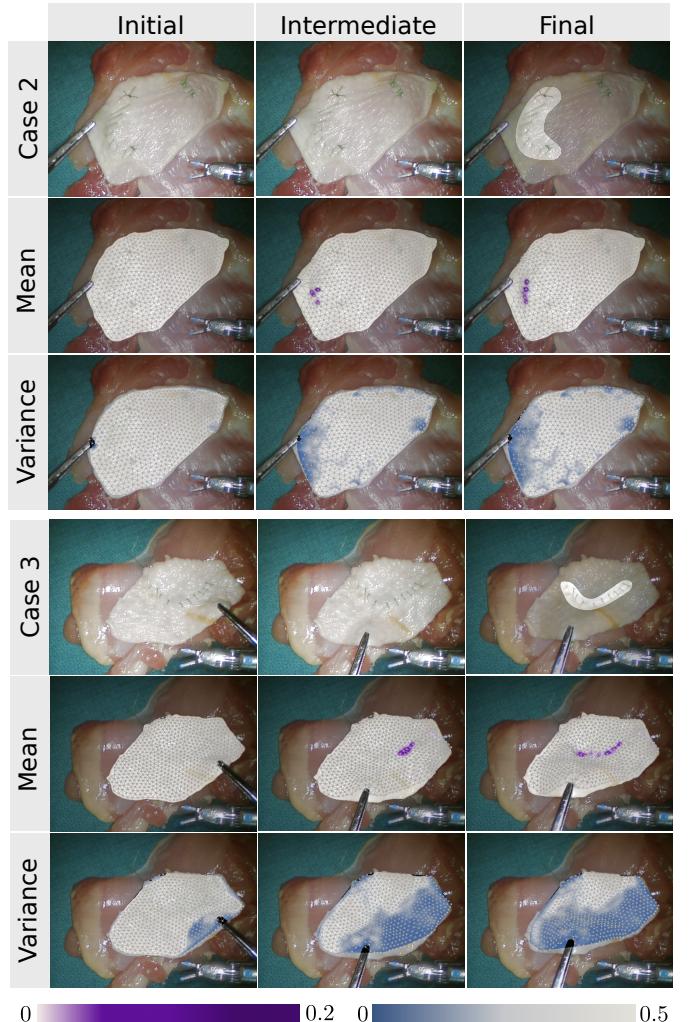


Fig. 8. This figure shows results from real-world tissue attachment point estimation experiments case 2 and case 3 (case 1 is shown in Fig. 1). Time progresses from left to right. A dark purple value corresponds to a stronger estimated attachment point, and a darker blue value corresponds to lower uncertainty about the boundary parameters at that region. Notice that as the tissue is deformed, the variance decreases, and the estimated mean of our boundary matches closer to the ground truth.

case 2, there is only a single grasp sequence. In case 3, we apply two grasp sequences to the same piece of chicken skin tissue by resetting and re-grasping when the algorithm cannot recover more information from the first grasp point without violating safety constraints. We again compare against the Adam optimizer baseline, along with another deep neural network approach, BANet [44]. The neural network takes in a 3D volume representing the tissues and its deformation field at the last time point of each test cases as inputs, classifying each voxel as boundary or not. We use the pre-trained weights from the original BANet work. More details on our adaptation of BANet is discussed in Appendix VIII-C.

Quantitative evaluations are done using the PCD and PUG metrics with up to two dilation iterations (about 5 mm expansion). We show the performance of the final estimation of each case in Table II. Because our method produces meaningful uncertainty, we also report our method's accuracy of results

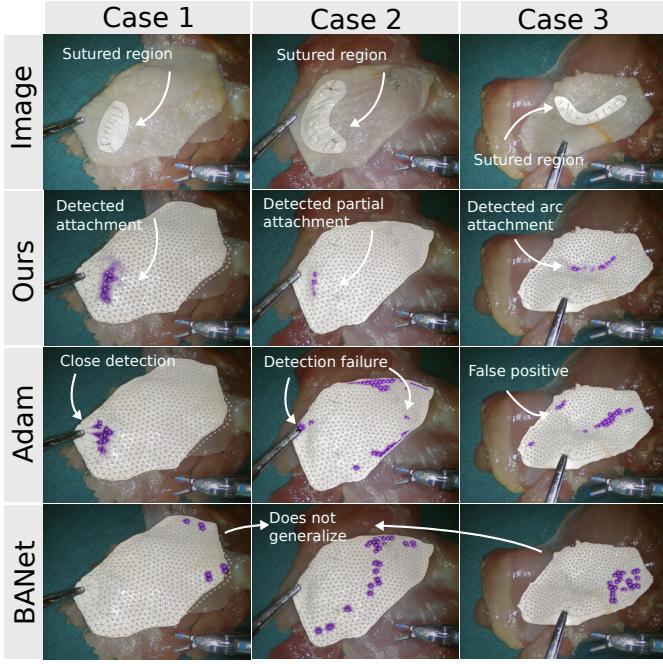


Fig. 9. Detailed comparison of estimated attachment points between the proposed method and the Adam optimizer. In the second row, our method successfully detects attachment shape. In comparison, the Adam optimizer estimates predictions that are close to the true region (case 1) but fails in other cases, producing many false positive estimations. Because BANet cannot generalize to our real-world data, it fails to predict meaningful results.

within a certainty threshold. We do this by excluding uncertain areas where the variance is larger than 0.1 when computing the accuracy metrics. In Table II, we denote it as “Ours- $\sigma=0.1$ ”. The PCD metric becomes better since our method is uncertain about some false positive results, and we are able to exclude them. A side effect is that the PUG metric for case 3 becomes lower as some true positive results are uncertain.

We also show example results from our proposed approach in case 1 in Fig. 1 and case 2 & 3 in Fig. 8. In case 2, the attached region is small and close to the grasp point. When the tissue is lifted up, the estimated results gradually uncover the part of the attached region. A larger mean value is shown using darker purple and corresponds to a stronger estimated attachment point. While the magnitude of the estimated mean is proportional to the strength of the attachment point, the estimated boundary parameters are only pertinent to the XPBD simulation and do not correspond to any physical units. A lower variance value is shown using darker blue and corresponds to an increased confidence in the attachment strength of a region. The final variance is relatively low around the grasping point compared to other not-displaced regions, and the estimated boundary points line up with the true sutures. In case 3, the attachment region has a larger arc shape and is further from the grasping point. Over time, our estimation improves as uncertainty gradually decreases across the tissue, and the mean value of the boundary parameters is correctly identified near the attachment points.

Our methods outperforms Adam optimizer in terms of PCD,

TABLE II  
RESULTS FROM OUR REAL WORLD ESTIMATION EXPERIMENT. THE NUMBER AFTER THE METRICS CORRESPONDS TO HOW MUCH DILATION IS APPLIED WHEN COMPUTING THE METRIC.

