AWARD NUMBER: W81XWH-18-1-0769

TITLE: A Fundamental Theory for Dexterous Surgical Skills Transfer to Medical Robots

PRINCIPAL INVESTIGATOR: Juan P Wachs

RECIPIENT: Purdue University, 315 N. Grant Street, West Lafayette, 47907, IN

REPORT DATE: JANUARY 2022

TYPE OF REPORT: Final Report

PREPARED FOR: U.S. Army Medical Research and Development Command
Fort Detrick, Maryland 21702-5012

DISTRIBUTION STATEMENT: Approved for public release; distribution is unlimited.

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision unless so designated by other documentation.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JANUARY 2022	Final	15Sep2018 - 14Sep2021

4. TITLE AND SUBTITLE

A Fundamental Theory for Dexterous Surgical Skills Transfer to Medical Robots

5a. CONTRACT NUMBER

5b. GRANT NUMBER
W81XWH-18-1-0769

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Juan P Wachs

E-Mail:

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Purdue University

West Lafayette, 47907, IN

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Medical Research and Development Command
Fort Detrick, Maryland 21702-5012

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTE

14.ABSTRACT
Chronic diseases like cancers that evade the immune system continue to impact the lives of millions of patients worldwide causing high morbidity and mortality as well as significant economic burden. It is now evident that in melanoma tumors, persisting antigen results in the development of dysfunctional or exhausted anti-tumor T cells (Texs). Continuous TCR stimulation and multiple suppressive mechanisms limit effector T cell functions, including high expression and signaling of multiple inhibitory receptors (e.g., PD-1, CTLA- 4, Lag3, Tim3) including PSGL-1 (P-selectin glycoprotein ligand-1), which we identified. Therapeutic success targeting these inhibitory checkpoints (PD-1/PDL-1 and CTLA-4) using blocking antibodies can reinvigorate Texs and improve pathogen and tumor control in mice and patients. These drugs are now standard treatment for melanoma, but at present are only effective in a subset of patients, highlighting the need to identify additional inhibitory pathways that can be targeted to improve patient outcomes. Our purpose is to understand when PSGL-1 signaling is required to establish T cell exhaustion and in what cell types PSGL-1 contributes to inhibition of T cell function. We will also understand whether PSGL-1 can be targeted to reverse T cell exhaustion to promote tumor control.

15. SUBJECT TERMS

NONE LISTED

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			USAMRDC
U	U	U	UU	75	19b. TELEPHONE NUMBER *(include area code)*

TABLE OF CONTENTS

	Page No.
1. Introduction	**3**
2. Keywords	**3**
3. Accomplishments	**4**
4. Impact	**77**
5. Changes/Problems	**79**
6. Products	**80**
7. Participants & Other Collaborating Organizations	**82**
8. Special Reporting Requirements	**85**
9. Appendices	**87**

Purpose: We will develop a theoretical framework for supervised autonomy, capable to self-adjust its autonomous behavior in presence of delays and perform procedures in never seen settings using a transfer learning paradigm.

Scope: Future battlefield medical operations in the Multi-Domain Battlespace calls for robotic platforms that are near-autonomous and can provide patient support systems integrated with perception engines for accurate detection and treatment of casualties in closed-loop enroute care systems. Such an integrated system should allow safe and effective-telerobotic surgery. When tele-operation is challenged due to limited bandwidth, latency and lost-of-signal, autonomy needs to step in. While new robots have been endowed with color and depth visual, acoustic and force sensing capabilities, together with new tooltips both rigid and flexible, there is a gap in knowledge about two critical aspects: (1) how to leverage operator's expertise under limited connectivity (rather than switching on/off autonomy); (2) how to apply the existing abundant knowledge about surgical motions and maneuvers from the OR to new, Scalably Austere Environments (SAE).

Major Findings: To tackle the effects of connectivity associated with telerobotic surgeries, we propose the DESERTS framework. DESERTS provides a novel simulator interface where the surgeon can operate directly on a virtualized reality simulation and the activities are mirrored in a remote robot, almost simultaneously. Thus, the surgeon can perform the surgery uninterrupted, while high-level commands are extracted from his motions and are sent to a remote robotic agent. We assessed the framework's performance in the presence of increasing delays for two tasks: a peg transfer and a surgical debridement. Notably, the system maintained a task success rate of 87% and 88% respectively from no-delays to 5 seconds of delay. We also explored transfer learning between surgical tasks. Our results demonstrate that our framework can boost the performance of surgeme recognition across surgical tasks. Using a pre-trained network improves the classification accuracy achieves a classification accuracy of 76% with only 8 sequences in target domain, which is 22.5% better than a no-transfer scenario.

1. **INTRODUCTION:** Narrative that briefly (one paragraph) describes the subject, purpose and scope of the research.

We research and study new approaches to transmit information (encoding/decoding) to maximize information content while reducing redundancy and common knowledge, under constrained communication settings. Based on the decoded and reconstructed information, we determine at what extent the machine needs to autonomously complete surgical procedures. The robot, in turn, will apply a sequence of maneuvers associated with the required surgical procedure. We theorize new approaches for transfer learning, where learnt patterns will be

2. **KEYWORDS:** Provide a brief list of keywords (limit to 20 words).

Telerobotics, Transfer learning, telemedicine, deep neural networks, simulation, surgical training, co-presence, tele-operation, surgeme recognition.

3. **ACCOMPLISHMENTS:** The PI is reminded that the recipient organization is required to obtain prior written approval from the awarding agency Grants Officer whenever there are significant changes in the project or its direction.

What were the major goals of the project?
List the major goals of the project as stated in the approved SOW. If the application listed milestones/target dates for important activities or phases of the project, identify these dates and show actual completion dates or the percentage of completion.

T1.1: Data Collection and Sensory-motor Library Generation

- Peg transfer collection (15-Sept-2019 – 15 Dec-2018) 100%
- Debridement data collection (15-Nov-2019 – 30-Sep-2020) 100%
- Library Generation (15-Sept-2018 – 30-Sep 2020) 100%

T1.2: Semantic Rule Creation

- Creation of terminal symbols (surgemes) (15-Jul-2020 – 10-Sept-2020) 100%
- Development of grammar and rules (temporal models) (15-Nov-2019 – 30-June-2021) 100%

T1.3 Learning with Imperfect and Randomly Delayed Information

- Find Information theoretic bounds for optimal algorithms (15-Sept-2018 – 15-Mar-2019) 100%
- Modification of standard algorithms (1-Feb -2020 – 15 July-2020) 100%

T2.1 Integrating predictions with semi-supervision

- Find the prediction error and competitive guarantees (15-July-2020 – 15-October-2020) 100%
- Coupling multiple expert trajectories with attendant visual cues (15-July-2019 – 15-Jan-2020) 100%

T2.2 Visual recognition and adaptive learning of primitives

- Train a deep neural network for surgical step recognition using simulator data (15-July-2019 – 15-Oct-2019) 100%
- Train a deep neural network (DNN) with surgical task data (15-July-2019 – 15-Oct-2019) 100%
- Develop a framework for transfer learning for different surgical tasks (15-July-2019 – 15-June-2021) 100%

T3.1 Motor command generation in constrained and new settings

- Train a network for surgical step target prediction (15-Nov-2020 – 15-May-2021) 100%
- Use Reinforcement Learning to optimize a policy for surgeme execution (10 Jan 2019 – 15 Oct 201) 100%

T3.2 Experimental Design: Metrics to evaluate the technical specifications of the system.

- Experiment 1: Metrics Metrics to assess movement generation plan (target prediction) (15-Nov-2020 – 15-May-2021) 100%
- Experiment 2: Metrics to assess learning with Imperfect and Randomly Delayed Information (15-Mar-2020 – 15-Oct-2020) 100%
- Experiment 3: Metrics to Assess the Accuracy of High-Level Surgical Descriptors (15-Oct-2019 – 15-June-2021) 100%
- Experiment 4: Metrics to Assess Connectivity Dropout and Recovery with Multiple Expert (15-Oct-2019 – 10-Dec-2020) 100%

What was accomplished under these goals?

For this reporting period describe: 1) major activities; 2) specific objectives; 3) significant results or key outcomes, including major findings, developments, or conclusions (both positive and negative); and/or 4) other achievements. Include a discussion of stated goals not met. Description shall include pertinent data and graphs in sufficient detail to explain any significant results achieved. A succinct description of the methodology used shall be provided. As the project progresses to completion, the emphasis in reporting in this section should shift from reporting activities to reporting accomplishments.

T1.1 Data Collection and Sensory motor Library Generation

The objective of the data collection was to generate a library of surgical actions for peg transfer and suture. Any procedure can be broken down into atomic surgical actions, and these actions can be used in a sequence to complete a procedure. The data was collected from three robotic platforms (see Figure 1): a) The Taurus robot from SRI, b) Simulated Taurus Robotic environment, c) the YuMi collaborative platform, adapted for the surgical environment and d) and the Da vinci Research Kit (dVRK). We refer to this dataset as DESK: a Robotic Activity Dataset for Dexterous Surgical Skills [1]

1.1.1 Peg Transfer with Taurus

The Taurus robot has two 7 DOF grippers that are controlled through the Razer Hydra® controllers. The system has a foot pedal that allows to switch control between the arms and the camera. In this setup, the robot only moves when the foot pedal is pressed. The setup for recording consisted of 48 trials and a total of 10 subjects. Each trial had 3 peg transfers for a total of 144 peg transfers. The initial position of the pegs was randomized. Half of the trials were initialized with the pegs on the left, creating left-to-right transfers and half of the trials were initialized with the pegs on the right. The place of initialization was also randomized. Figure 1 depicts the peg transfer setup.

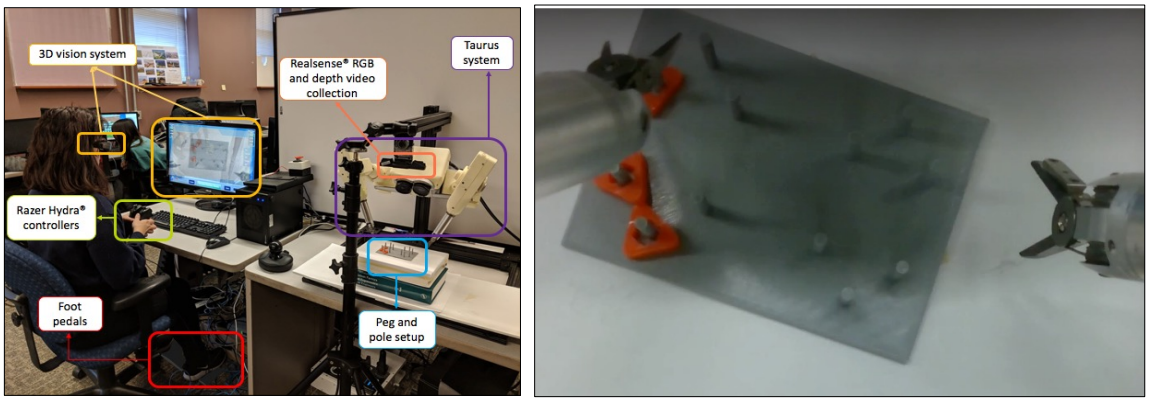
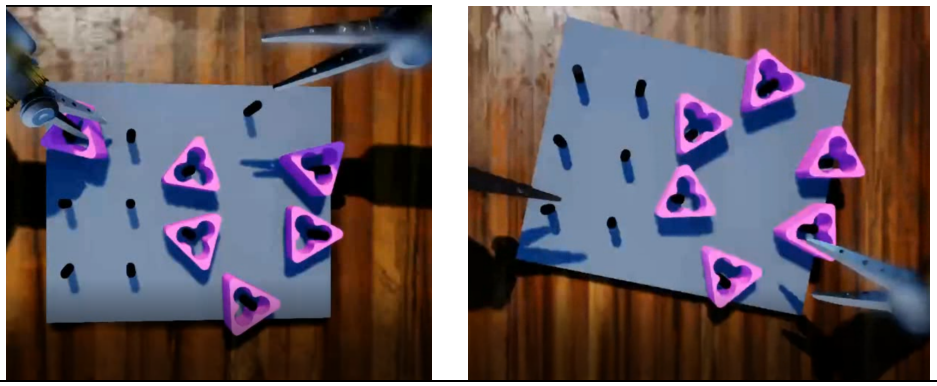


Figure 1. Taurus Data collection Setup

1.1.1 Simulated Peg Transfer with Taurus

A total of 100 peg transfers were recorded in total with average of approximately 17 peg transfers per subject. The recording setup was the same as in the previous sessions. Each subject recorded 3 trials with each trial consisting of 6 transfers. To add variability to data, the start and goal pegs for each transfer were randomized. Also, the direction of transfer was randomized for each transfer, i.e. pick with left arm, transfer to right arm, and place or pick with right arm, transfer to left arm, and place. The orientation of the peg board was fixed through all trials. This new data together with the data collected with free peg board contains a total of 233 peg transfers. Figure 2 shows the difference between fixed and free peg board.



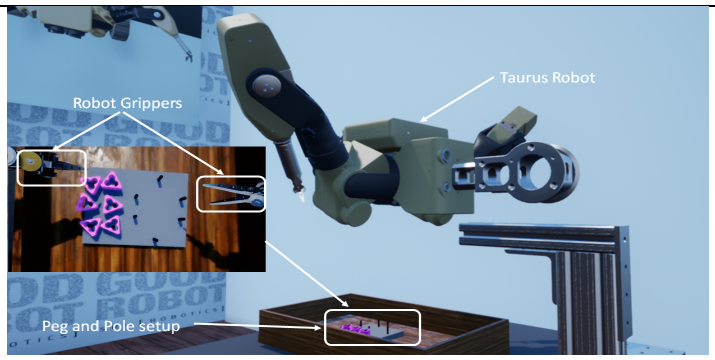


Figure 2. Top Left: Fixed peg board orientation. Top Right: Free peg board, the orientation changes during task performance. Down: Taurus entire Simulated Environment

1.1.2 Peg transfer with Yumi

The adaptation of the YuMi robot (7 DOF) for surgical tasks consists of 3D printed grippers and teleoperation interface using the HTC VIVE™ motion controllers. The controller´s motion is mapped to the robot's end effector while the trigger button is pressed, allowing to reset the workspace of each arm independently. Opening and closing of the gripper is controlled with the Trackpad button. During teleoperation, the translation of the robot's end effector corresponds to the translation of the controllers in the real world, while the changes in orientation for each gripper are based on the controller's coordinate frame. During the data collection, the teleoperator stands in front of the YuMi robot and gets visual feedback through direct observation of the robot's workspace (See Figure 3).

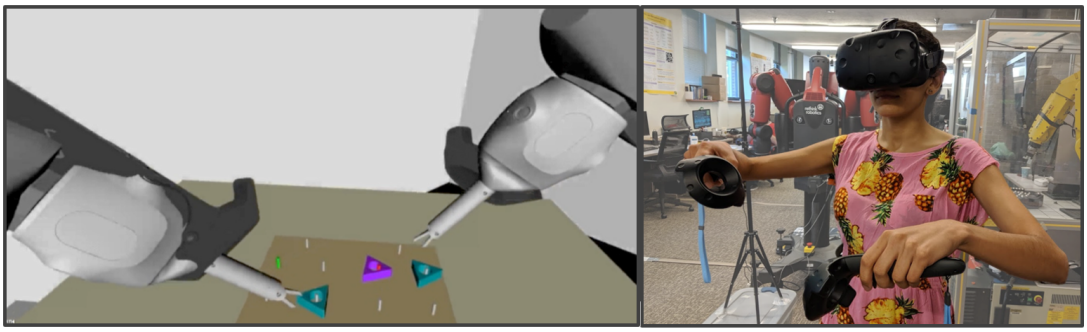


Figure 3. Data collection environment for the Yumi robot

Six subjects collected a total of 72 trials consisting of 3 peg transfers for a total of 216 instances of peg transfer were recorded, following the randomization described in the previous section. The peg board setup for the YuMi robot was twice the size of the peg board used in the Taurus Data Collection.