	PCD ↑	PCD-1	PCD-2	PUG ↑	PUG-1	PUG-2
Case1-BANet	0	0	0	0	0	0
Case1-Adam	46.2	<b>84.6</b>	84.6	54.5	100.0	100.0
Case1-Ours	<b>47.4</b>	84.2	<b>89.5</b>	<b>81.8</b>	100.0	100.0
Case1-Ours- $\sigma=0.1$	60	93.3	100	81.8	100.0	100.0
Case2-BANet	0	0	7.1	0	0	7.5
Case2-Adam	0.0	2.1	10.4	0.0	2.5	17.5
Case2-Ours	<b>33.3</b>	<b>83.3</b>	<b>100.0</b>	<b>5.0</b>	<b>27.5</b>	<b>37.5</b>
Case2-Ours- $\sigma=0.1$	40.0	100	100.0	11.1	55.6	72.2
Case3-BANet	8.0	20.0	32.0	4.2	14.5	22.5
Case3-Adam	<b>45.2</b>	61.3	67.7	<b>42.4</b>	<b>81.8</b>	97.0
Case3-Ours	44.5	<b>72.2</b>	<b>88.9</b>	24.2	78.8	<b>100.0</b>
Case3-Ours- $\sigma=0.1$	40.0	70.0	100.0	14.8	37.0	63.0

especially in the hardest case 2; this is because our method is less prone to false positives in the estimation, demonstrating the EKF’s ability to handle noisy observations. This is illustrated in Fig. 9 as more false positive results appear for the Adam optimizer. For PUG, our method perform comparably to the Adam optimizer but is better in case 2. Shown in Fig. 9, the comparison method is not able to predict attachment close to the sutured region whereas ours can. BANet performs relatively poorly in our experiments as shown in both Table II and the last row of Fig. 9. This is possibly because of its inability to generalize to out-of-distribution targets and the real-to-sim gap between our data and the simulation data that it was trained on.

#### F. Computation Performance

We report the average runtime of each component of our method. In all experiments, tissues are simulated with 600 particles. The update step of our estimation framework requires 0.25s to update belief distribution. Using the multiple shooting metric with 3 samples increases the runtime to 0.36s. A single optimization iteration of our local method takes 0.11s, for LG-D, and 0.8s, for LG-H. SL-D returns an action after 2.9s when run for 5 optimization iterations with 100 samples. These numbers support the use of JIGGLE online. Additionally, parallel and asynchronous computation can be incorporated to achieve further performance gain.

## VI. CONCLUSION

In this paper, we proposed JIGGLE: a novel estimation and active sensing framework to identify attachment points in a deformable surgical environment. We provide a complete framework for boundary parameter identification from vision to estimation to active control. With our probabilistic approach, we were able to provide an estimation method that is robust to the noisiness of real data and an active sensing method to maximize information gain safely. By providing this robust fully integrated framework and demonstrating its effectiveness even with topological changes, we take large strides towards improving the field of surgical automation.

Despite our choice of XPBD, our approach is not particular to the choice of differentiable simulator. Though we do not explicitly consider test cases where the true strength of the attachment regions varies, our framework supports the estimation of continuous parameters capable of identifying boundary parameters of varying stiffness. While we demonstrate our method on homogenous tissues, with uniform attributes like tissue stiffness, our method is able to handle small amounts of variability in the tissue parameters through the noise term in the EKF. This is showcased in our real world experiments as real tissues, like chicken skin, show natural variation in their tissue attributes. In future work we seek to further explore the challenges presented by heterogeneous tissues through simultaneous estimation of tissue and boundary parameters. Finally, while we primarily focus on the surgical use case, our method can be applied to other general thin shell deformables, such as household tasks that involve manipulating cloth. In doing so, our method lays the groundwork for a generalizable approach for manipulators to achieve complex cutting and forming of other soft bodies.

Though the framework is evaluated on thin-shell subjects, we believe it can potentially work for volumetric deformable objects as well. The major challenge will be decoupling tissues' internal deformation and deformation caused by boundary attachments. To do this, we plan on adding additional active sensing objectives in the future to encourage direct observation of possible attachment regions. Despite the efficacy of our results, the imperfect nature of our estimation stems from the real-to-sim gap [22] and discretization errors. In future works, we also seek to address these to further improve our results through better system identification and model refinement.

## VII. ACKNOWLEDGEMENTS

This work is supported by the US Army Telemedicine and Advanced Technologies Research Center (TATRC) and NSF CAREER award 2045803. Nikhil Uday Shinde is supported by the National Defense Science and Engineering Graduate (NDSEG) Fellowship Program. The authors would like to thank Dylan Hirsch and Azra Begzadić for valuable discussions on the proofs presented in the Appendix.

## REFERENCES

- [1] Arash Asgharivaskasi, Shumon Koga, and Nikolay Atanasov. Active mapping via gradient ascent optimization of shannon mutual information over continuous  $se(3)$  trajectories. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12994–13001, 2022. doi: 10.1109/IROS47612.2022.9981875.
- [2] Pasu Boonvisut and M Cenk Çavuşoğlu. Identification and active exploration of deformable object boundary constraints through robotic manipulation. *The International journal of robotics research*, 33(11):1446–1461, 2014.
- [3] Hsiao-yu Chen, Edith Tretschk, Tuur Stuyck, Petr Kadlec, Ladislav Kavan, Etienne Vouga, and Christoph Lassner. Virtual elastic objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15827–15837, 2022.
- [4] Zih-Yun Chiu, Albert Z. Liao, Florian Richter, Bjorn Johnson, and Michael C. Yip. Markerless suture needle 6d pose tracking with robust uncertainty estimation for autonomous minimally invasive robotic surgery. *CorR*, abs/2109.12722, 2021. URL <https://arxiv.org/abs/2109.12722>.
- [5] Tao Du, Kui Wu, Pingchuan Ma, Sebastien Wah, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. Diffpd: Differentiable projective dynamics. *ACM Trans. Graph.*, 41(2), nov 2021. ISSN 0730-0301. doi: 10.1145/3490168. URL <https://doi.org/10.1145/3490168>.
- [6] Mathieu Dubied, Mike Yan Michelis, Andrew Spielberg, and Robert Kevin Katzschmann. Sim-to-real for soft robots using differentiable fem: Recipes for meshing, damping, and actuation. *IEEE Robotics and Automation Letters*, 7(2):5015–5022, 2022. doi: 10.1109/LRA.2022.3154050.
- [7] Animesh Garg, Siddarth Sen, Rishi Kapadia, Yiming Jen, Stephen McKinley, Lauren Miller, and Ken Goldberg. Tumor localization using automated palpation with gaussian process adaptive sampling. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 194–200. IEEE, 2016.
- [8] Jiawei Ge, Michael Kam, Justin D. Opfermann, Hamed Saeidi, Simon Leonard, Leila J. Mady, Martin J. Schnermann, and Axel Krieger. Autonomous system for tumor resection (astr) - dual-arm robotic midline partial glossectomy. *IEEE Robotics and Automation Letters*, 9(2):1166–1173, 2024. doi: 10.1109/LRA.2023.3341773.
- [9] Tipakorn Greigarn, Michael S Branicky, and M Cenk Çavuşoğlu. Task-oriented active sensing via action entropy minimization. *IEEE Access*, 7:135413–135426, 2019.
- [10] Eric Heiden, Miles Macklin, Yashraj Narang, Dieter Fox, Animesh Garg, and Fabio Ramos. Disect: A differentiable simulation engine for autonomous robotic cutting. *arXiv preprint arXiv:2105.12244*, 2021.
- [11] Eric Heiden, Christopher E Denniston, David Millard, Fabio Ramos, and Gaurav S Sukhatme. Probabilistic inference of simulation parameters via parallel differentiable simulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3638–3645. IEEE, 2022.
- [12] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. Diffitaichi: Differentiable programming for physical simulation. *arXiv preprint arXiv:1910.00935*, 2019.
- [13] Zhe Hu, Tao Han, Peigen Sun, Jia Pan, and Dinesh Manocha. 3-d deformable object manipulation using deep neural networks. *IEEE Robotics and Automation Letters*, 4(4):4255–4261, 2019.