1.1.3 Simulated peg transfer with Yumi

Yumi simulation dataset were recorded from six subjects. Each subject was trained for 5+ hours to achieve a good level of dexterity for the task. The training sessions consisted on moving the arms through straight paths, followed by pick and place operations and finally performing peg transfer for 5 sessions. The data was recorded after the training, with a total of 213 peg transfers. The recording setup was the same as the real Yumi setup in DESK dataset [1]. Each subject recorded 12 trials with 3 peg transfers per trial. Alternating with objects initialized on the left or the right. At the beginning of each session the initial position of the objects was randomized across the 6 pegs on the chosen side of the board. Figure 4 shows the interface for the YuMi simulator.

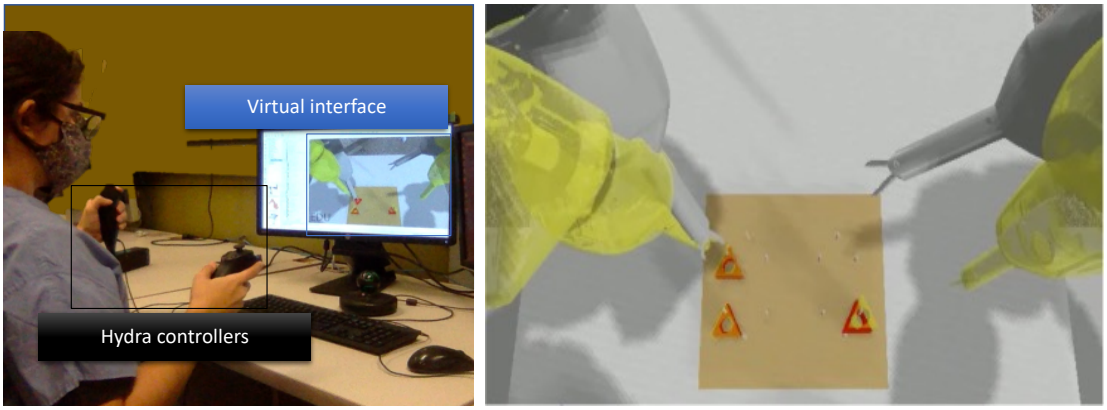


Figure 4. Data collection interface for the simulated YuMi

1.1.4 Peg transfer with DvRK

To exploit the flexibility and accuracy of the dVRK system, a professional laparoscopic pegboard was purchased. Thus, the task difficulty was increased to match the precision level offered by the da Vinci robot. Figure 5-Left shows the peg board used for the da Vinci setup.

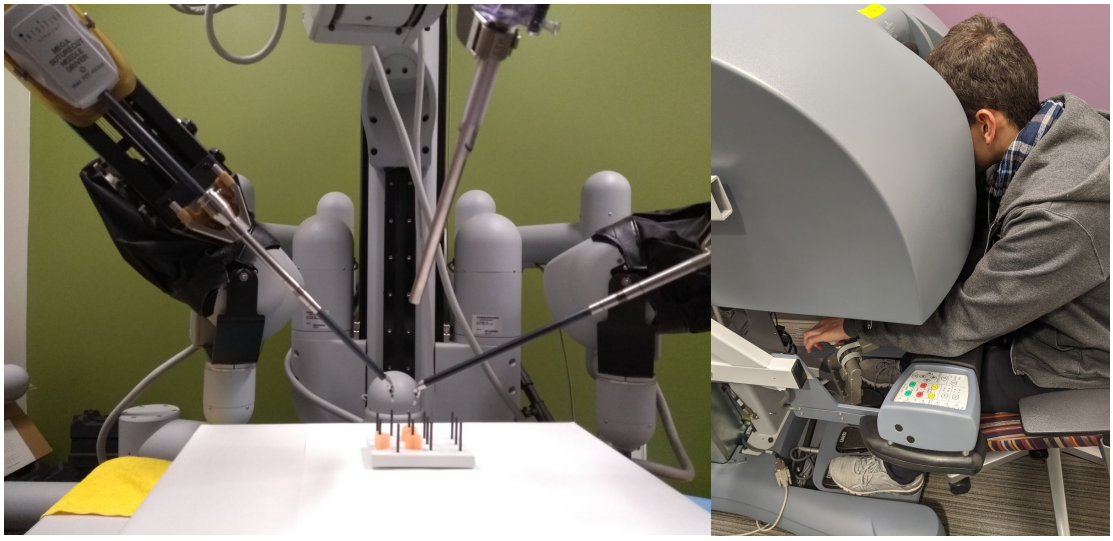


Figure 5. (Left) dVRK peg transfer setup. (Right) dVRK teleoperation setup for data collection.

A total of 144 peg transfers were collected from eight subjects. Each subject performed six trials. The subjects had a 15 minute practice session to familiarize themselves with the da Vinci console and controls (See Figure 5-Right) before the data recording. Every trial required three transfers from one side of the peg board to the opposite, starting from a random position. Both RGB video and kinematic information were collected during this phase.

1.1.5 Debridement with simulated Taurus:

To model the elasticity of skin in the debridement task, we use springs within the simulated skin. In the simulation we have 3 patches of dead skin which must be removed to successfully complete the surgical task. Further, after removing the necrosed tissue, it must be deposited over a surgical tray. We use a

Taurus robot as our execution robot inside the simulator. We used the Taurus robot with tool tips in the form dissection scissors to cut through the dead skin, as shown in Figure 6.

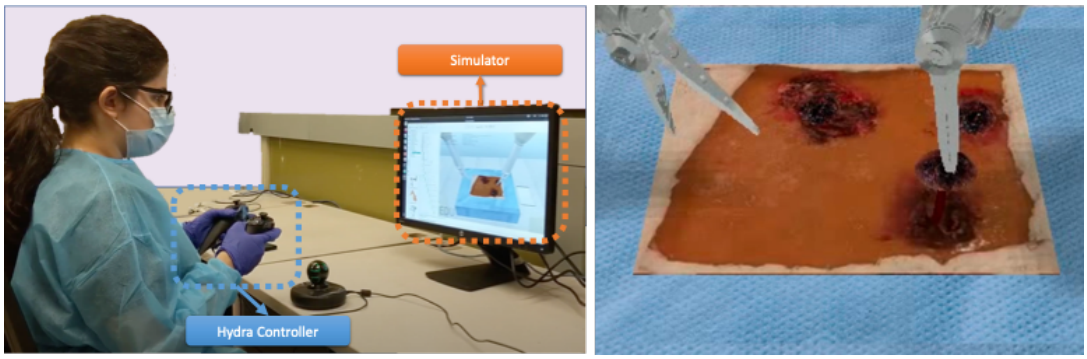


Figure 6. Debridement data collection Setup

The robot is controlled using Razer Hydra sensing controller which allows a natural movement for operator's hand throughout the task, as shown in figure 3. The collected kinematics includes the robot end-effector position in Cartesian coordinates from the robot's frame of reference and orientation angles for both, the left and the right arms. Additionally, it also includes the state of the gripper to include whether the gripper is in open or in closed position. This schema results a kinematic sequence where the kinematic vectors consist of 3+4+1 elements for each arm resulting in 16 elements for both arms. We collected and annotated data from 3 subjects.

Each subject completed 12 trials of a debridement task and each debridement task requires removing all 3 dead tissue patches from the skin. Each subject conducted 6 trials. In each trial, it is required to grasp the necrosed tissue from the left robot arm and cut the skin from the right robot arm. Similarly, they also completed 6 trials where the roles of the arms were inverted to allow the model to learn ambidexterity

T1.2 Development of models for surgeme recognition

We developed a methodology that uses compact image representations with kinematic features for surgeme recognition in the DESK dataset (Taurus II, Taurus II simulator, and YuMi robots data). This dataset offers samples for surgical procedures over different robotic platforms with high variability in the setup. We performed surgeme classification in two setups: 1) No transfer, 2) Transfer from a simulated scenario to two
real deployable robots. Then, the results were compared with recognition accuracies using only kinematic data with the same experimental setup. The results show that our approach improves the recognition performance over kinematic data across different domains.

Our approach combines features extracted from the video image and kinematic data to perform transfer learning in surgeme recognition. The approach is based on robot agnostic feature extraction that can be applied to different environments in a transfer learning scenario.

An overview of our system architecture is given in Figure 7. We used a Convolutional Neural Network (CNN) to extract meaningful features from video images. The images extracted from the videos were resized from 1920×1080 to a 228×128. We employed the CNN model ResNet18 [3] that is pre-trained on ImageNet and removed the last two layers (i.e., fully connected and softmax) to create a high dimensional feature representation for each video frame. The extracted features have dimensions of 1×512×4×8, which were flattened to a single vector of size N = 16384. Due to the large dimension in the output of the ResNet18 model a dimensional reduction module was required

to reduce each image/frame features to a lower-dimensional vector (M = 30, where M << N). Principal Component Analysis (PCA) was used to reduce the dimension of each frame image to a lower 30-dimensional vector.

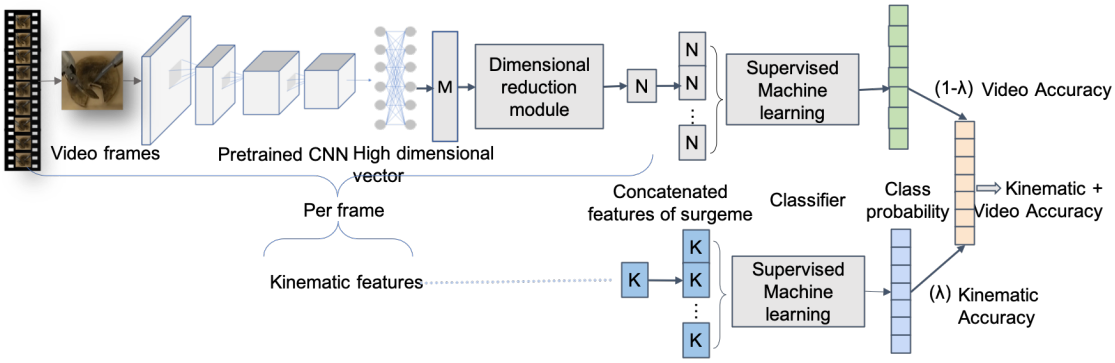


Fig. 7: Architecture overview of the approach, M: Number of dimensional features extracted using pre-trained CNN, K: number of kinematic features, N: number of image features after Dimensional reduction module.

In parallel, we extracted features from kinematic data by reducing the kinematic features from multiple domains to the commonly shared features: position, orientation, and gripper status of the end-effector (14 features: seven features in each arm). Since each surgeme instance consists of a variable number of frames, we re-sampled (via linear interpolation) the original instances to a fixed number of frames (40) to generate a sequential feature instance for each surgeme. In particular, for the kinematics, we concatenated the 14 features corresponding to each frame, to create a single 560-dimensional vector per surgeme (40 × 7 × 2). On the other hand, for the video data, we concatenated the 30 features obtained after the PCA reduction into a single feature vector of 1200 dimensions (40×30) that represents each surgeme. Another PCA reduction was applied on the 1200-dimensional feature vector to obtain a 100-dimensional feature vector for the entire image sequence of 40 frames.

In our pipeline, visual features and kinematic features came from different distributions. Thus, to take advantage of both visual and kinematic modalities, we trained the supervised learning algorithms separately and combined their output class probabilities as shown in the following equation:

$$P(C) = \lambda P_{kin}(C) + (1 - \lambda) P_{video}(C),$$

where C is the class and $\{P_{kin}(C), P_{video}(C)\}$ are the probabilities given by supervised learning models using kinematic and video features respectively. Here, λ is the hyperparameter (from 0 to 1), which allows us to modify the contribution of the models based on kinematics or video frames. When λ is set to 0, the classification is based on video data alone. If λ = 1, the classification uses only kinematic data. Note that our architecture allows us to plug in separate supervised machine learning model for kinematics and video. Thus, the models for video and kinematics can be designed separately. For simplicity, in our experiments, we use the same supervised learning model for both video and kinematic features.

The image features obtained from the pre-trained ResNet18 and the extracted kinematic features are agnostic to the robot system and the surgical setup. This means we get the same feature set, even when the actions or the robot varies. Having a common set of features facilitates surgeme

classification across different domains, since it helps models to leverage on the information coming from another domain.

1.2.1 Experimental Setup:

The proposed architecture was tested over two scenarios: no-transfer scenario (train and test data are obtained from the same domain), and domain-transfer scenario (train on one domain and test on the other). The complete domain transfer scenario used the Taurus simulator (S) data or a combination of simulation and real robot data (S+R) during training, while the algorithm was tested entirely on the real robot data (R). A value α represents the percentage of real data (i.e., target domain) that was added to the simulator data (source domain) for the model training. The rest of the real data (in the target domain) was used as the testing set. Thus, $\alpha = 0$ indicates a complete transfer. Two supervised learning methods were tested: (1) Random Forest (RF), and (2) Support Vector Machines (SVM), using the scikit-learn [4] implementation. A five-fold cross-validation approach was used with a data split of 80-20% for training and testing respectively. Furthermore, hyperparameter setting, we used a *kernel="poly"* for SVM classifier. For RF, we set n estimators = 200 (number of trees in the forest), and maximum depth = 10. We set the combined model's hyper parameter λ to 0.5 for RF and to 0.8 for SVM (empirical best).

1.2.2 Results:

Table 1 shows the surgeme recognition results using our approach for kinematics, video information and a combination of both modalities for the non-transfer scenario. Kinematic (Kin) features alone perform substantially better than features based on video frames (Visual). RF performs better than SVM for visual data achieving up to 63% accuracy on the Taurus real robot. Moreover, in most of the cases, accuracies improved when both kinematic data and image features are used (Kin+Visual). In general, adding visual data improved the classification accuracy of surgemes compared to the use of kinematic or visual data independently.

Two transfer learning scenarios were evaluated: (i) Taurus Simulator to Taurus real robot (Sim2Taurus) and (ii) Taurus Simulator to YuMi (Sim2YuMi). The accuracy results for the domain-transfer scenario (Sim2Taurus): simulator + real-robot (S+R) are presented in Figure 8. The results show that kinematic+visual features surpass the accuracy and consistently outperforms results that use only kinematic features.

Table 1. Classification accuracy on the no-transfer scenario using RF and SVM on kinematic only (Kin), visual only (Visual), and both (Kin+Visual) features. In most cases adding visual features with kinematic features improve the accuracy.