- [14] Maani Ghaffari Jadidi, Jaime Valls Miro, and Gamini Dissanayake. Mutual information-based exploration on continuous occupancy maps. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6086–6092. IEEE, 2015.
- [15] Peter Kazanzides, Zihan Chen, Anton Deguet, Gregory S. Fischer, Russell H. Taylor, and Simon P. DiMaio. An open-source research kit for the da vinci® surgical system. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6434–6439, 2014. doi: 10.1109/ICRA.2014.6907809.
- [16] Peter Kazanzides, Zihan Chen, Anton Deguet, Gregory S. Fischer, Russell H. Taylor, and Simon P. DiMaio. An open-source research kit for the da vinci® surgical system. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6434–6439, 2014. doi: 10.1109/ICRA.2014.6907809.
- [17] Ben Kehoe, Gregory Kahn, Jeffrey Mahler, Jonathan Kim, Alex Lee, Anna Lee, Keisuke Nakagawa, Sachin Patil, W. Douglas Boyd, Pieter Abbeel, and Ken Goldberg. Autonomous multilateral debridement with the raven surgical robot. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1432–1439, 2014. doi: 10.1109/ICRA.2014.6907040.
- [18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.
- [19] Michail Kontitsis, Evangelos A. Theodorou, and Emanuel Todorov. Multi-robot active slam with relative entropy optimization. In *2013 American Control Conference*, pages 2757–2764, 2013. doi: 10.1109/ACC.2013.6580252.
- [20] Yang Li, Florian Richter, Jingpei Lu, Emily K. Funk, Ryan K. Orosco, Jianke Zhu, and Michael C. Yip. Super: A surgical perception framework for endoscopic tissue manipulation with surgical robotics. *IEEE Robotics and Automation Letters*, 5(2):2294–2301, April 2020. ISSN 2377-3774. doi: 10.1109/lra.2020.2970659. URL <http://dx.doi.org/10.1109/LRA.2020.2970659>.
- [21] Yifei Li, Tao Du, Kui Wu, Jie Xu, and Wojciech Matusik. Diffcloth: Differentiable cloth simulation with dry frictional contact. *ACM Trans. Graph.*, 42(1), oct 2022. ISSN 0730-0301. doi: 10.1145/3527660. URL <https://doi.org/10.1145/3527660>.
- [22] Xiao Liang, Fei Liu, Yutong Zhang, Yuelei Li, Shan Lin, and Michael Yip. Real-to-sim deformable object manipulation: Optimizing physics models with residual mappings for robotic surgery. *arXiv preprint arXiv:2309.11656*, 2023.
- [23] Shan Lin, Albert J. Miao, Jingpei Lu, Shunkai Yu, Zih-Yun Chiu, Florian Richter, and Michael C. Yip. Semantic-super: A semantic-aware surgical perception framework for endoscopic tissue identification, reconstruction, and tracking. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4739–4746, 2023. doi: 10.1109/ICRA48891.2023.10160746.
- [24] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *International Conference on 3D Vision (3DV)*, 2021.
- [25] Fei Liu, Zihan Li, Yunhai Han, Jingpei Lu, Florian Richter, and Michael C. Yip. Real-to-sim registration of deformable soft tissue with position-based dynamics for surgical robot autonomy. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12328–12334, 2021. doi: 10.1109/ICRA48506.2021.9561177.
- [26] Fei Liu, Entong Su, Jingpei Lu, Mingen Li, and Michael C. Yip. Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics. *IEEE Robotics and Automation Letters*, 8(7):3964–3971, 2023. doi: 10.1109/LRA.2023.3264766.
- [27] Yonghao Long, Zhaoshuo Li, Chi Hang Yee, Chi-Fai Ng, Russell H. Taylor, Mathias Unberath, and Qi Dou. E-DSSR: efficient dynamic surgical scene reconstruction with transformer-based stereoscopic depth perception. *CoRR*, abs/2107.00229, 2021. URL <https://arxiv.org/abs/2107.00229>.
- [28] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. Xpb: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*, pages 49–54, 2016.
- [29] Lyudmila Mihaylova, Tine Lefebvre, Herman Bruyninx, Klaas Gadeyne, and Joris De Schutter. A comparison of decision making criteria and optimization methods for active robotic sensing. In *Numerical Methods and Applications: 5th International Conference, NMA 2002 Borovets, Bulgaria, August 20–24, 2002 Revised Papers 5*, pages 316–324. Springer, 2003.
- [30] Lauren M. Miller and Todd D. Murphrey. Trajectory optimization for continuous ergodic exploration on the motion group  $se(2)$ . In *52nd IEEE Conference on Decision and Control*, pages 4517–4522, 2013. doi: 10.1109/CDC.2013.6760585.
- [31] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.
- [32] Braden P Murphy and Farshid Alambeigi. A surgical robotic framework for safe and autonomous data-driven learning and manipulation of an unknown deformable tissue with an integrated critical space. *Journal of Medical Robotics Research*, 8(01n02):2340001, 2023.
- [33] Igor Peterlik, Hadrien Courtecuisse, Christian Duriez, and Stéphane Cotin. Model-based identification of anatomical boundary conditions in living tissues. In *International Conference on information processing in computer-assisted interventions*, pages 196–205. Springer, 2014.

- [34] Florian Richter, Shihao Shen, Fei Liu, Jingbin Huang, Emily K. Funk, Ryan K. Orosco, and Michael C. Yip. Autonomous robotic suction to clear the surgical field for hemostasis using image-based blood flow detection. *IEEE Robotics and Automation Letters*, 6(2):1383–1390, 2021. doi: 10.1109/LRA.2021.3056057.
- [35] Allison Ryan and J. Karl Hedrick. Particle filter based information-theoretic active sensing. *Robotics and Autonomous Systems*, 58(5):574–584, 2010. ISSN 0921-8890. doi: <https://doi.org/10.1016/j.robot.2010.01.001>. URL <https://www.sciencedirect.com/science/article/pii/S0921889010000023>.
- [36] H. Saeidi, H. N. D. Le, J. D. Opfermann, S. Leonard, A. Kim, M. H. Hsieh, J. U. Kang, and A. Krieger. Autonomous laparoscopic robotic suturing with a novel actuated suturing tool and 3d endoscope. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1541–1547, 2019. doi: 10.1109/ICRA.2019.8794306.
- [37] Hadi Salman, Elif Ayvali, Rangaprasad Arun Srivatsan, Yifei Ma, Nicolas Zevallos, Rashid Yasin, Long Wang, Nabil Simaan, and Howie Choset. Trajectory-optimized sensing for active search of tissue abnormalities in robotic surgery. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5356–5363, 2018. doi: 10.1109/ICRA.2018.8460936.
- [38] Brent Schlotfeldt, Dinesh Thakur, Nikolay Atanasov, Vijay Kumar, and George J. Pappas. Anytime planning for decentralized multirobot active information gathering. *IEEE Robotics and Automation Letters*, 3(2):1025–1032, 2018. doi: 10.1109/LRA.2018.2794608.
- [39] Tim Schneider, Boris Belousov, Georgia Chalvatzaki, Diego Romeres, Devesh K. Jha, and Jan Peters. Active exploration for robotic manipulation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9355–9362, 2022. doi: 10.1109/IROS47612.2022.9982061.
- [40] Azad Shademan, Ryan S. Decker, Justin D. Opfermann, Simon Leonard, Axel Krieger, and Peter C. W. Kim. Supervised autonomous robotic soft tissue surgery. *Science Translational Medicine*, 8(337):337ra64–337ra64, 2016. doi: 10.1126/scitranslmed.aad9398. URL <https://www.science.org/doi/abs/10.1126/scitranslmed.aad9398>.
- [41] Tuur Stuyck and Hsiao-yu Chen. Diffxpbd: Differentiable position-based simulation of compliant constraint dynamics. *Proc. ACM Comput. Graph. Interact. Tech.*, 6(3), aug 2023. doi: 10.1145/3606923. URL <https://doi.org/10.1145/3606923>.
- [42] Zachary Sunberg and Mykel Kochenderfer. Online algorithms for pomdps with continuous state, action, and observation spaces. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 28, pages 259–263, 2018.
- [43] Priya Sundaresan, Rika Antonova, and Jeannette Bohg. Diffcloud: Real-to-sim from point clouds with differentiable simulation and rendering of deformable objects. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10828–10835, 2022. doi: 10.1109/IROS47612.2022.9981101.
- [44] Eleonora Tagliabue, Diego Dall’Alba, Micha Pfeiffer, Marco Piccinelli, Riccardo Marin, Umberto Castellani, Stefanie Speidel, and Paolo Fiorini. Data-driven intraoperative estimation of anatomical attachments for autonomous tissue dissection. *IEEE Robotics and Automation Letters*, 6(2):1856–1863, 2021. doi: 10.1109/LRA.2021.3060655.
- [45] Eleonora Tagliabue, Marco Piccinelli, Diego Dall’Alba, Juan Verde, Micha Pfeiffer, Riccardo Marin, Stefanie Speidel, Paolo Fiorini, and Stéphane Cotin. Intraoperative update of boundary conditions for patient-specific surgical simulation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part IV 24*, pages 373–382. Springer, 2021.
- [46] Bao Thach, Brian Y Cho, Alan Kuntz, and Tucker Hermans. Learning visual shape control of novel 3d deformable objects from partial-view point clouds. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8274–8281. IEEE, 2022.
- [47] Brijen Thananjeyan, Animesh Garg, Sanjay Krishnan, Carolyn Chen, Lauren Miller, and Ken 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)*, pages 2371–2378, 2017. doi: 10.1109/ICRA.2017.7989275.
- [48] Yuehao Wang, Yonghao Long, Siu Hin Fan, and Qi Dou. Neural rendering for stereo 3d reconstruction of deformable tissues in robotic surgery. In *International conference on medical image computing and computer-assisted intervention*, pages 431–441. Springer, 2022.
- [49] Tian Xu, Yong Lei, XiaoLiang Cheng, and Murong Li. Identification of young’s modulus and equivalent spring constraint boundary conditions of the soft tissue with locally observed displacements for endoscopic liver surgery. *Computer Methods in Biomechanics and Biomedical Engineering*, 25(4):439–454, 2022.
- [50] Shiyi Yang, Soo Jeon, and Jongeun Choi. Efficient sampling for rapid estimation of 3-d stiffness distribution via active tactile exploration. *IEEE/ASME Transactions on Mechatronics*, 25(4):1729–1738, 2020. doi: 10.1109/TMECH.2020.2992573.
- [51] Yutong Zhang, Fei Liu, Xiao Liang, and Michael Yip. Achieving autonomous cloth manipulation with optimal control via differentiable physics-aware regularization and safety constraints. *arXiv preprint arXiv:2309.11655*, 2023.

### VIII. APPENDIX

#### A. Relationship Between Uncertainty-Weighted Displacement (UWD) and Boundary Entropy

In this section we present the following proposition, and provide a proof tying the relationship between the UWD, i.e., displacement,  $\mathcal{D}$ , of the tissue particles from their initial position, and the reduction in boundary entropy  $\mathcal{H}(\mathbf{b}_{t+1})$  at the next timestep  $t + 1$ .

**Proposition 1.** *The active sensing objective of maximizing the uncertainty-weighted deformation displacement, i.e.,  $\mathcal{D}$  in Eq. 17, is designed to decrease the entropy  $\mathcal{H}(\mathbf{b}_{t+1})$  of boundary estimation and uncover more unknown information. That is,*

$$\min \mathcal{H}(\mathbf{b}_{t+1}) \propto \max \mathcal{D} \quad (27)$$

■

To prove it, we first establish a connection to entropy,  $\mathcal{H}$ , via the observation Jacobian  $J_{t+1} = \frac{\partial h}{\partial \mathbf{b}}$ , defined in Eq. 10 of the EKF equations in section VIII-A1. Second, we derive the relationship between the Jacobian  $J$  and the tissue displacement in section VIII-A2. Third, we combine these results to show proposition 1 in section VIII-A3.

1) **Boundary Entropy Regarding the Observation Jacobian:** As defined in Eq. 15, it is evident that the entropy  $\mathcal{H}(\mathbf{b}_{t+1})$  is directly proportional to the determinant of the covariance matrix outputted by the Extended Kalman Filter (EKF) estimator. Since  $\Sigma_{t|t}, W_t, V_t$  are all positive semi-definite covariance matrices, they allow the following factorization:

$$\Sigma_{t+1|t} = \Sigma_{t|t} + W_t = \Sigma^{\frac{1}{2}} \Sigma^{\frac{1}{2}} \quad (28)$$

Combining this fact with the Extended Kalman Filter (EKF) equations, as defined in Eq. 7 and 8, and for the sake of simplicity, neglecting the subscripts of observation Jacobian  $J_{t+1}$  and Kalman gain  $K_{t+1}$ , we obtain:

$$\begin{aligned} \mathcal{H}(\mathbf{b}_{t+1}) &\propto |\Sigma_{t+1|t}| \\ &\propto |(I - KJ)\Sigma_{t+1|t}| \\ &\propto |(I - \Sigma_{t+1|t} J^\top [J\Sigma_{t+1|t} J^\top + V_t]^{-1} J)\Sigma_{t+1|t}| \\ &\propto |(I - \Sigma^{\frac{1}{2}} \Sigma^{\frac{1}{2}} J^\top [J\Sigma^{\frac{1}{2}} \Sigma^{\frac{1}{2}} J^\top + V_t]^{-1} J)\Sigma^{\frac{1}{2}} \Sigma^{\frac{1}{2}}| \\ &\propto |(I - \Sigma^{\frac{1}{2}} J^\top [J\Sigma^{\frac{1}{2}} \Sigma^{\frac{1}{2}} J^\top + V_t]^{-1} J\Sigma^{\frac{1}{2}})| |\Sigma^{\frac{1}{2}} \Sigma^{\frac{1}{2}}| \end{aligned} \quad (29)$$

Denoting  $\mathbf{A} = J\Sigma^{\frac{1}{2}}$  and  $\Sigma_{t+1|t} = \Sigma = \Sigma^{\frac{1}{2}} \Sigma^{\frac{1}{2}}$ , we can perform singular value decomposition (SVD) where  $D$  is a matrix with singular values  $\lambda_i$  on its diagonal,  $UU^\top = U^\top U = I$ , and  $QQ^\top = Q^\top Q = I$ .

$$\mathbf{A} = J\Sigma^{\frac{1}{2}} = U D Q^\top \quad (30)$$

For simplification and to maintain a straightforward representation of observation uncertainty, we assume  $V_t = \alpha I$ , with

$\alpha$  denotes the maximum expected observation noise variance. This diagonal covariance matrix implies that the observation noise is isotropic and of uniform magnitude across different dimension. Substituting them into Eq. 29, we get:

$$\begin{aligned} \mathcal{H}(\mathbf{b}_{t+1}) &\propto \\ &\propto |I - \mathbf{A}^\top [\mathbf{A}\mathbf{A}^\top + \alpha I]^{-1} \mathbf{A}| |\Sigma| \\ &\propto |QQ^\top - QD^\top U^\top [UDD^\top U^\top + \alpha I]^{-1} UDQ^\top| |\Sigma| \\ &\propto |Q| |I - D^\top U^\top [UDD^\top U^\top + \alpha I]^{-1} UD| |Q^\top| |\Sigma| \\ &\propto |Q| |Q^\top| |I - D^\top U^\top [UDD^\top U^\top + \alpha I]^{-1} UD| |\Sigma| \\ &\propto |I - D^\top U^\top [UDD^\top U^\top + \alpha I]^{-1} UD| |\Sigma| \\ &\propto |I - D^\top U^\top [UDD^\top U^\top + \alpha UU^\top]^{-1} UD| |\Sigma| \\ &\propto |I - D^\top U^\top U [DD^\top + \alpha I]^{-1} U^\top UD| |\Sigma| \\ &\propto |(I - D^\top (DD^\top + \alpha I)^{-1} D)| |\Sigma| \\ &\propto |\Sigma| \prod_i^N \left(1 - \frac{\lambda_i^2}{\lambda_i^2 + \alpha}\right) \end{aligned} \quad (31)$$

In Eq. 31 and Eq. 28, the only term influenced by our control actions at time  $t$  is the observation Jacobian  $J$ , as  $\Sigma$  only depends on the estimation from the previous step. These equations show that selecting  $J$  in a manner that increases the magnitudes of the singular values  $\|\lambda_i\|_2$  of  $J\Sigma^{\frac{1}{2}}$  will result in a reduction of the entropy within the belief distribution.