	RF			**SVM**		
Robot	**Kin**	**Visual**	**Kin+Visual**	**Kin**	**Visual**	**Kin+Visual**
Simulator (Taurus)	92.69	56.46	93.03	91.66	33.16	92.51
Taurus	94.68	63.77	96.49	91.43	35.19	92.73
YuMi	96.87	60.65	97.23	92.90	21.91	93.38

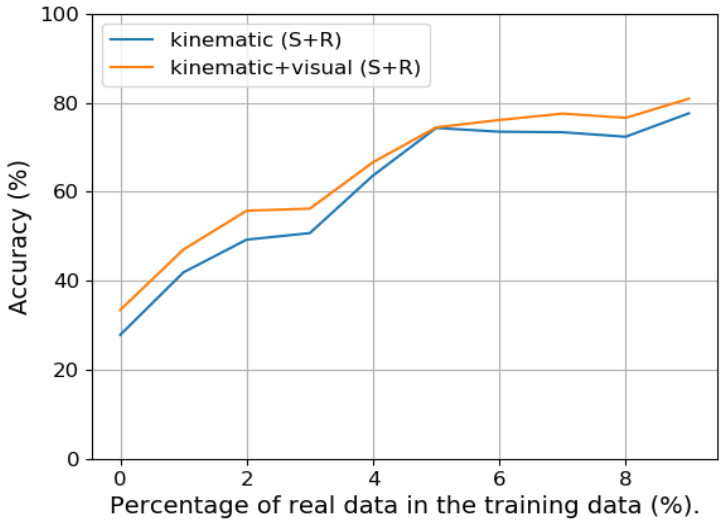


Fig. 8: Performance comparison for transfer learning from the Taurus Simulator to the Taurus Real Robot (Random Forest). Overall transfer accuracy improved quickly and it reaches over 93% with only 50% real data in the training.

Table 2. Domain transfer accuracy (Random Forest) when the models are trained on the Taurus simulator robot and tested on real robots (Taurus (Sim2Taurus) and YuMi (Sim2Yumi)). α is the percentage of real-robot data used in the training. Features: kinematic only (Kin), visual only (Visual), and both (Kin+Visual).

	Sim2Taurus			**Sim2Yumi**		
α(%)	**Kin**	**Kin+Visual**	**Gain(%)**	**Kin**	**Kin+Visual**	**Gain(%)**
0	27.92	33.51	20.00	14.20	14.80	4.24
1	41.94	47.05	12.19	25.64	33.78	31.75
2	49.27	55.22	13.17	37.42	43.07	15.08
3	50.74	56.22	10.82	39.03	46.84	20.00
4	63.65	66.62	4.67	50.38	54.14	7.46
10	78.93	82.54	4.57	74.73	73.40	-1.78
50	93.77	94.03	0.28	88.70	90.14	1.62
90	93.51	96.10	2.77	95.24	96.43	1.25

Figure 9 shows that the performance of both the (Visual) and (Kin+Visual) data increases as we add more real-robot data (YuMi) in the training (S+R). The improvement obtained for the transfer

learning scenario using (Kin+Visual) data is higher than knowledge transfer based on kinematic data alone. Hence, when training data is limited on the target domain, the addition of visual data from simulation could help boost initial classification accuracy.

Table 2 shows transfer accuracy for kinematic only features (Kin), and both kinematic and visual features (Kin+Visual). It also shows the percentage gain which is calculated as
where A1 and A2 are the accuracies using Kin features, and Kin+Visual features, respectively.

$$Gain = \frac{A_2 - A_1}{A_1} \times 100,$$

It can be observed that adding visual features boosts the transfer accuracy in both Sim2Taurus and Sim2YuMi robot settings achieving up to 20% and 31% gain respectively. Another observation is that the performance improvement is generally higher for smaller α values. Intuitively, when the target domain data is limited during training, visual information can play an important role to boost up initial transfer accuracy. As more target domain data is added to the training, both Kin and Kin+Visual accuracy increase quickly. In particular, when only 50% data is added to the simulator training, the accuracy reaches over 93% and 88% for Sim2Taurus and Sim2YuMi respectively.

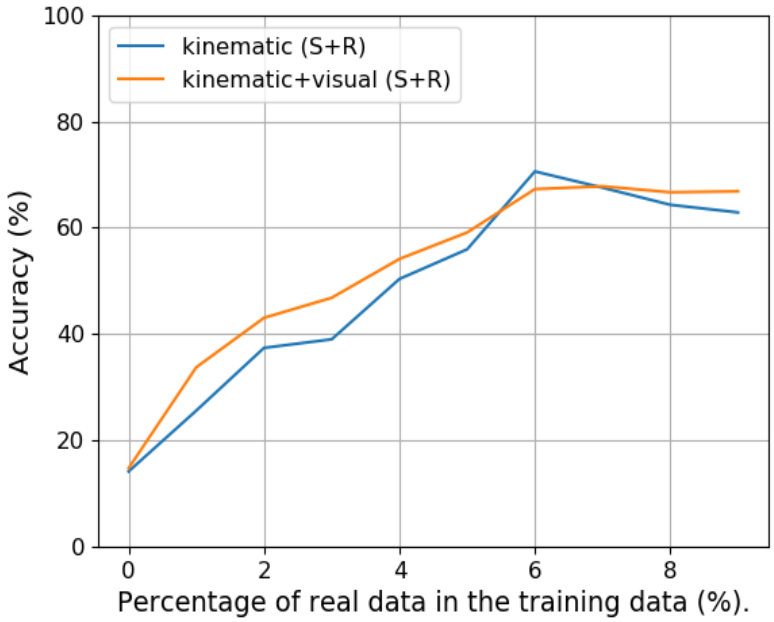


Figure. 9: Performance comparison for transfer learning from the Taurus Simulator to the YuMi Real Robot (Random Forest). Overall transfer accuracy improved quickly and it reaches over 88% with only 50% real data in the training.

T1.3 Learning with Imperfect and Randomly Delayed Information

We investigated the performance of decision-making agents for stochastic control. Based on the optimization goal for undelayed processes, we developed a policy to decide actions from imperfect and/or delayed state information. Results for them are presented in the following subsections.

1.3.1: Policy for imperfect and delayed observations

An agent at each time t is to observe the state $s_t \in S$, and take an action $a_t \in A$to obtain a reward r_t. This action results in a transition to next state $s_{t+1}$with probability $p(s_{t+1}|s_t, a_t)$. The decision rule or policy $\pi: S \rightarrow A$is the mechanism through which the agent selects the action after observing the current state. The task for the agent is to find a policy $\pi: S \rightarrow A$ which maximizes the optimization problem

$$\max_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} \gamma^t r_t(s_t, \pi(s_t))|s_1 = s\right]$$

Where $\gamma(< 1)$is the weight given to future rewards to keep the summation bounded for large values of T. The optimal value of the sum is denoted as V^*. When $T$is large or $T \rightarrow \infty$
a function $Q: S \times A \rightarrow R$ can be defined as,

$$Q^*(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)}\left[V^*(s')\right]$$

The optimal policy which solves the optimization problem is given as

$$\pi^*(s) = \arg\max_{a \in A} Q^*(s, a)$$

However, in our setup, the state information s_t might not be observed by the agents, instead, it receives an observation $o_t$at time t. For imperfect state observation o, the proposed optimal policy is given as

$$\pi^*(o) = \arg\max_{a \in A} \mathbb{E}_{s|o}\left[Q^*(s, a)\right]$$

1.3.2 Lower bounds on the policy obtained

We also provide a lower bound on the performance of this policy. We compare the performance with the instance when there exists an oracle which can provide the optimal action for the current state from the imperfect or delayed observation. With the policy $\pi(o)$, the total expected reward achieved is lower bounded as

$$V^{\pi} \geq V^* - \frac{R_{max}}{(1-\gamma)^2}\left(1 - \frac{1}{|A|}\right)$$

where R_{max} is the maximum possible reward the agent can obtain.

1.3.3 Algorithm implementation details

To obtain the policy we utilize a model-based implementation of reinforcement learning to learn probability distributions of the next state s_{t+1} given the current state s_t and action a_t.

We implemented an algorithm (named EMQL) to simultaneously learn the policy using reinforcement learning and state distribution for discrete state spaces and using deep neural networks. The algorithm trains a deep Q network to calculate the Q function for each state and action. The algorithm, in parallel, also trains a generative neural network to generate probable current states from the imperfect/delayed observations available. Since generative neural networks do not give the probability, we generate the probable states from the last known state and action sequence. We use Monte Carlo sampling is used to generate a large number (~5000) probable current states. Later, from the Q function network trained in parallel, we calculate the Q-values for each action for all the probable current states generated from the generative network. Lastly, the action which maximizes the empirical mean of Q function is selected.

1.3.4 Performance evaluations using existing methods for discrete state spaces.

We evaluated the algorithm on the OpenAI Gym platform for preliminary testing. We use stochastic delays for testing purposes. Each observation is delayed by discrete time steps. The delays are generated from Geometric distribution. We denote the probability of success p as the probability of delaying the observation at any instant. Thus, the average delay is $\frac{1}{1-p}$. Further, the delays cause the state observations to become asynchronous. The algorithm always considers the latest state observation to make a decision. The stale values are used to train the generative and deep Q networks.

Further, the delays cause the state observations to become asynchronous. The algorithm always considers the latest state observation to make a decision. The stale values are used to train the generative and deep Q networks.

Figure 10. shows the performance of EMQL under expected delays of 2 units, and 4 units. The observations are averaged over 20 different runs of the algorithm. The performance of the algorithm is limited by the performance of the generative neural network. We conjecture that by improving the generative neural network, the performance of EMQL algorithm also improves, and reduces the gap from no delays as well.

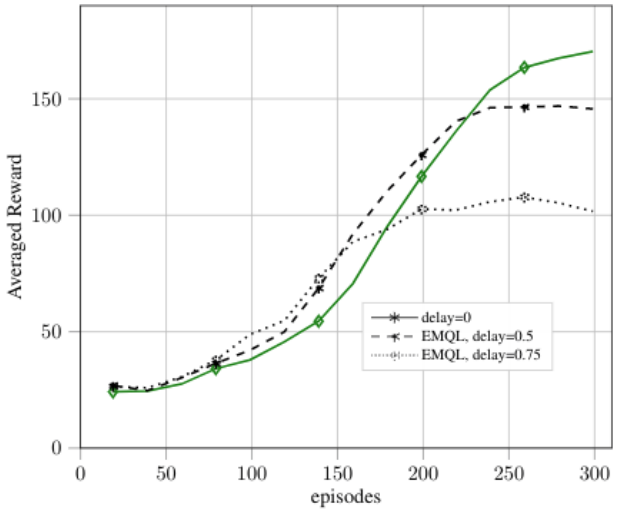


Figure 10. EMQL performance with different delay probabilities

T2.1 Integrating predictions with semi-supervision

2.1.1 Target Prediction

We use a Gaussian process to model the movement of a simulated robot controlled by an operator. The goal of this model is to predict the future pose of the user's trajectory so the robot can begin executing the maneuver For the operator trajectories, we rely on our DESK [1] which consists of a peg-transfer tasks performed using multiple robots by various subjects. The goal of the model is to leverage the available trajectory history to predict the future position of the robot tool-tip. This is achieved by minimizing the Euclidean distance between the position of the tool-tip produced by the teleoperator at the i^{th} frame and the predicted position for that frame.

To predict the pose at future f, using a Gaussian process, we use an h-length pose history. The idea behind providing the history till the h[th] frame is to allow the model to capture the current position and the higher-order derivatives including the velocity, acceleration, etc. It may be desirable to use all the available trajectory history for position prediction, however, this is not computationally tractable. To understand the impact that the length of the history has on accuracy, we performed an experiment to determine how much history is sufficient to predict future positions within an acceptable Euclidean error. Additionally, we compare the proposed model against Long Short-Term Memory [7], which is a deep learning technique to model sequential data.

2.1.2 Formulation

Let $x_n \in \mathbb{R}^3$ be a vector that represents the coordinates of the robot tool tip at time step n. n denotes the discrete time step sampled at a frequency of 30 Hz. Let $\{x_n\}$ be kinematic sequence with discrete indexing n. At every step n, we want to predict x_{n+k}, which is the position of the robot tool tip after k steps. To achieve this prediction, we use the available history h of the robot tool tip $\{x_m\}_{m=1}^{n}$, to predict the future position of the robot, defined as $\{x_m\}_{m=n-h+1}^{n}$. The goal of the model is to minimize the expected mean squared error or the average squared Euclidean distance as:

$$\mathbb{E}\left[\left\|\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n\right\|_2^2\right]$$

where $\hat{x}_n$ is the prediction of robot tool tip position at time step n.

3.2. Gaussian Process

We let $x_{n+k} = f_n(x_1, \ldots, x_n)$, where $f_n(\,.\,)$ is some function. However, since learning all f_n will require a large amount of data, we limit the history till last h steps. Let $\hat{x}_n = \left[x_{n-h+1}^T, \ldots, x_n^T\right]^T$ be the vectorization of $\{x\}_{m=n-h+1}^{n}$. Then we aim to learn the function $x_{n+k} = f(\hat{x}_n)$. We model the function f using Gaussian process.

If we have a training dataset $D = \{\tilde{x}_m, f(\tilde{x}_m)\}_{m=1}^{N}$, then using the assumption that we are fitting a Gaussian process, we can write,

$$\begin{bmatrix} f(\tilde{\boldsymbol{x}}_1) \\ \vdots \\ f(\tilde{\boldsymbol{x}}_M) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\tilde{\boldsymbol{x}}_1) \\ \vdots \\ m(\tilde{\boldsymbol{x}}_M) \end{bmatrix}, \begin{bmatrix} K(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_1) & \cdots & K(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_M) \\ \vdots & \ddots & \vdots \\ K(\tilde{\boldsymbol{x}}_M, \tilde{\boldsymbol{x}}_1) & \cdots & K(\tilde{\boldsymbol{x}}_M, \tilde{\boldsymbol{x}}_M) \end{bmatrix}\right) \quad (1)$$

where have the mean $m(\tilde{x}_m)$:

$$m(\tilde{\boldsymbol{x}}_m) = \mathbb{E}\left[f(\tilde{\boldsymbol{x}}_m)\right]$$

and covariance kernel $K(\tilde{x}_m, \tilde{x}'_m)$:

$$K(\tilde{\boldsymbol{x}}_m, \tilde{\boldsymbol{x}}'_m) = \mathbb{E}\left[(f(\tilde{\boldsymbol{x}}_m) - m(\tilde{\boldsymbol{x}}_m))(f(\tilde{\boldsymbol{x}}'_m) - m(\tilde{\boldsymbol{x}}'_m))^T\right]$$

For simplicity of the notation, let $K(X_t, X_t)$ denote the covariance matrix in Equation (1). For a novel point, $\tilde{x}$, $K(X_t, \tilde{x}_n)$ will denote the stacked covariance matrix $K[(X_t, \tilde{x}_n) \ldots, K(\tilde{x}_M, \tilde{x})]$. Also, we let $f(X_t) = [f(\tilde{x}_1)^T, \ldots, f(\tilde{x}_M)^T]^T$ and $m(X_t) = [m(\tilde{x}_1)^T, \ldots, (\tilde{x}_M)^T]^T$.

This Gaussian process allows us to generate a set of plausible values of the function at a novel point $\tilde{x}$ and the probability of each of the point using the mean and covariance kernel. Now on obtaining a new sample $\tilde{x}$, again using the property of the Gaussian process, we can write

$$\begin{bmatrix} f(\tilde{\boldsymbol{x}}) \\ f(X_t) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\tilde{\boldsymbol{x}}_n) \\ m(X_t) \end{bmatrix}, \begin{bmatrix} K(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}) & K(X_t, \tilde{\boldsymbol{x}}) \\ K(X_t, \tilde{\boldsymbol{x}})^T & K(X_t, \tilde{\boldsymbol{x}}) \end{bmatrix} \right)$$

Also, we can write the conditional probability distribution of the target value as

$$f(\tilde{\boldsymbol{x}})|\tilde{\boldsymbol{x}}, X_t \sim \mathcal{N}\Big(m(\tilde{\boldsymbol{x}}) + K(X_t, \tilde{\boldsymbol{x}})K(X_t, X_t)^{-1} f(X_t),$$
$$K(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}) - K(X_t, \tilde{\boldsymbol{x}})K(X_t, X_t)^{-1} K(X_t, \tilde{\boldsymbol{x}})^T\Big)$$

The next goal is to determine the covariance kernel which appropriately correlates the past poses to the future pose. The major covariance kernels can be classified as [8]:
1. Squared Exponential (SE): The squared exponential covariance function is a stationary covariance kernel (which only depends on the Euclidean distance between two inputs).
2. Rational Quadratic Kernel: The rational quadratic kernel is also a stationary kernel. It is effectively a sum of multiple SE kernels.
3. Dot Product Kernel: A dot product kernel is a nonstationary kernel which allows to model translations of input data.

Additionally, one can take a linear combination of the three kernels specified to obtain effective kernel as:

$$K(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2) = w_1 K_{SE}(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2) + w_2 K_{RQ}(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2)$$
$$+ w_3 K_{DP}(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2)$$

This linear combination of the kernels essentially acts as an OR operator and the resulting covariance is high if any of the kernel gives high covariance [8] This allows to model multiple patterns in a single kernel.

2.1.4 DElay-tolerant SEmi-autonomous Robot Teleoperation for Surgery (DESERTS) Famework

Our framework uses a simulator-based interface instead of live-streamed video to reduce the amount of information exchange. While the user operates the simulated robot, they can also see the pose of the actual robot and other objects related to the task, presented as an alpha-blended view in the simulator. At the remote site, an object detector identifies the objects of interest in the robot's field of view and sends them to the simulator for reconstruction. The simulator works on atomic units of surgical maneuvers referred to as surgemes. We tested our framework using the two tasks from the DESK dataset: 1) A peg transfer and 2) Surgical debridement.

Using surgemes we obtain a high-level abstraction of the surgical procedure, hence reducing the size of the content to be exchanged. Our framework consists of four primary components: vision, recognition, communication, and execution modules (shown in Figure 11), which are described in detail, as follows.

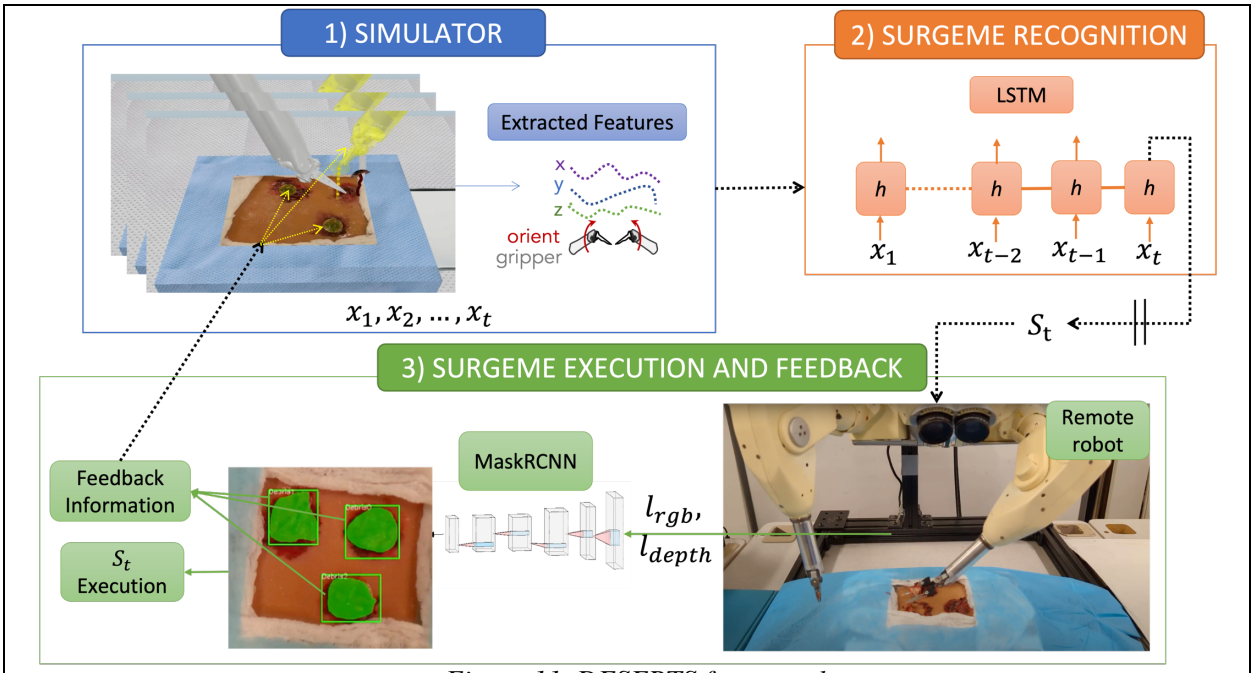


Figure 11. DESERTS framework

2.1.4.1 Vision Module

The vision module first identifies the objects and estimates their position in the real world (using semantic segmentation). The simulator then uses this information to update the scene. Figure 12 shows the simulator interface. We first describe the object recognition unit running at the robot. Then we describe the simulator's display unit, which reproduces the remote robot scene in the simulator. Object Recognition Unit. A 3D camera (Intel Realsense) is assembled on top of a remote ABB YuMi teleoperated robot. The camera streams color (RGB) and Depth image frames. This image stream is used to understand the environment state. This module extracts the 3D object poses and robot tool-tip poses using two neural network-based architectures, Darknet (YoloV3), and Mask-RCNN. The YoloV3 (Darknet) network detects the 2D object bounding boxes in the RGB image frame. Since the YoloV3 network can only identify objects as regions of interest, an object tracker is added to track objects of the same class (for example, three triangular objects). Figure 13 Shows the tracking and segmentation of the peg transfer objects and Figure 14 shows the result of the same vision processing on the debridement skin.

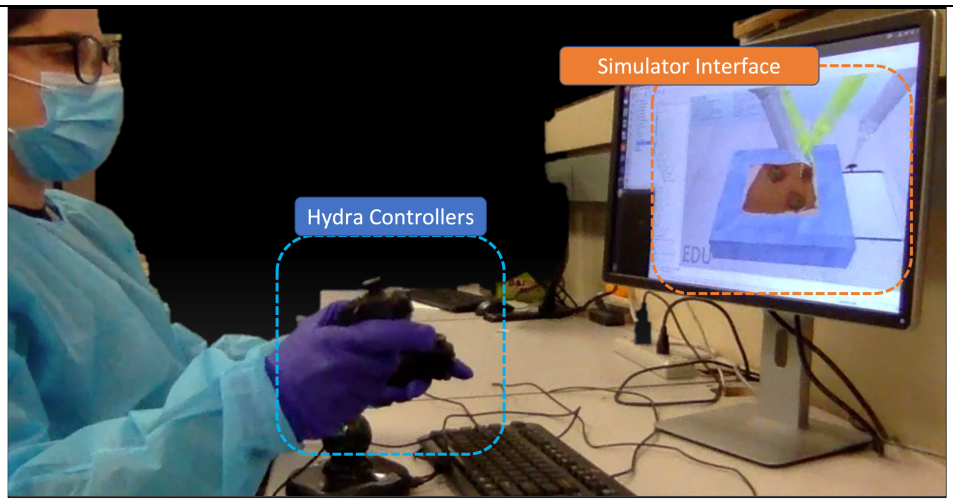


Figure 12. Simulator interface for debridement

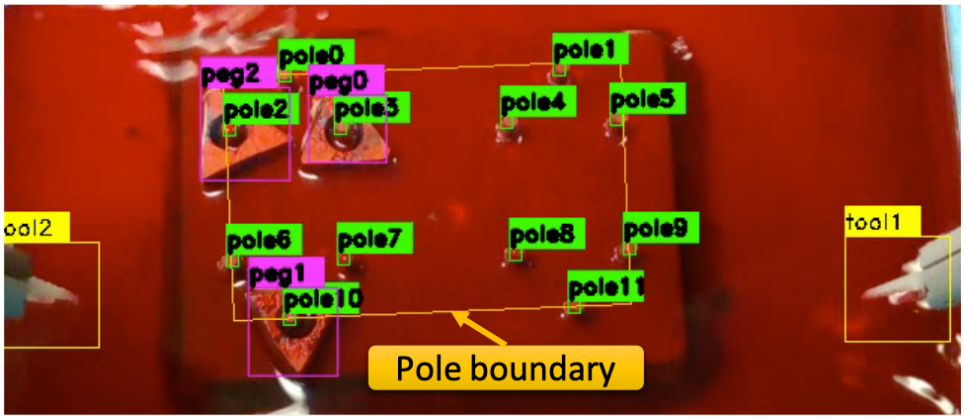


Figure 13. Peg transfer object tracking

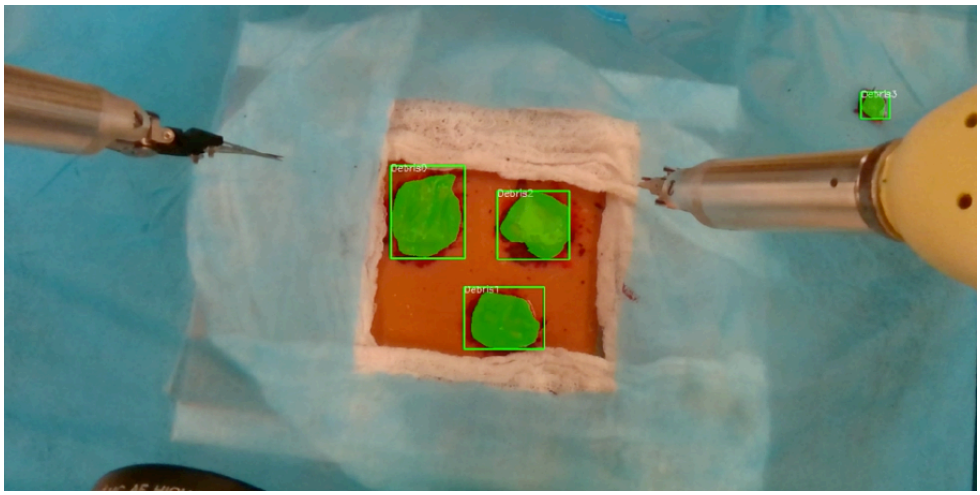


Figure 14. Debridement skin tracking

Vision challenges in turbulence: The rapid changes in the vision increased the ratio of false positives from 0.01 to 0.5. The false positives produced jumps in the object tracker, making it difficult to

estimate the real position of the object in the scene. The false positives were filtered out using the attributes listed below. For reference, Figure 15 shows the implementation of the turbulence setup.

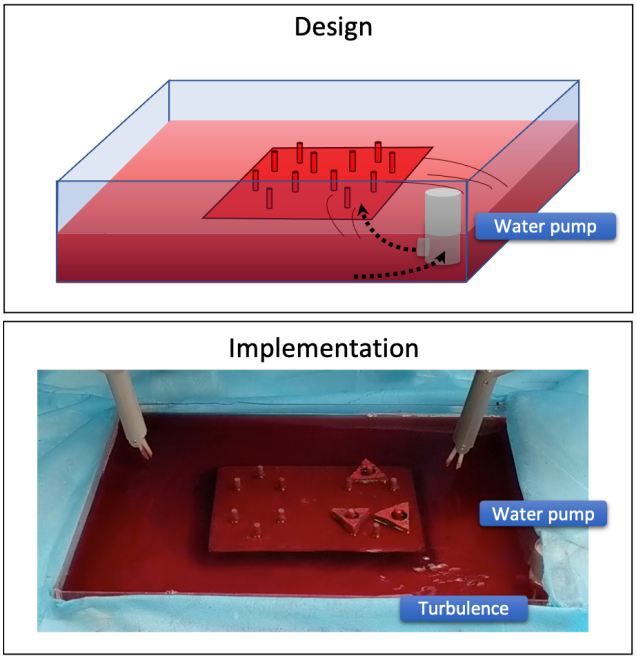


Figure 15. Turbulence design using a pump at water surface level.

Area filtering: We included knowledge about the peg board to further filter false positives with 'pole' labels. We defined the pole boundary that served as a ROI (Region of Interest) for the pole detection. The corners of the boundary were defined as$[max(x_i], \max(y_i)), [max(x_i), min(y_i)]$ and $[\max(x_i), \min(yi)]$, where x_i and y_i are the coordinates of pole i, with i=0,..,11 for a total of 12 poles. Figure 13 depicts the generated boundary.

Shape filtering: Since the dimensions of the detected objects have similar width and heights, only detections with an α proportion ($0.5 < \alpha < 1.5$.) were considered.

We extend the tracking algorithm, SORT, which can additionally address noise, occlusion, and failed detection. The tracker uses a filter to smooth the objects' motion per frame. It uses the object's velocity and a Kalman filter to estimate objects' position. The estimates are used to keep track of objects' position. The tracking system accumulates the object detection information of the last ten frames and estimates the confidence of the objects' presence, which reduces false positives. Display Unit. This unit allows the operator to see what is occurring at the remote site by reconstructing the scene from the objects recognized by the semantics identification unit. The simulator shows the remote robot environment in an alpha-blended layout (objects, remote robot), as shown in Figure 16. The simulator updates the alpha-blended objects at regular time steps. As the simulated robot may run independently, the position of the simulated robot and the alpha-blended real robot may diverge. To control this discrepancy, the operator can reset the simulated robot to synchronize with the alpha-blended version. This allows the operator to perform error recovery when the remote robot fails. This offers a safety measure in case the remote. Figure 16 shows the alpha blended objects in yellow.