After establishing the correlation between the Jacobian and entropy reduction, we need to link the uncertainty-weighted displacement (UWD) of particles to the Jacobian. We must identify the choices of UWD that lead to entropy reduction. Demonstrating these connections will validate that our Proposition 27.

2) **Observation Jacobian Regarding the Uncertainty-Weighted Displacement (UWD):** To link the Jacobian,  $J$ , at the next timestep  $t + 1$ , to displacement we derive an analytical form of  $J$ . To derive the exact form of observation Jacobian, we illustrate the simulated mesh with a simplified graph in Fig. 10. Hence, we partition our whole tissue state representation  $\mathbf{x} = \{\mathbf{x}_{t+1}^B, \mathbf{x}_{t+1}^U\}$  into particles related to boundary constraints  $\mathbf{x}_{t+1}^B$  and other particles denoted as  $\mathbf{x}_{t+1}^U$ . Of the  $n$  total particles, we assume that there are  $m$  particles in  $\mathbf{x}_{t+1}^B$  and  $q$  particles in  $\mathbf{x}_{t+1}^U$  such that  $m + q = n$ . From the definition in Eq. 2, the XPBD simulation involves minimizing the total energy potential of constraints:

$$\arg \min_{\mathbf{x}} \mathcal{U}(\mathbf{x}, \mathbf{b}) = \arg \min_{\mathbf{x}} \mathcal{U}_1(\mathbf{x}_{t+1}^B, \mathbf{b}) + \mathcal{U}_2(\mathbf{x}_{t+1}^U, \mathbf{x}_{t+1}^B) \quad (32)$$

where the first term,  $\mathcal{U}_1(\mathbf{x}_{t+1}^U, \mathbf{x}_{t+1}^B)$ , representing the energy only related to the current boundary constraints,  $\mathbf{b}$  and  $\mathcal{U}_2(\mathbf{x}_{t+1}^U, \mathbf{x}_{t+1}^B)$  denotes the energy term associated with both  $\mathbf{x}_{t+1}^U$  and  $\mathbf{x}_{t+1}^B$ . In our XPBD simulator, all defined constraints

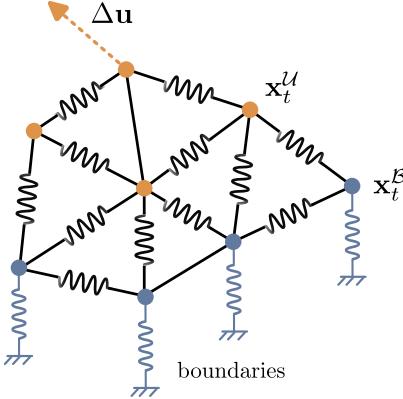


Fig. 10. This figure shows a simplified model of the XPBD simulation. The particles shown in blue,  $\mathbf{x}_{t+1}^B$ , are related to the boundary constraints. We are trying to estimate the boundary parameters associated with the boundary constraints on these particles. The remaining particles are shown in orange,  $\mathbf{x}_{t+1}^U$ . The virtual particle  $u$  is used to apply control on the tissue. All constraints are implemented as set distance constraints and are depicted using springs with different spring constants set according to the associated constraint.

are distance constraints between particles. For the boundary energy, we are able to write:

$$\begin{aligned}\mathcal{U}_1(\mathbf{x}_{t+1}^B, \mathbf{b}) &= \frac{1}{2} \mathbf{C}^\top(\mathbf{x}_{t+1}^B) \text{diag}(\mathbf{b}) \mathbf{C}(\mathbf{x}_{t+1}^B) \\ &= \frac{1}{2} \sum_{i=1}^m b_i \left( \left\| \mathbf{x}_{t+1}^{B,i} - \mathbf{x}_0^{B,i} \right\|_2 - d_0^{B,i} \right)^2\end{aligned}\quad (33)$$

$\mathbf{b} \in \mathbb{R}^m$  where  $b_i$  is the stiffness of the  $i$ -th boundary spring, of the total  $m$  springs.  $\mathbf{x}_{t+1}^{B,i} - \mathbf{x}_0^{B,i}$  is the displacement of particle  $i$  related to the boundary constraint from time 0 to the next time step  $t + 1$ .  $d_0^{B,i}$  is the length of the boundary spring associated with particle  $i$  at rest time,  $t = 0$ . For our work, we consider a zero length spring at rest,  $d_0^{B,i} = 0, \forall i$ , for all boundary springs. Similarly, we describe the rest of spring potential energy by:

$$\begin{aligned}\mathcal{U}_2(\mathbf{x}_{t+1}^U, \mathbf{x}_{t+1}^B) &= \frac{1}{2} \mathbf{C}^\top(\mathbf{x}_{t+1}^U, \mathbf{x}_{t+1}^B) \text{diag}(\mathbf{k}) \mathbf{C}(\mathbf{x}_{t+1}^U, \mathbf{x}_{t+1}^B) \\ &= \frac{1}{2} \sum_{j=1}^s k_j \left( \left\| \mathbf{x}_{t+1}^{j,1} - \mathbf{x}_{t+1}^{j,2} \right\| - d_0^{1,2,j} \right)^2\end{aligned}\quad (34)$$

where  $\mathbf{x}_{t+1}^{j,1}, \mathbf{x}_{t+1}^{j,2} \in \{\mathbf{x}_{t+1}^B \cup \mathbf{x}_{t+1}^U\}$  are a pair of two particles connected by the  $j$ -th spring within the simulated mesh.  $d_0^{1,2,j}$  is the distance of that  $j$ -th spring constraint between particle 1 and 2 at rest, at time  $t = 0$ . Using this, we are able to define the matrices  $\mathbf{L} \in \mathbb{R}^{3n \times 3n}$  and  $\mathbf{P} \in \mathbb{R}^{3n \times 3s}$  as follows:

$$\begin{aligned}\mathbf{L} &= \left( \sum_{j=1}^s k_j \mathbf{A}^j \mathbf{A}^{j,\top} \right) \otimes \mathbf{I}_3 \\ \mathbf{P} &= \left( \sum_{j=1}^s k_j \mathbf{A}^j \mathbf{S}^{j,\top} \right) \otimes \mathbf{I}_3 \\ \mathbf{d} &= \frac{\mathbf{x}_0^{j,1} - \mathbf{x}_0^{j,2}}{\left\| \mathbf{x}_0^{j,1} - \mathbf{x}_0^{j,2} \right\|_2} d_0^{1,2,j}\end{aligned}\quad (35)$$