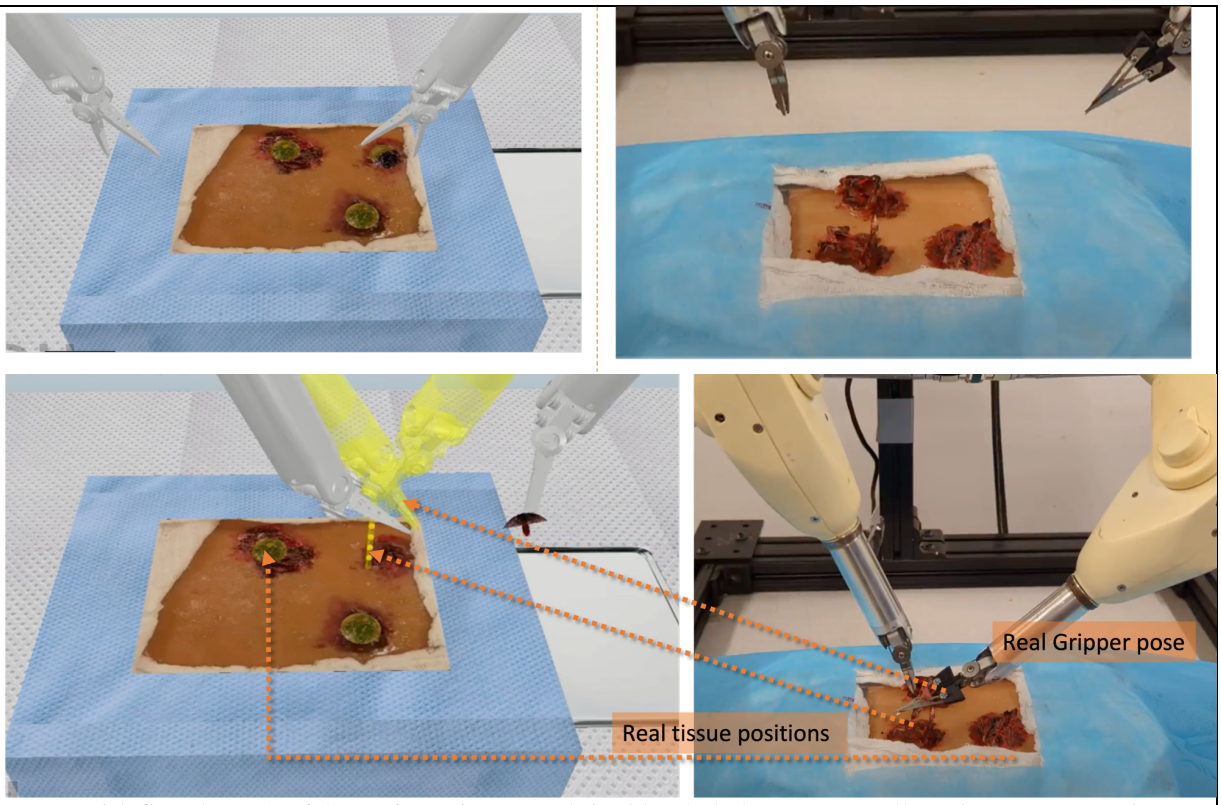


Figure 16. Simulator (Left)interface showing alpha blended elements in yellow that represent the status of the remote scenario (Right)

2.1.4.2. Recognition Module

The operator works with a simulation that corresponds to the remote robot's environment. A recognition module automatically identifies high-level surgical actions from the trajectories and video scenes. The recognized high-level surgical actions are sent to the remote robot through a communication network continuously. To leverage the history of activity data, the recognition module predicts surgemes using an LSTM architecture. A notation for the LSTM is defined as follows. Let each surgeme instance be a sequence of kinematic and video frames of length T, with true labels $\{y_t\}_{t=1}^{T}$where $y_t \in \Delta K$ and ΔK is a probability simplex in K dimensions for K classes. The LSTM predicts class probability $y_t \in \Delta K$ at time t. As our framework consists of 7 surgeme classes, we have K = 7. We use a cross entropy loss defined in Equation below to train the LSTM network.

$$L(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{t=1}^{T} \sum_{k=1}^{7} \boldsymbol{y}_{t,i} \log(\hat{\boldsymbol{y}}_{t,i})$$

As compared to the earlier methods that use a transfer learning framework, or use fixed-length sequences , using an LSTM to predict the surgeme at every time step allows identifying the surgeme class accurately before the operator completes the action. Further, this early surgeme prediction allows DESERTS to mitigate the impact of delays in the system.

2.1.4.3. Communication Module

The communication module is responsible for the data transfer between the operator and the robot side by facilitating message passing, including high-level surgeme commands and feedback information. The communication module has two components: (1) Operator to Robot: The identified

surgeme in simulator is sent to the remote robot. In case of any disruption (delay or interruption), the robot needs to get the missing surgeme information from the operator, without data loss, when the connection re-establishes. To ensure this, surgemes are communicated using a TCP protocol. (2) Robot to Operator: The operator has to be updated with the latest feedback as early as possible. In case of disruption (delay or interruption), the most recent feedback is be sent to the operator when the connection re-establishes. The proirity here is to send the latest data rather than to avoid packet drop. Thus, a UDP protocol is adopted for this path to alleviate the costly handshakes and to update the simulator with the latest feedback messages.

2.1.4.4. Execution Module

After receiving the surgeme from the simulator via the communication module, the robot executes it in the real environment. The robot receives the surgeme label, prediction confidence and the surgeme parameters from the recognition module. The robot executes the surgemes only if there is sufficient confidence in the classification. The threshold for surgeme execution is set empirically. The recognition module can classify the surgeme with high confidence at an early stage of the execution, then the remote robot starts performing the surgeme before the operator finishes thereby reducing the lag between the operator and robot execution. Surgeme Execution Unit. This unit performs a model-based execution of the surgemes to complete the peg transfer task. It requires two inputs from the recognition module; surgeme label and parameter; to execute a surgeme. Additionally, the model-based execution unit needs to identify the surgeme parameters, for example, the object grasp points. To recognize the surgeme parameters we perform instance segmentation and object shape detection using a Mask-RCNN network. The surgeme parameters such as grasp points can be obtained from the object masks. The execution of the surgemes on the peg transfer task is particularly challenging in a blood-occluded setting [9]. Because the foreground and background in the environment are uniform, and the liquid surface produces reflections, the depth image may be incorrect. To mitigate this issue, the object mask is dilated to fill any holes in the depth image. Figure 17 shows the and segmentation of the peg transfer objects and Figure 18 shows the segmentation of the debridement objects



Figure 17. Top: Object masks obtained from the image segmentation. Botom: The proposed method first obtains the contour of the mask (in blue). It then finds the closest point from the centroid on each edge of the contour (in red). The grasp point is the contact point with minimum distance (in green).

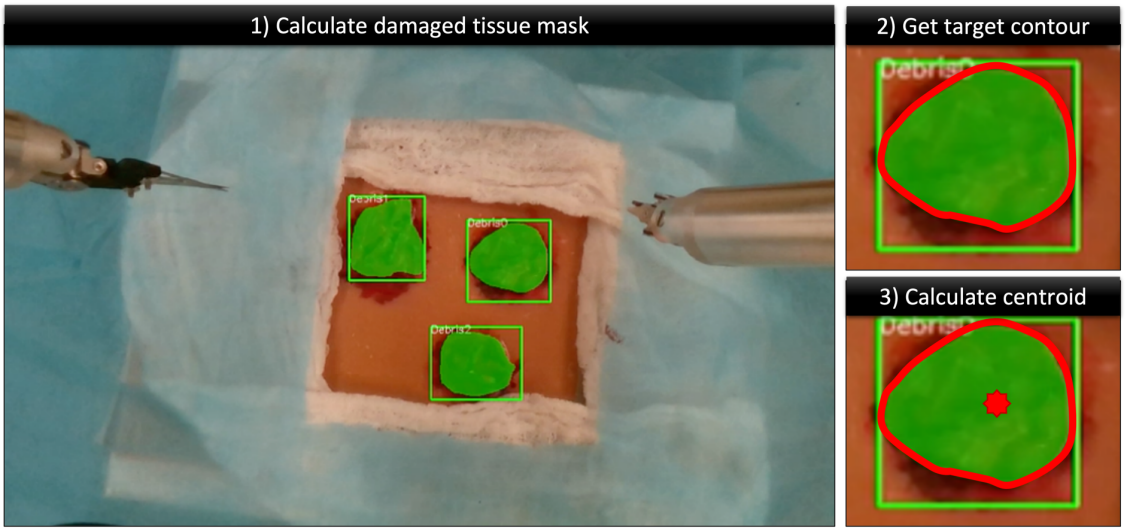


Figure 18. Steps taken for damaged skin segmentation and calculation of the grasp point.

T 2.2 Visual recognition and adaptive learning of primitives

Due to the vast amount of information involved in teleoperation, it is more efficient to use a high-level representation of the environment, patient and robot, to accomplish this task. These main pieces of information would be enough to predict the action being performed by the teleoperating surgeon and complete it autonomously by the robot. In order to effectively interpret the environment, it is necessary to recognize the current and previous surgical actions, referred to as 'surgemes'[10]. Nevertheless, surgical data for austere environment procedures is scarce, hindering the training networks that can automatically recognize these surgical actions. Moreover, field medical robots are diverse, holding different kinematic configurations, workspaces, and operate under partially unknown constraints. We developed a framework aiming to leverage the abundance of data available from more accessible environments to find recurrent patterns and apply such insights to new scenarios. To transfer knowledge between surgical tasks, the surgeme classes of the laparoscopic peg-transfer are mapped to the surgeme classes of the surgical debridement. Then an LSTM-based architecture is trained using sequences of kinematic features from a peg-transfer task. Once the classifier learns to recognize surgemes in the source domain (peg-transfer task), it is not required to be trained from scratch in the target domain (debridement task) but initialized with source domain parameters. Reusing the knowledge learned from the source domain features significantly reduces the amount of training required from the target domain. Figure 19 shows the source and the target domain

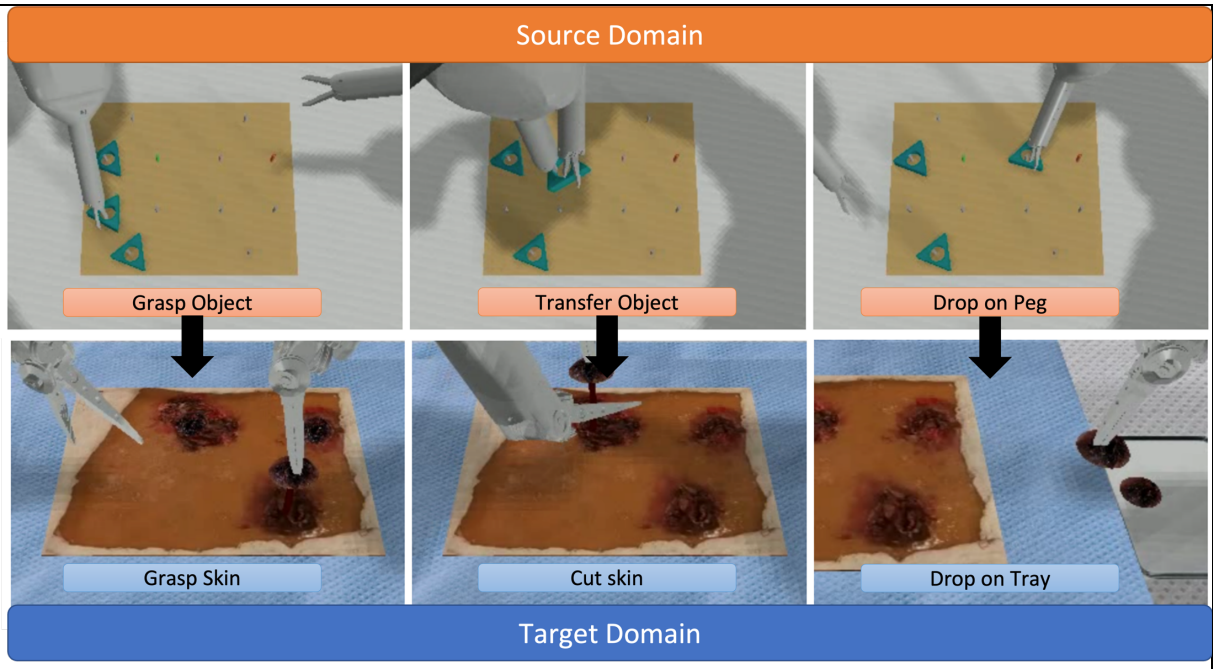


Figure 19. Source domain: Peg transfer task. Target Domain: Surgical debridement task.

We utilized the data DESK dataset that was collected for surgical debridement. This dataset contains 84 debridement trials by 3 subjects, where they were required to remove the dead tissue from a patch of skin.

The operator interacting with the simulator generates a sequence of kinematics corresponding to a surgeme. To classify the kinematic sequence, we use an LSTM based neural network was used in combination with the a neural network.

We use the simulated robot kinematics as the input to the classifier. Let $\{x_t\}_{1<t<T}$ be the input kinematic sequence where x_t is the simulated robot kinematics at time t, and T is the length of the surgical task. Note that a human operator generates the sequence data in an online manner instead of generating complete sequence. This requires making the surgeme classification task to be causal as it cannot depend on future kinematics to predict the current surgeme. To make the neural network classifier causal, we pass the kinematics $x_t$as the input to the LSTM at time t. Further, as the length of each each surgeme is variable, we cannot use methods for fixed length sequences such as *3D-CNN*.

In our network architecture, the LSTM layer is followed by a dense network to compute logit values. We used softmax activation on logits to classify the input sequence into the 7 surgemes. Let the true labels be $\{y_t\}_{t=1}^{T}$ where $y_t \in \Delta^K$ and Δ^k is a probability simplex in K dimensions for K classes (for 7 surgemes $K = 7$). The LSTM is used to predict the class probability $\hat{y_t} \in \Delta^K$ at time t. We use a cross entropy loss defined defined in Equation (1) to train the LSTM network.

$$L(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{t=1}^{T} \sum_{k=1}^{7} \mathbf{y}_{t,i} \log(\hat{\mathbf{y}}_{t,i})$$

To implement the described network, we configured the LSTM network with a hidden state size of 32 followed by a fully connected layer of size 32×7. The final softmax layer returns the probabilities

distribution of the current surgeme given the current kinematic input. The surgeme that will be eventually executed is the one with the highest probability given the input kinematic x_t.

The Experiments are results are described in Section 3.2.

T3.1 Motor command generation in constrained and new settings

3.1.1 Surgeme Performance using reinforcement learning

Previously Proximal Policy Optimization (PPO) was used with a hand shaped reward to learn the approach and grasp surgemes. This methodology achieved an average surgeme success rate of 81 %. However, using Hindsight experience replay (HER) with Deep Deterministic Policy Gradient (DDPG) eliminates the need for extensive hand shaped reward. Instead, only a sparse reward at end of surgeme is sufficient to learn the surgeme. Using HER + DDPG resulted in 100% success rate for the approach surgeme.

Preliminaries:

A policy can either be stochastic or deterministic. A stochastic policy is denoted by:

$\pi_\theta: S \rightarrow P(A)$ which defines the probability distribution $P(A)$ of performing an action.

Or deterministic, denoted by $\mu_\theta(s_t)$ which is a discrete mapping of $S \rightarrow A$. In both cases, θ refers to parameters of the function estimator used to represent the policy. With policy gradient we want to find a policy that maximizes an objective function representing the return or sum of rewards, denoted $R(\tau)$, where τ represents a surgeme trajectories consisting of state action pairs $\tau = (s_0, a_0, s_1, a_1, \dots, s_T, a_T)$:

$$J(\theta) = E_\tau[R(\tau)] = E_\tau[\sum_{t=0}^{T} \gamma^t, r(s_t, a_t, s_{\{t+1\}})]$$

where, $r(s_t, a_t)$ refers to the reward function for a given state and action taken. γ^t refers to the discount factor at time t. We can maximize the objective function using the standard approach of gradient ascent/descent by updating the parameters using:

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$

How we compute $\nabla_\theta J(\theta)$ is more difficult and depends on the algorithm, but a general policy gradient is then given by:

$$\nabla_\theta J(\theta) = E_\tau[\sum_{t=0}^{T} \nabla_\theta \ln \pi_\theta(s_t, a_t), R(\tau)]$$

The return of the sampled trajectory can be replaced by the action-value function which measures expected return for a given state and action ($Q^{\pi(s,a)}$). However, this has problems with high variance of the gradient. To reduce variance, the state-value function ($V^{\pi(s)}$) which measures the expected return of a state can be used as a state-dependent baseline. Additionally, ($Q^{\pi(s,a)}$) measures the

expected return given a state and action pair. Thus, the advantage function which describes how much better it is to take an specific action a in state s over a random action can be defined:

$$A^{\pi(s,a)} = Q^{\pi(s,a)} - V^{\pi(s)}$$

We can substitute in the advantage function to update the objective. Additionally, we can obtain the loss of the policy gradient by integrating the objective function:

$$L = \int \nabla_\theta J(\theta) = E[ln\, \pi_\theta(s,a)\, , A(s,a)]$$

However, performing optimization with this loss results in destructively large policy updates. Methods such as Trust Region Policy Optimization maximize a surrogate objective subject to a constraint on the size of the policy update. PPO simplifies this to instead using specialized clipping to prevent the updated policy from moving too far from the old policy. Finally, PPO uses an actor-critic style algorithm where one network, the actor, wich is optimized using the aforementioned gradient, learns the policy. A separate network, the critic, learns the $V(S)$ which can be used to calculate the advantage.

3.1.1.1 Deep Deterministic Policy Gradients and Hindsight Experience Replay:

Hindsight Experience Replay (HER) is a technique for off-policy reinforcement learning algorithms which allows for sample-efficient learning from rewards which are sparse or binary, which also avoids having to use reward engineering. Additionally, it supports environments with multiple goals. While off-policy models can be less stable, they generally have increased sample efficiency. The key idea behind HER is that in a given episode, even if we have not succeeded at the end goal, the robot has at least achieved a different goal. This goal substitution is repeated many times, and eventually many arbitrary goals will be learned, including the target goal which we would like to achieve. Off-policy algorithms already replay experience, thus HER can be combined with any off-policy algorithm. We chose to use Deep Deterministic Policy Gradient (DDPG) which has been shown to work well with HER. DDPG is a model-free, off-policy , actor-critic algorithm which essentially an extension of Deep Q-Networks for continuous action spaces.

3.1.1.2 Results:

HER was found to work well for the approach action. Figure 20. shows results from training with 10 parallel MPI processes with physics disabled, as interaction with the peg is not necessary for the approach task. To summarize, the model was able to converge in only 9 epochs or 7 hours of real training time. Training with physics enabled also performs well, with convergence in about double the number of epochs. To compare, the previous PPO implementation took ~ 20 times the number of time-steps to converge.

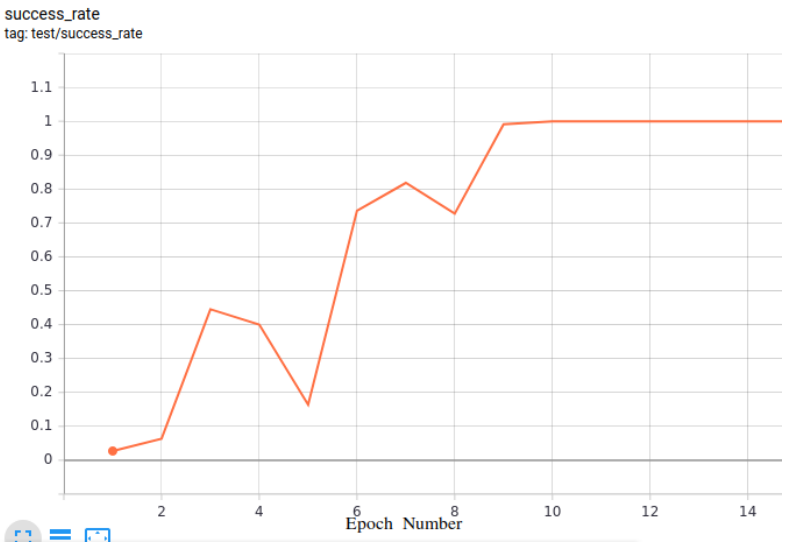
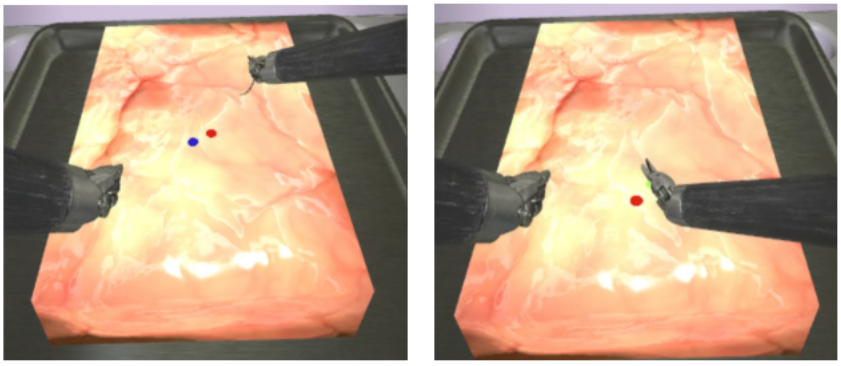


Fig 20. Success rate for approach surgeme with HER + DDPG

3.1.2 Reach and suture execution using Reinforcement learning

Using our dVSS (da Vinci Surgical Simulator) and the environment we created (dVSS-RL), we performed RL on two tasks, the Reach task and the Suture task, as seen below in Figure 21.



(a) Reach task view (b) Suture task view

Fig 21. Da Vinci surgical simulator

The Reach task consists of approaching the target in the simulator, while the Suture task involves guiding the needle through 3 phases at the ingress and egress site (see Figure 22).

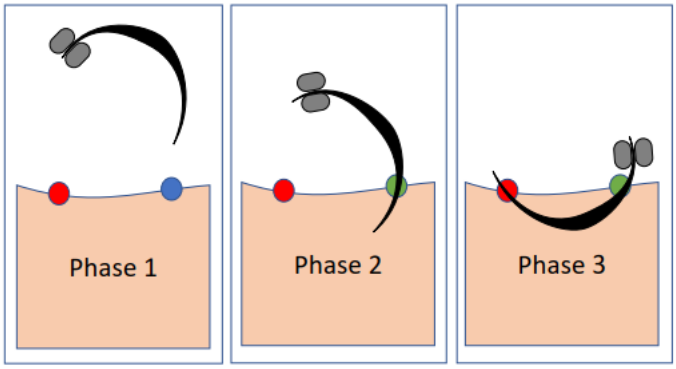


Fig 22. Suture phases.

We compared different modes of operation: image-based, mixed image and state data, and depth and image-based (see Figure 23). We found that we had to use an RL (Q) loss together with an L2 loss to allow the network to learn – learning directly from images with only an RL loss was too difficult for the network.

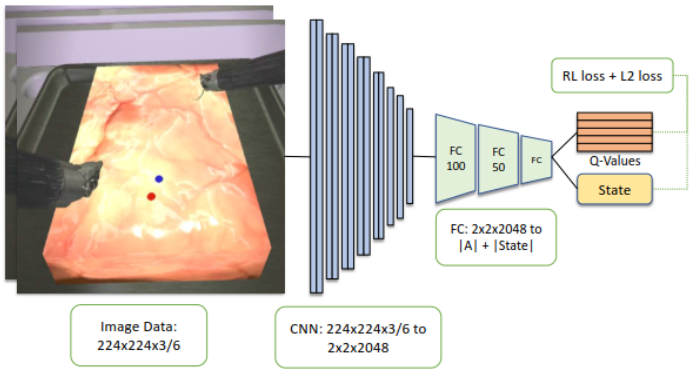


Fig 23. Image based neural network.

In order to accelerate the relatively slow and complex dVSS-RL beyond real-time, we invented a playback-environment system which we call Hybrid-Batch Learning (HBL), as shown in Figure 24.

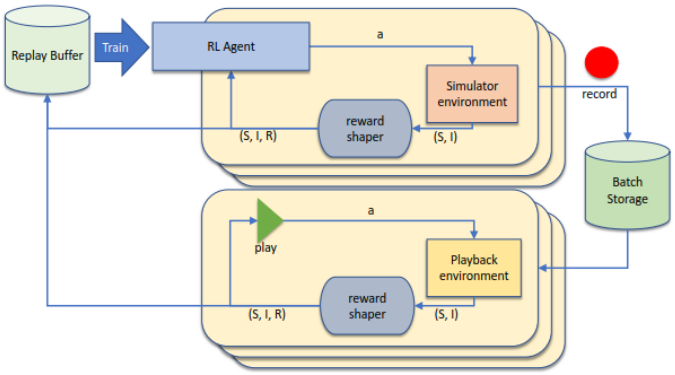
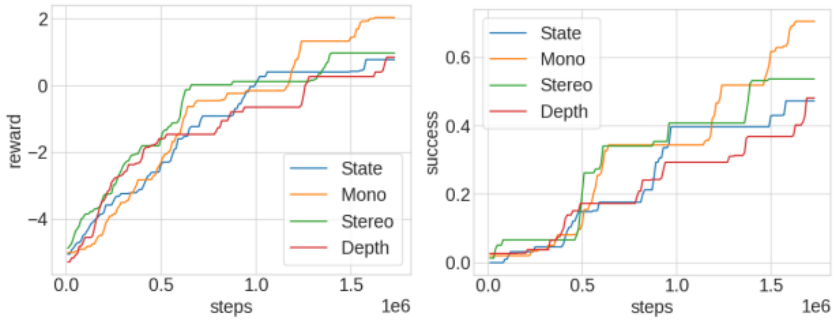
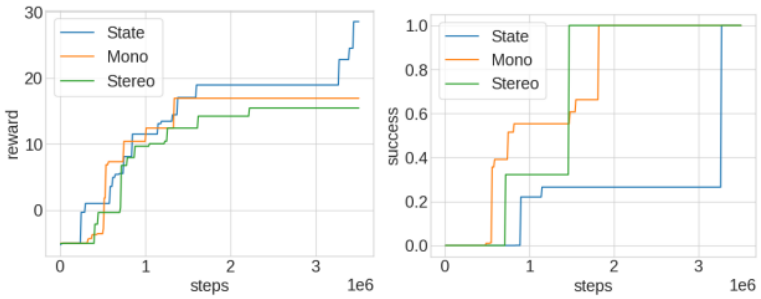


Fig 24. Hybrid Batch Learning architecture.

This system allows us to record entire episodes and play them back simultaneously while running other parallel environments in real-time. The updated results for the Reach task and Suture task are shown in Figure 25.



(a) Reach task rewards (b) Reach task success



(a) Suture task rewards (b) Suture task success

Figure 25. Accumulated reward obtained for the reach and suture tasks.

We tested the HBL system at different ratios of playback data to real-time sim data, and found that good results were obtained even at rates as high at 96% playback (see Figure 8). Additionally, HBL allows us to use CHBL or Curriculum HBL, where we use a curriculum learning-like approach to start with easier problems (such as state-based learning) and use that data for harder problems (such as image-based learning).

In order to investigate the effect of CHBL we used our simpler Needle Master environment. We found, to our surprise, that CHBL improved not only wall-clock time, but also learning time in terms of required steps, as show in Figure 26.

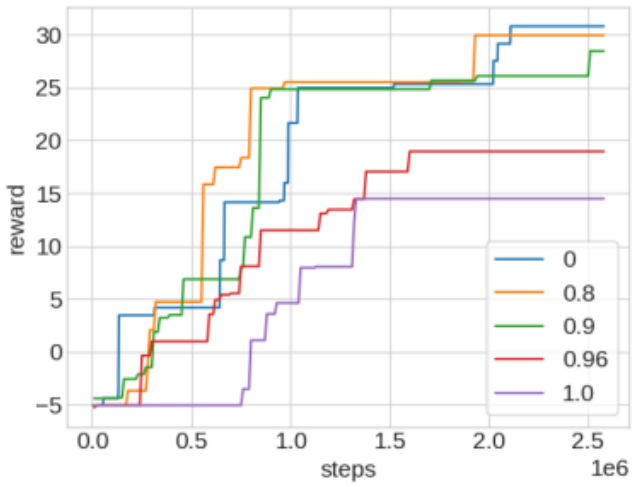
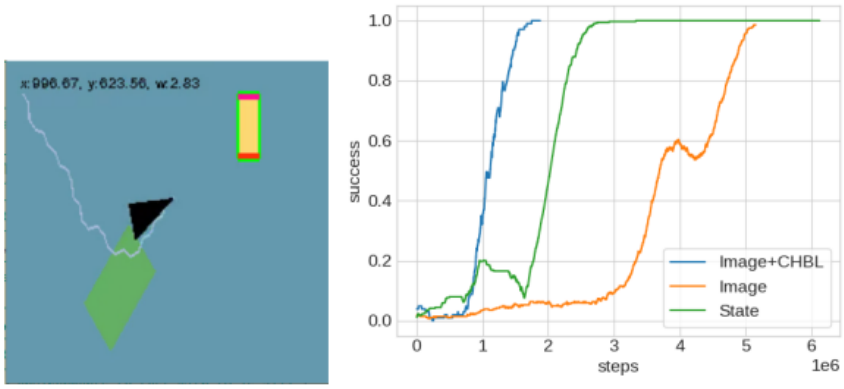


Fig 26. Reward obtained using at different playback rates.



(a) Agent view of NMRL (b) Average success vs steps in NMRL

Fig 27. Success rate for HBL and CHBL.

In order to further explore HBL and CHBL and the difficulty our RL methods had in learning on the dVSS-RL, we created a 3D version of Needle Master: Needle Master 3D (see Figure 27).

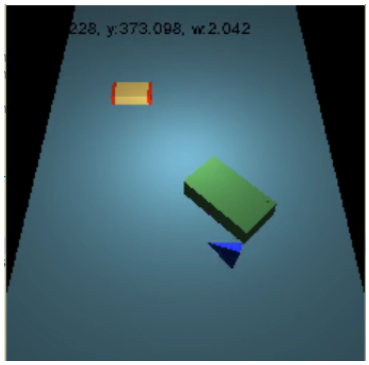
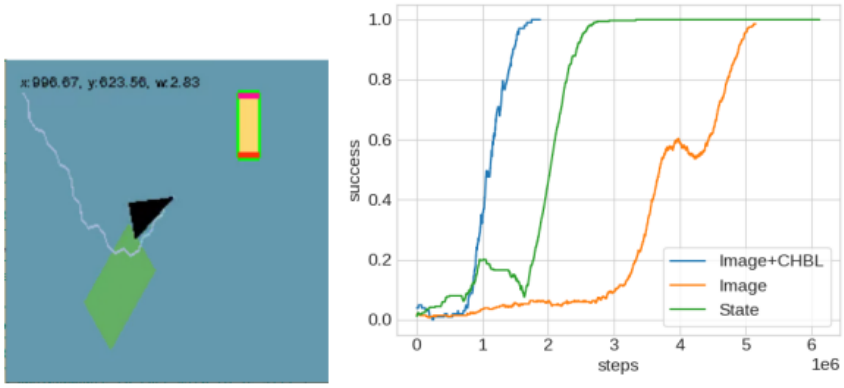


Fig 10 . Needdle master 3D environment.

3.1.3 Exploration of Needle Master

After successfully training on the da Vinci simulator, we applied reinforcement learning algorithms to the Needle Master (NMRL) and Needle Master 3d (NM3DRL) environment, with the goal of training a similar real-world environment.



(a) Agent view of NMRL (b) Average success vs steps in NMRL

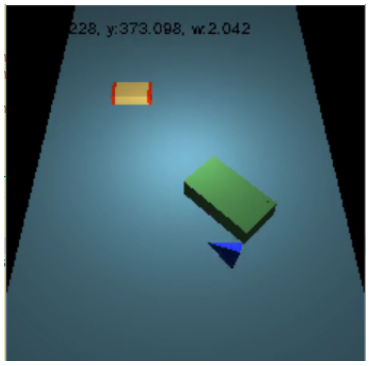


Figure 28. Performance on the Needle master system.