Here,  $\mathbf{A}_j \in \mathbb{R}^n$  is the connectivity vector of the  $j$ -th spring, i.e.,  $\mathbf{A}^{j,1} = 1$ ,  $\mathbf{A}^{j,2} = -1$ , and zero otherwise.  $(j, 1)$  and  $(j, 2)$  denote the indices of the first and second particle, respectively, associated with the  $j$ -th spring constraint.  $\mathbf{S}^j \in \mathbb{R}^s$  is  $j$ -th spring indicator, i.e.,  $\mathbf{S}^j = \mathbf{1}_s$ ,  $s$  is the number of non-boundary related springs or edges, and  $\otimes$  denotes Kronecker product.  $\mathbf{d}$  can be interpreted as a spring in the rest pose, at time  $t = 0$ , that is rotated in a specific direction, effectively behaving like a constant. Eq. 36 can be re-factorized into:

$$\mathcal{U}_2(\mathbf{x}_{t+1}^U, \mathbf{x}_{t+1}^B) = \frac{1}{2} \mathbf{x}^\top \mathbf{L} \mathbf{x} - \mathbf{x}^\top \mathbf{P} \mathbf{d} \quad (36)$$

The solution to the minimization function described in Eq. 32 can be regarded to satisfy:

$$\nabla_{\mathbf{x}} \mathcal{U}(\mathbf{x}, \mathbf{b}) = \nabla_{\mathbf{x}} \mathcal{U}_1(\mathbf{x}_{t+1}^B, \mathbf{b}) + \nabla_{\mathbf{x}} \mathcal{U}_2(\mathbf{x}_{t+1}^U, \mathbf{x}_{t+1}^B) = 0 \quad (37)$$

Before moving forward we explicitly clarify the structure of our state  $\mathbf{x}$ . Assuming a system with 2 particles,  $\mathbf{x}_{t+1}^B = [\mathbf{x}_{t+1}^B[0], \mathbf{x}_{t+1}^B[1], \mathbf{x}_{t+1}^B[2]]^\top$  is the single particle related to boundary constraints and  $\mathbf{x}_{t+1}^U = [\mathbf{x}_{t+1}^U[0], \mathbf{x}_{t+1}^U[1], \mathbf{x}_{t+1}^U[2]]^\top$  is the single other particle.  $\mathbf{x}$  is a flattened joint state of the 2 particles in the following form:  $\mathbf{x} = [\mathbf{x}_{t+1}^B[0], \mathbf{x}_{t+1}^B[1], \mathbf{x}_{t+1}^B[2], \mathbf{x}_{t+1}^U[0], \mathbf{x}_{t+1}^U[1], \mathbf{x}_{t+1}^U[2]]^\top$ . For multiple particles we follow a similar structure where the flattened form of the boundary related particles is stacked above the flattened form of the other particles. This gives us:  $\mathbf{x}_{t+1}^B \in \mathbb{R}^{3m}$ ,  $\mathbf{x}_{t+1}^U \in \mathbb{R}^{3q}$ ,  $\mathbf{x} \in \mathbb{R}^{3n}$

**Lemma 1. [Implicit-Function Theorem]** Considering the solution function defined in Eq. 37,

$$\mathcal{L}(\mathbf{x}^*, \mathbf{b}) = \nabla_{\mathbf{x}} \mathcal{U}(\mathbf{x}^*, \mathbf{b}) = 0 \quad (38)$$

with  $\mathbf{x}^*$  defined as an equilibrium point, the sensitivities of the solution with respect to the parameter  $\mathbf{b}$  can be computed as:

$$\frac{\partial \mathbf{x}^*}{\partial \mathbf{b}} = - \left( \frac{\partial \mathcal{L}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathcal{L}}{\partial \mathbf{b}} \quad (39)$$

■

In our EKF formulation, the observation Jacobian is given by

$$J = \frac{\partial h}{\partial \mathbf{b}} = \frac{\partial f(\mathbf{x}^{\text{ref}}, u, \mathbf{b})}{\partial \mathbf{b}} = \frac{\partial \mathbf{x}}{\partial \mathbf{b}} \quad (40)$$

Combining it with the implicit-function theorem in Eq. 39, we obtain the Jacobian as,

$$\begin{aligned}J &= - \left( \frac{\partial \mathcal{L}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathcal{L}}{\partial \mathbf{b}} = - \left( \frac{\partial \nabla_{\mathbf{x}} \mathcal{U}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \nabla_{\mathbf{x}} \mathcal{U}}{\partial \mathbf{b}} \\ &= - \left( \frac{\partial^2 \mathcal{U}}{\partial \mathbf{x} \partial \mathbf{x}^\top} \right)^{-1} \frac{\partial^2 \mathcal{U}}{\partial \mathbf{b} \partial \mathbf{x}^\top}\end{aligned}\quad (41)$$

Given  $\mathbf{x} = \{\mathbf{x}_{t+1}^B, \mathbf{x}_{t+1}^U\}$ , we can computer the following gradients as,

$$\begin{aligned}\frac{\partial \mathcal{U}_1}{\partial \mathbf{x}} &= \left[ \Delta \mathbf{x}_{t+1}^B^\top \text{diag}(\mathbf{b}) \quad \mathbf{0} \right] \in \mathbb{R}^{3n \times 1} \\ \frac{\partial \mathcal{U}_2}{\partial \mathbf{x}} &= \left[ \begin{array}{c} \mathbf{x}_{t+1}^B^\top \mathbf{L}_{BB} + \mathbf{x}_{t+1}^U^\top \mathbf{L}_{UB} \\ \mathbf{x}_{t+1}^U^\top \mathbf{L}_{UU} + \mathbf{x}_{t+1}^B^\top \mathbf{L}_{BU} \\ - (\mathbf{Pd})^\top \end{array} \right]^\top \\ &\in \mathbb{R}^{3n \times 1} \\ \frac{\partial^2 \mathcal{U}_1}{\partial \mathbf{x} \partial \mathbf{x}^\top} &= \begin{bmatrix} \text{diag}(\mathbf{b}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{3n \times 3n} \quad (42) \\ \frac{\partial^2 \mathcal{U}_2}{\partial \mathbf{x} \partial \mathbf{x}^\top} &= \begin{bmatrix} \mathbf{L}_{BB} & \mathbf{L}_{BU} \\ \mathbf{L}_{UB} & \mathbf{L}_{UU} \end{bmatrix} = \mathbf{L} \in \mathbb{R}^{3n \times 3n} \\ \frac{\partial^2 \mathcal{U}_1}{\partial \mathbf{b} \partial \mathbf{x}^\top} &= \begin{bmatrix} \widehat{\Delta \mathbf{x}_{t+1}^B} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{3n \times m} \\ \frac{\partial^2 \mathcal{U}_2}{\partial \mathbf{b} \partial \mathbf{x}^\top} &= \mathbf{0} \in \mathbb{R}^{3n \times m} \\ \Delta \mathbf{x}_{t+1}^B &= (\mathbf{x}_{t+1}^B - \mathbf{x}_0^B) \in \mathbb{R}^{3m \times 1} \\ \widehat{\Delta \mathbf{x}_{t+1}^B} &= \sum_i^m \mathbf{e}_i \mathbf{e}_i^\top \otimes [\Delta \mathbf{x}_{t+1}^B]_i \in \mathbb{R}^{3m \times m} \quad (43)\end{aligned}$$