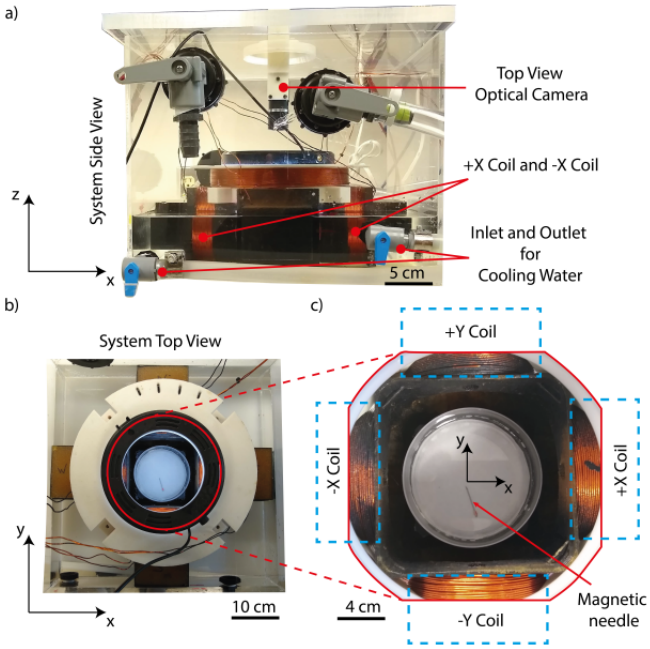
The NM3DRL environment allows us to explore more realistic-looking environments than NMRL, as well as another degree of freedom. We can solve both environments reliably for different configurations of objects, as shown in figure Figure 28.

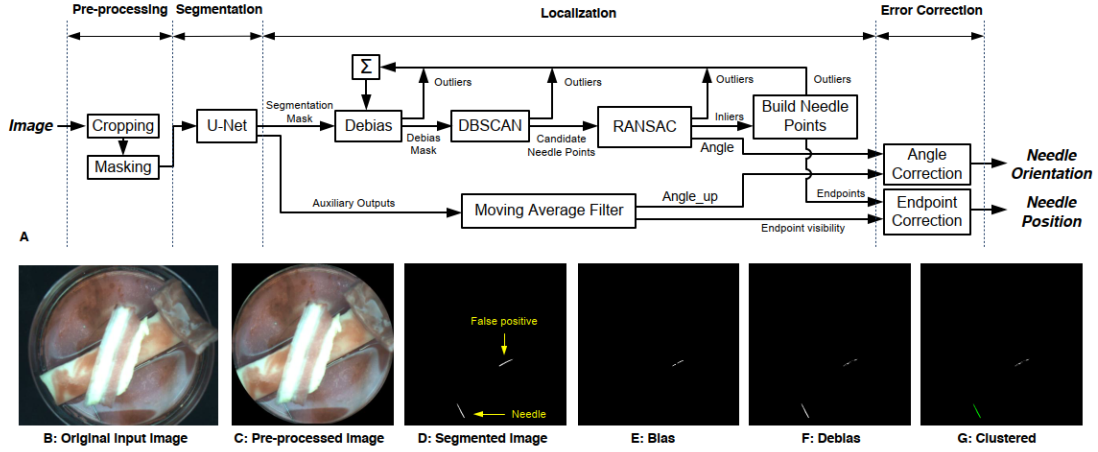
3.1.4 Debridement segmentation and execution

The debridement task requires removal of dead tissue or dead tissue fragments to allow the underneath healthy tissue to heal. This requires manipulating the robotic arms such that it grasps the diseased tissue, incise it, and then safely discard the bio-waste. Considering these tasks, we divide a debridement task robotic maneuver into 7 surgemes (labelled as $Si \in \{1,\cdots,7\}$ as follows:

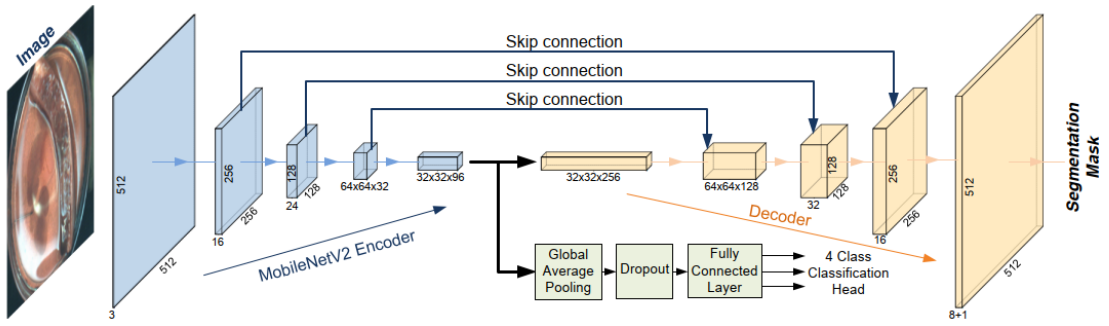
3.1.4.1 Localization of a magnetic robotic needle

We used a combination of deep learning methods and classical algorithms to attain accurate localization for a magnetic system called MagnetoSuture™. The system is composed of a robotic needle actuated by 4 electromagnets.



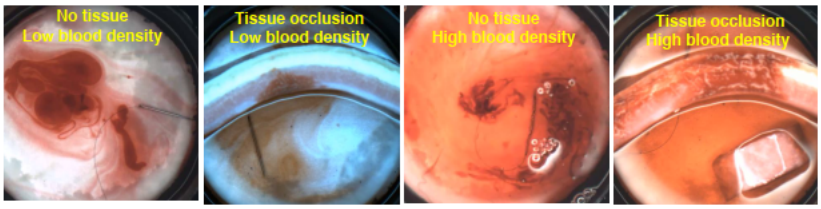


We built the system using a combination of deep learning, producing a detailed segmentation map, and a high-level combination of algorithms: DBSCAN for clustering, a filtering heuristic involving pixel clusters of size similar to the needle width, and RANSAC for fitting lines through the filtered clusters. We also keep track of outliers over time to debias the algorithm so as to ignore outliers that persist over time.

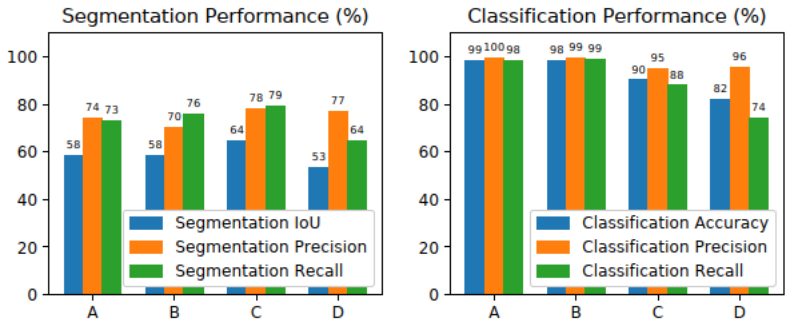


The deep learning system was composed of a MobileNetv2 encoder (for performance), followed by a decoder. We further split the decoder head, using global average pooling to obtain additional information, such as what the full orientation of the needle is, and whether the tip is visible.

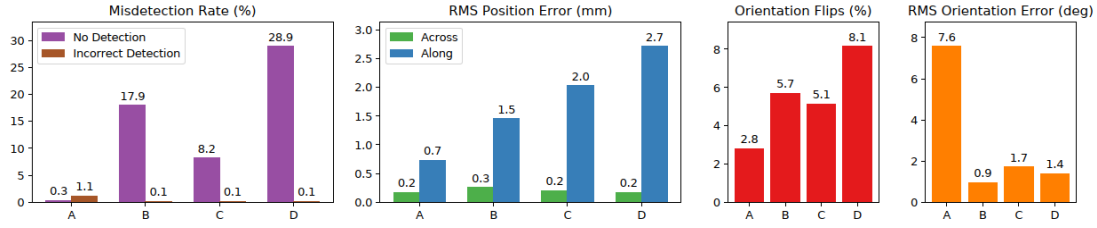
We collected and annotated image data over several videos, grouping data into 4 different classes.



We analyzed our segmentation and classification performance in terms of IOU, accuracy, precision and recall.



While our segmentation performance was decent – particularly precision – our classification accuracy, precision and recall were far better. This is due to the fact that our ground truth segmentation maps were made crudely, and the neural network learned the task far better than was reflected by the pure scores on the ground truth segmentation maps.



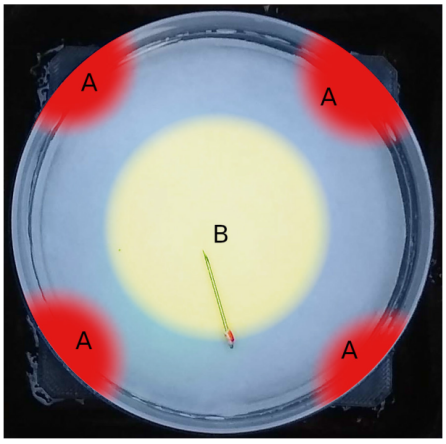
We noticed that our RMS position errors were low (only a couple of millimeters), but we suffered from orientation flips due to an insufficient accuracy on our network's classification head.

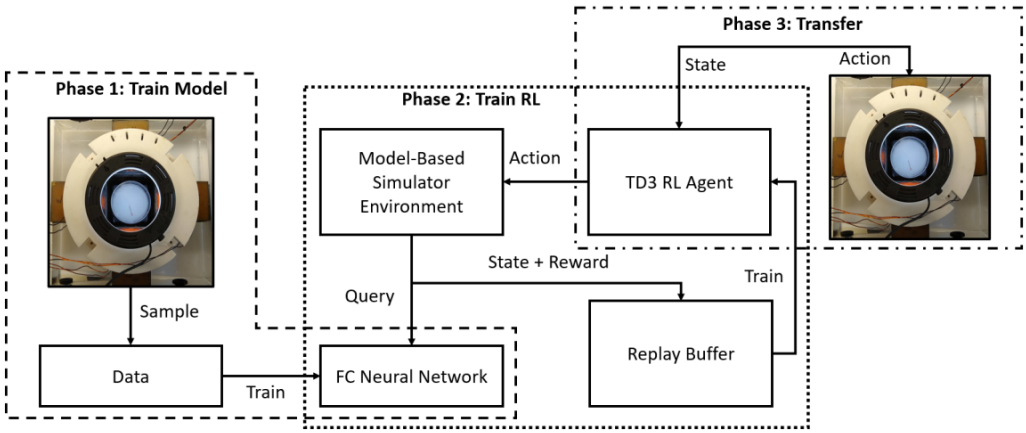
3.1.4.2 Closed loop control of a magnetic robot via reinforcement learning

After obtaining good performance on localization, we proceeded to train the same MagnetoSuture™ system for closed-loop control using reinforcement learning.

We first noticed that the orientation flips we experienced during localization were causing inaccuracies in our dataset that were intolerable for the purpose of reinforcement learning. We therefore added an additional orientation-detection approach that utilized a colored marker on the back of the needle and color-aided localization on top of the algorithm mentioned above.

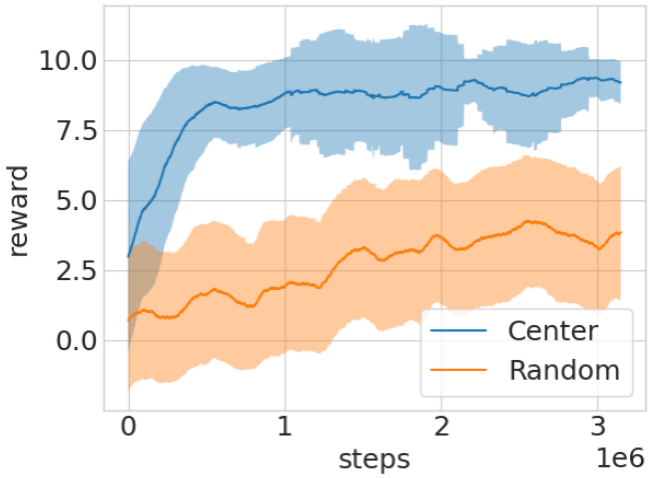
We also noticed that movement within the Petri dish in the system performed best in the area labeled 'B' below, whereas areas labeled 'A' were effective dead zones due to the magnetic dynamics. As shown in the Figure 27





We first tried to use a simulator to train the RL algorithm for transfer to the real world. This attempt failed, as we found that the slightest change in parameters caused the simulator dynamics to diverge from the behavior of the real world due to the sensitivity of friction forces and the complexity of the magnetic fields.

We then attempted a different approach, first exploring the Petri dish and collecting data, then fitting a neural network to the data so as to learn the system dynamics. We used the learned model as the heart of a pseudo-simulation with which to train the RL system.

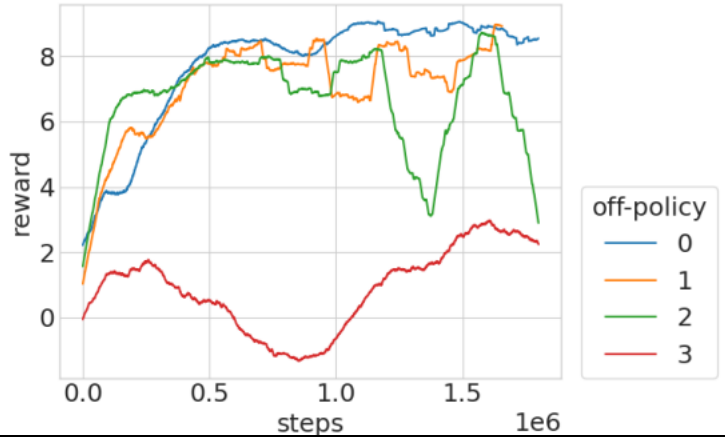


Running the RL algorithm on the pseudo-simulator (which we termed a MBS – model-based simulation), we trained on two separate tasks – moving to the center of the dish, and moving to a random, user-provided location (see Figure 28). The former task was far easier, while the latter task was far more challenging and thus took longer to train. We were unable to reach the same maximum reward for the second task due to the fact many areas in the Petri dish are impossible to navigate.

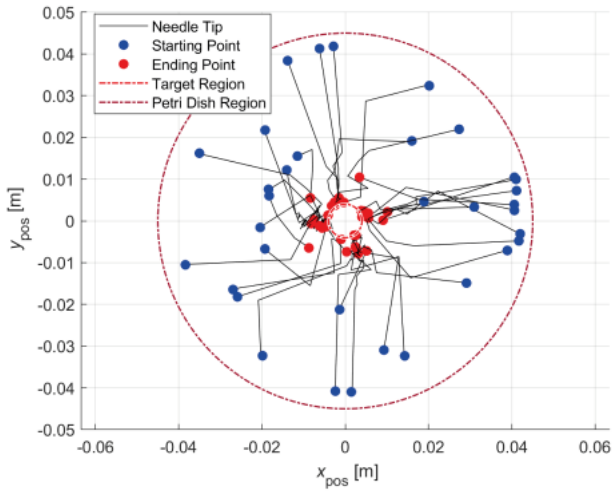
We also compared the training times of the different approaches: MBS, simulator, and real-world training. Real-world training in particular was completely infeasible due to the many hours required for real-time RL learning. The system is experimental and can overheat due to electromagnet operation, making real world operation for the required amount of time completely infeasible.