$\Delta \mathbf{x}_{t+1}^B$  is the displacement of all boundary related particles from time 0 to time  $t$ .  $\widehat{\Delta \mathbf{x}_{t+1}^B}$  is a concatenated matrix of the displacement of all boundary related particles.  $\mathbf{e}_i$  is the  $i$ -th column vector of the standard basis of  $\mathbb{R}^m$ , and  $[\Delta \mathbf{x}_{t+1}^B]_i \in \mathbb{R}^{3 \times 1}$  be the deformation displacement of the  $i$ -th boundary related particle from  $\Delta \mathbf{x}_{t+1}^B$ . Note that this is the same expression from the UWD. It should be noticed that  $\mathbf{L}$  is symmetric and positive semi-definite. Given these partials we get:

$$\begin{aligned}\frac{\partial^2 \mathcal{U}}{\partial \mathbf{b} \partial \mathbf{x}^\top} &= \frac{\partial^2 \mathcal{U}_1}{\partial \mathbf{b} \partial \mathbf{x}^\top} + \frac{\partial^2 \mathcal{U}_2}{\partial \mathbf{b} \partial \mathbf{x}^\top} \\ \frac{\partial^2 \mathcal{U}}{\partial \mathbf{x} \partial \mathbf{x}^\top} &= \frac{\partial^2 \mathcal{U}_1}{\partial \mathbf{x} \partial \mathbf{x}^\top} + \frac{\partial^2 \mathcal{U}_2}{\partial \mathbf{x} \partial \mathbf{x}^\top} \quad (44)\end{aligned}$$

Substituting the above into equation 39, we get an expression for the observation jacobian:

$$J = - \begin{bmatrix} \mathbf{L}_{BB} + \text{diag}(\mathbf{b}) & \mathbf{L}_{BU} \\ \mathbf{L}_{UB} & \mathbf{L}_{UU} \end{bmatrix}^{-1} \begin{bmatrix} \widehat{\Delta \mathbf{x}_{t+1}^B} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{3n \times m} \quad (45)$$

The first term in this expression is a function of the current boundary estimate  $\mathbf{b}$ , but is constant with respect to the particle states  $\mathbf{x}_t$ . We refer to this term as  $\mathcal{R}(\mathbf{b})$ :

$$\mathcal{R}(\mathbf{b}) = - \begin{bmatrix} \mathbf{L}_{BB} + \text{diag}(\mathbf{b}) & \mathbf{L}_{BU} \\ \mathbf{L}_{UB} & \mathbf{L}_{UU} \end{bmatrix}^{-1} \quad (46)$$

For the ease of notation we will denote that  $\mathbf{D} = \begin{bmatrix} \widehat{\Delta \mathbf{x}_{t+1}^B} & \mathbf{0} \end{bmatrix}^\top$ . Writing  $J = \mathcal{R}(\mathbf{b})\mathbf{D}$  we can see that the jacobian given a specific boundary estimate  $\mathbf{b}$  is a direct function of the displacement.

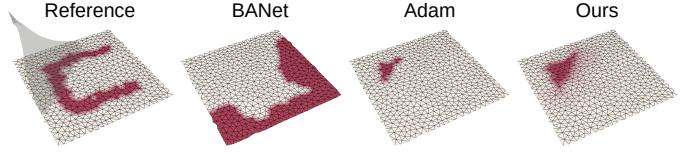


Fig. 11. Estimated attachment regions of different method on test case *U*-shape. BANet is trained to estimate hard boundary conditions therefore does not generalize to our data.

**3) Boundary entropy Regarding the UWD:** From section VIII-A1 and section VIII-A2, we see that to minimize entropy we have to maximize the singular values,  $\lambda_i^2$ , of  $\mathcal{R}(\mathbf{b}_t)\mathbf{D}\Sigma^{\frac{1}{2}}$ . Where  $\mathcal{R}(\mathbf{b}_t)$  is a constant given the belief at time  $t$ , and  $\Sigma^{\frac{1}{2}}$  is a constant at time  $t$ .

The Frobenius norm of a matrix  $\|\mathcal{R}(\mathbf{b}_t)\mathbf{D}\Sigma^{\frac{1}{2}}\| = \sqrt{\sum_i^n \lambda_i^2}$ . Thus maximizing the frobenius norm is equivalent to maximizing the magnitude of the  $\lambda_i^2$  and decreasing entropy. Leveraging the submultiplicative property of the frobenius norm, the known invertibility of matrix  $\mathcal{R}(\mathbf{b}_t)$ , and that  $\mathcal{R}(\mathbf{b}_t), \Sigma^{\frac{1}{2}}$  are constant matrices we can write:

$$\begin{aligned}\|\mathcal{R}(\mathbf{b}_t)^{-1}\mathcal{R}(\mathbf{b}_t)\mathbf{D}\Sigma^{\frac{1}{2}}\Sigma^{\frac{1}{2}}\| &\leq \|\mathcal{R}(\mathbf{b}_t)^{-1}\| \|\mathcal{R}(\mathbf{b}_t)\mathbf{D}\Sigma^{\frac{1}{2}}\Sigma^{\frac{1}{2}}\| \\ \|\mathbf{D}\Sigma\| &\leq \|\mathcal{R}(\mathbf{b}_t)^{-1}\| \|\mathcal{R}(\mathbf{b}_t)\mathbf{D}\Sigma^{\frac{1}{2}}\Sigma^{\frac{1}{2}}\| \\ \frac{\|\mathbf{D}\Sigma\|}{\|\mathcal{R}(\mathbf{b}_t)^{-1}\| \|\Sigma^{\frac{1}{2}}\|} &\leq \|\mathcal{R}(\mathbf{b}_t)\mathbf{D}\Sigma^{\frac{1}{2}}\| \\ \eta \|\mathbf{D}\Sigma\| &\leq \|\mathcal{R}(\mathbf{b}_t)\mathbf{D}\Sigma^{\frac{1}{2}}\| \\ \eta &= \frac{1}{\|\mathcal{R}(\mathbf{b}_t)^{-1}\| \|\Sigma^{\frac{1}{2}}\|} \quad (47)\end{aligned}$$

Note that the Uncertainty weighted displacement (UWD) is defined as  $\mathcal{D} = \|\mathbf{D}\Sigma\|$ .

Combining Eq. 31, Eq. 45 and Eq. 47, we show that at a given timestep  $t$ , by maximizing this lower bound  $\|\mathbf{D}\Sigma\|$  we can maximize  $\mathcal{R}(\mathbf{b}_t)\mathbf{D}\Sigma^{\frac{1}{2}}$  and thus increase the magnitude of the singular values  $\lambda_i^2$  and decrease entropy.

With this we have shown proposition 1: The active sensing objective of maximizing UWD is designed to decrease entropy  $\mathcal{H}(\mathbf{b}_{t+1})$ .

### B. Entropy loss implementation Details

In this section we discuss the implementation details for the entropy based loss from Eq. 22. This loss is optimized in our active sensing algorithm, LG-H, to solve for the best next action.

The entropy of our belief is specified in Eq. 15 as :

$$\mathcal{H}(\mathbf{b}_{t+1}) \propto \ln(|\Sigma_{t+1}|) \quad (48)$$

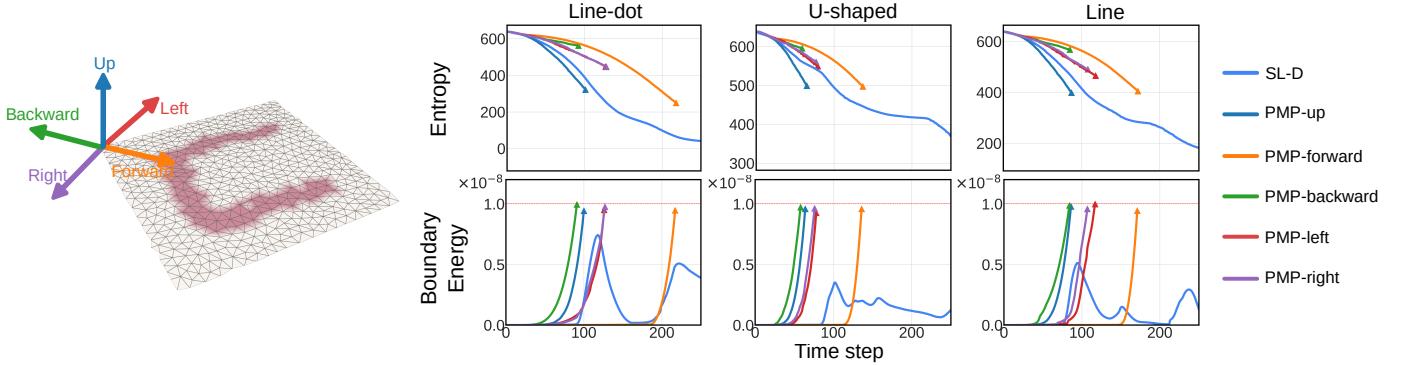


Fig. 12. Additional comparison of active sensing performance between all choices of predefined motion primitives (PMP). They are *up*, *forward*, *backward*, *left*, *right*. Results from our proposed Sampling-based Large step controller (SL-D) are included for comparsion. All PMPs result in safety violation. Before violation, *up* and *forward* achieve more entropy reduction, but *up* achieves that faster in general. In comparsion, the proposed method outperform all PMPs, achieving more entropy reduction at the cost of smaller boundary energy.

TABLE III

ADDITIONAL COMPARISON OF ESTIMATION ACCURACY IN SIMULATED CASES BETWEEN BANET, THE ADAM OPTIMIZER, AND THE PROPOSED METHOD. ONLY ONE LIFT-UP ACTION IS APPLIED TO ALL TISSUE BECAUSE BANET ONLY HANDLES A SINGLE INPUT.

Cases	PCD ↑			PUG ↑		
	BANet	Adam	Ours	BANet	Adam	Ours
Arc	10.7	75.0	100.0	37.0	5.6	14.8
Arc-line	9.4	75.0	100.0	18.2	5.5	16.4
Line	2.5	75.0	100.0	16.7	10.0	16.7
Line-dot	6.4	/	100.0	13.3	0.0	5.1
U-shape	2.3	80.0	100.0	4.8	11.5	22.1

The  $\ln$  makes directly optimizing this quantity in our loss function challenging. As uncertainty decreases and  $|\Sigma_{t+1}|$  approaches 0,  $\ln(|\Sigma_{t+1}|)$  tends to  $-\infty$ . This makes it hard to set the weights to balance this loss with the other metrics from Eq. 22 during our unconstrained optimization.

As a result instead of directly minimizing entropy,  $\mathcal{H}(\mathbf{b}_{t+1})$ , we minimize the term:

$$|I - K_{t+1}J_{t+1}| \quad (49)$$

From equations Eq. 15 and Eq. 8 we get that:  $\mathcal{H}(\hat{\mathbf{b}}_{t+1}) \propto \ln(|(I - K_{t+1}J_{t+1})\Sigma_{t+1|t}|)$ . We can remove  $\ln$  as  $\ln(x)$  is monotonically increasing for  $x > 0$ . Thus minimizing  $x$  is equivalent to minimizing  $\ln(x)$ .  $\Sigma_{t+1|t}$  is a constant at the given time step and the only term we can impact with our actions is  $K_{t+1}$ . Additionally  $0 \leq |I - K_{t+1}J_{t+1}| \leq 1$ , making this term easier to weight in an unconstrained optimization. As a result we choose Eq. 49 in place of entropy when implementing our entropy based loss.

### C. Additional Estimation Results

**BANet [44] Adaption and Comparison.** In the main text, we only compared against previous method BANet on the real world data. Here, we provide additional information about how BANet was adapted for evaluation against our framework

and also show additional comparisons against BANet on simulation data.

BANet takes in a 3D volume representing the tissues and its deformation field at the end of each test cases as inputs. It then classifies if each voxel is boundary or not. We convert our triangular mesh to be volumetric by marking the closest voxel to each of our mesh particles to be part of the volumetric representation. The mesh's particle displacement can be calculated with our proposed method (i.e.  $\mathbf{x}_{\text{end}}^{\text{ref}} - \mathbf{x}_0^{\text{ref}}$ ). Assuming knowledge of mesh particles' displacement, A displacement field is computed by assigning the mesh particles' displacement to the closest voxel and setting the displacement everywhere else to be zero. If the network output (after a sigmoid operation) on a voxel is larger than a threshold, we treat it as a boundary. We convert the network output (after a sigmoid operation) to binary boundary representations with a threshold of 0.1. We assign the threshold to be small (0.1) to include more relatively low-scored results from BANet because it assumes hard boundary conditions whereas our scenarios do not rely on that specific assumption. Lastly, boundary condition of mesh particles are determined by checking the BANet outputs corresponding to their closest voxel.

Additional comparisons between our method and BANet are shown in Table III. Looking at the PCD metric, BANet predicts inaccurate results. Fig. 11 explaines it by showing that BANet tends to regard not-displaced particles as boundaries. In fact, if we increase the binary threshold to 0.5, BANet predicts few boundary conditions. Note that this is correct for BANet because it assumes hard boundary conditions in its training data whereas in our test cases there is no hard boundary condition. But it also showed that BANet does not generalize to our attachment scenario that is common in surgery.

### D. Additional Active Sensing Results

**Motion Primitive Details.** We show additional details from the experiment discussed in the paper when comparing our method with the motion primitives in [2]. Due to ground contact, moving downward is forbidden. Therefore, we only

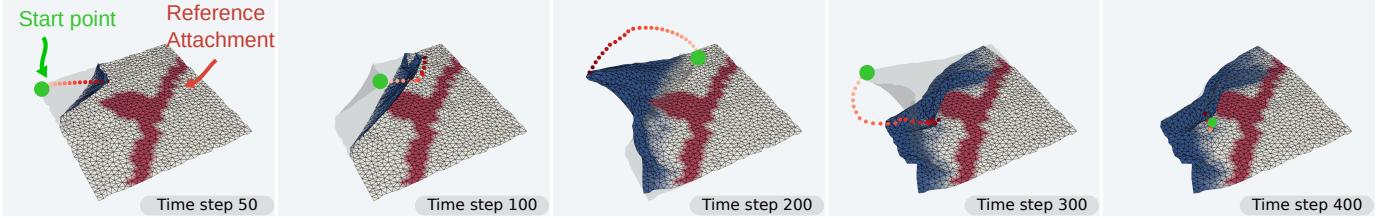


Fig. 13. Active sensing performance of our framework with doubled amount of mesh particles (1200) on test cases *Line-dot*. Our framework can still perform complex maneuvers such as switching directions and flipping for information gain even with more particles, showing that it can generalize to complex meshes.

consider 5 motion primitives of *up*, forward, *backward*, *left*, and *right*. Their direction w.r.t to the tissue is shown in the left of Fig. 12. To avoid sliding, all primitives are preceded by a grasp sequence that lifts up the tissue. Fig. 12 shows additional comparison between the proposed method and all choices of predefined motion primitives (PMP). All PMPs result in safety violation. Before violations, *up* and *forward* achieve more entropy reduction, but *up* achieves that faster in general. In comparison, our proposed method outperforms all PMPs, achieving more entropy reduction at the cost of smaller boundary energy.

#### E. Scalability

We evaluate our framework’s ability to handle higher dimensional meshes. This becomes important when simulating surgical scenes of higher complexity. Specifically, we double the number of particles on our mesh (1200 particles). We show the performance of our framework with our most computation-demanding task, active sensing, in Fig. 13. Our framework is able to maintain the similar amount of information gain with more particles. The computation time of our active sensing controller increases from 2.9s to 4.5s. Despite increase in computation time, we believe our algorithm can be accelerated via techniques such as asynchronous computation and multi-resolution meshes to handle more complex scenarios.