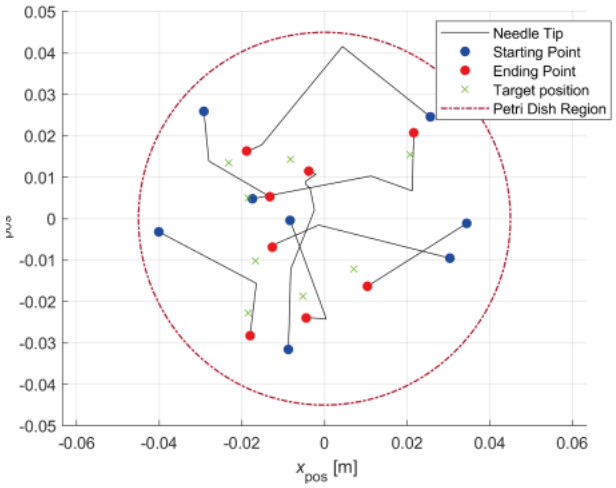
RL method	task	training time (hrs)
MBS	center	1.5
MBS	random	3.5
sim	center	5.5
sim	random	27
real	center	NA*

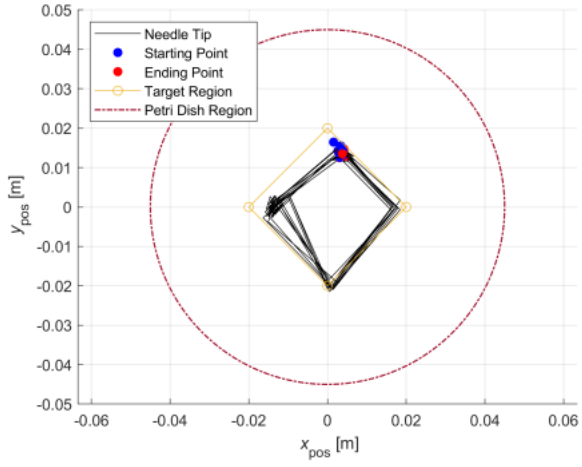
We also tested using the same collected data which we used to train the MBS for off-policy RL. We mixed off-policy and on-policy environments, and noted that having 3 out of 3 off-policy environments was unstable, while mixing on-policy environments and some off-policy ones in parallel can provide a minor performance boost while causing instability in RL training. Seemingly, the off-policy data was insufficient to train RL directly, but was sufficient for training the MBS.



Finally, we tested transfer to the real world on 3 tasks: moving to the center, moving to a user-provided random location, and following a square path.







We compared the performance of the RL agents on these 3 tasks, including the sim-based RL agent. We noted that the sim-based agent completely failed to transfer, whereas the MBS-based controller transferred successfully.

RL agent	test	mean error (mm)	stdev (mm)	transfer
Sim-based center	center	31	29	no
MBS center	center	6.3	2.1	yes
MBS target	point	5.3	0.8	yes
MBS target	box	7.4	0.7	yes

3.1.5 Learning cricothyroidotomy execution through RL

Cricothyroidotomy is an emergency procedure which is included in first aid training. The procedure is short and is critical to saving lives especially in battlefields. The proposed framework is adopted to perform remote surgical cricothyroidotomy. A difference between peg-transfer task, which is simulated laparoscopy, and cricothyroidotomy is mode of sensing. In cricothyroidotomy the surgeon uses palpation to understand the profile of the trachea and identify the location for incision. Thus, force sensing is paramount to performing surgical procedures such as tracheotomy or cricothyroidotomy.

3.1.5.1 Surgical end effectors for cricothyroidotomy

The remote surgical procedure for peg-transfer was implemented on the ABB Yumi with a modified surgical end-effectors to simulate the laproscopic tools. The cricothyroidotomy is implemented using two Rethink Sawyer robots. The end-effectors of the robot are modified to resemble the laproscopic end-effectors. The end of the tool is designed to accommodate different tools such as scalpel, forceps and force sensors. Figure 29 shows the designed end-effectors with the tools mounted.

(a) Syntouch force sensor (b) Scalpel

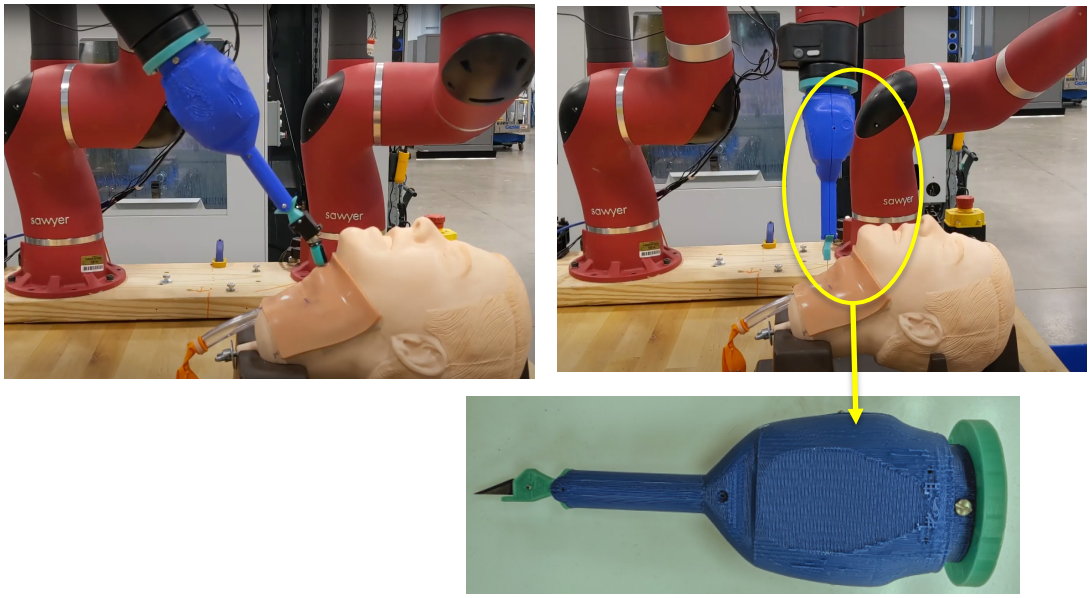


Figure 29. The designed surgical end-effectors with different tools.

3.1.5.2 Simulator development

The operator performs the task in a simulated environment which replicates the real setup. Similar to the peg-transfer and debridement tasks, a simulator is developed for the cricothyroidotomy task. In addition to being the interface, the simulator is leveraged to collect demonstrations of surgeme performance to learn the surgeme execution for the robot. The demonstrated surgeme data are used to instantiate reinforcement learning policy for executing the surgemes. Figure 30 shows the model of the developed simulator. The simulator is developed in CoppeliaSim simulation software. The neck is modeled based on the real trachea to simulate the ridges necessary for palpation. It is covered with a soft skin membrane with varying stiffness across the surface and along the depth.

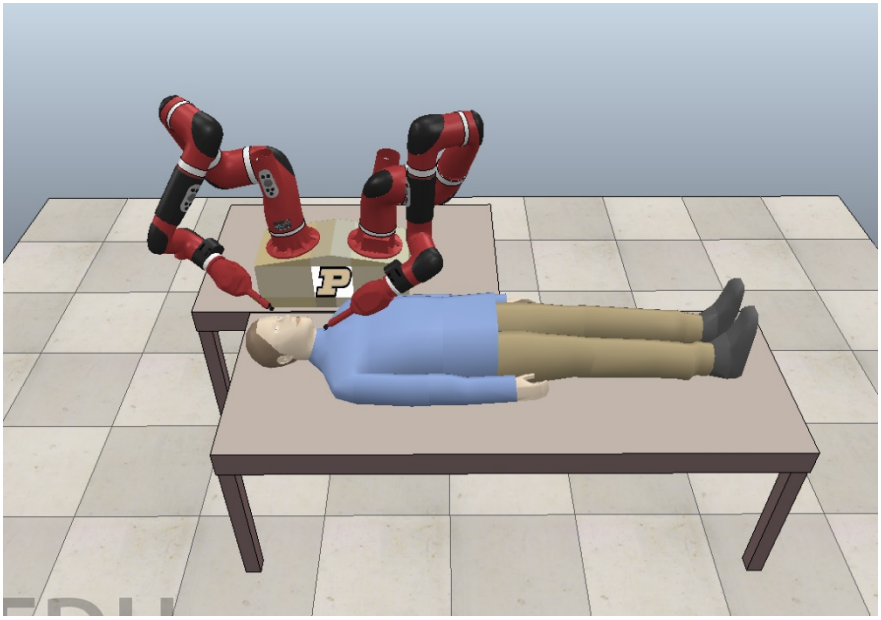


Figure 30. Cricothyroidotomy simulator.

3.1.5.2 Cricothyroidotomy surgeme execution: (palpation and incision)

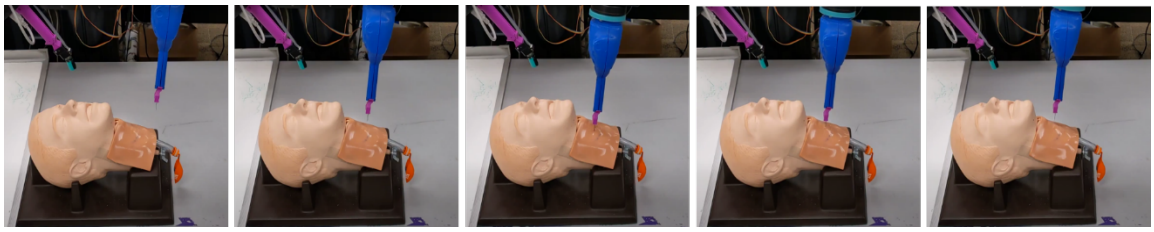
The robot learns the execution from teleoperated demonstrations of the surgeme. The execution is done using a reinforcement learning policy. The action space and the reward are inferred from the demonstration in the simulator. The surgemes in cricothyroidotomy require both force and position control, and feedback from force, tactile and camera to complete the task. Thus, a multi-modal execution approach is implemented.

The execution framework to learns policies for the surgemes from few demonstrations without explicit reward engineering. The teleoperated demonstrations are semantically segmented to identify tool-object interactions and goal states. Subsequently, the reward can be learned from goals identified in the demonstrations, which removes the need for manual engineering of reward. Additionally, the interaction modes are used to define the action space for the policy. A multi-modal representation for the state is developed from the sensor data, which improves the learning performance of the surgemes. The surgeme models are first trained in simulation and then translated to the real robot. An example

of the robot execution is available at [1]. Figure 31 shows the sequence of surgeme execution for the task.



(a) Palpation surgeme execution



(b) Incision surgeme execution

Figure 31. The surgeme execution on the Sawyer robot showing the (a) palpation and (b) incision sequences

T3.2 Experimental Design

3.2.1 Experiment 1: Metrics to assess movement generation plan

The we used our DESK dataset to extract kinematic of data collected from a simulated YUMI robot (see Figure 1) which can be controlled by the operator. A sample of the motion trajectory along x-axis is shown for reference in Figure 32. The task goal is to infer the action performed by the operator and the target position of the operator. The inferred action and target is communicated to a distant robot and a real robot executes task. The target prediction will help in start performing the action early and reduce the latency in the process.

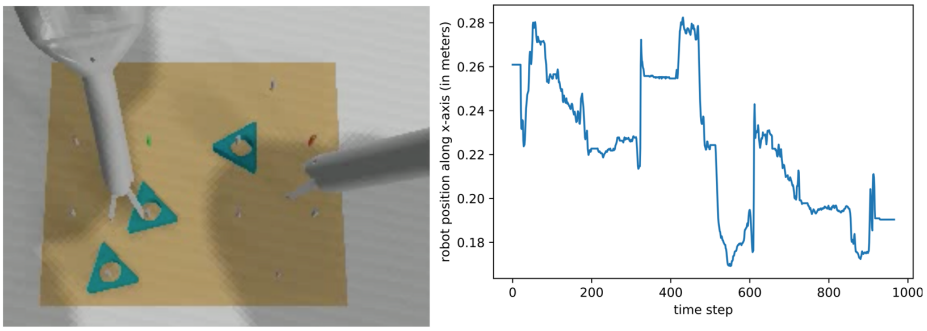


Figure 32. Left: Simulated YUMI Robot performing peg transfer. Right: Sample trajectory of the robot arm along

15. For our experiments with Gaussian Processes, we choose a kernel that combines the kernels described in section 3.1. We used Python 3 and scikit-sklearn Gaussian Process to implement the

code. The training data consisted of a total of h length segments generated from two trajectories. Using the entire dataset of 12 trajectories (12000 robot poses) becomes computationally infeasible for Gaussian Processes.

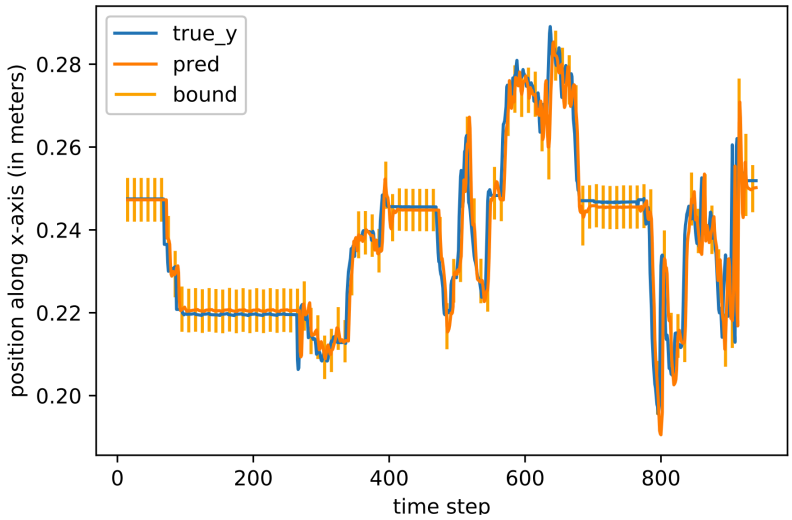


Figure 33. Trajectory and the confidence intervals predicted by Gaussian Process for the robot along x-axis for k = 4, h =15.

Lastly, we compared the performance of the Gaussian process approach with LSTMs. We used a linear activation for last layer and LSTM with 64 hidden cells. For training, we use Stochastic Gradient Descent optimizer with a constant learning rate $\eta = 0.2$. We train the LSTM for 400 epochs. For LSTM, we perform a 70:30 split for training and testing.

We present the average Euclidean errors for the Gaussian Process model in Table 3. From Table 3, we note that adding more history does not help in predicting distant future. This behavior is consistent across all future lengths. We suspect this is because the final few coordinates capture sufficient information such as velocity and acceleration to predict the next position along the trajectory.

We also compared the prediction performance of the two strategies, Gaussian Process based, and LSTM based. We note that the Gaussian process model can learn the correct relationship between the predicted data and the input data for the case of prediction for the current step. We next compare the performance of the models when they require a prediction, i.e. $k > 0$. The results are presented in Table 3.

We also present the predicted trajectory obtained with the Gaussian Process in Figure 33. The Gaussian process also provides confidence intervals for the predicted positions, which can be used to generate the cone of uncertainty around the trajectory.

We also present the predicted trajectory obtained with the Gaussian Process in Figure 2. The Gaussian process also provides confidence intervals for the predicted positions, which can be used to generate the cone of uncertainty around the trajectory.

Table 3. Average Euclidean distance (in meters) of the trajectory using Gaussian Process

h	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
2	5.83×10^{-5}	0.003	0.004	0.007	0.009
5	6.71×10^{-5}	0.003	0.005	0.007	0.01
10	1.28×10^{-4}	0.003	0.005	0.008	0.01
15	1.3×10^{-4}	0.003	0.006	0.008	0.01

Table 4. Average Euclidean distance (in meters) of the trajectory using LSTM. Note that there is no particular history as LSTM as expected to run using arbitrary history lengths.

$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
0.073	0.106	0.156	0.210	0.273

Table 4 shows the average Euclidean errors for the LSTM based neural network in. This model produces an error of 7 cm when predicting the current position (k = 0). The prediction error only increases when the model is used to generate future positions.

3.2.2 Experiment 2: Metrics to assess learning with Imperfect and Randomly Delayed Information:

This approach considers a scenario of surgeme classification where the information about the "ground truth" label of the previous surgemes and the current surgeme might delayed. The sections below explain both setups in detail.

3.2.2.1 Delayed feature label approach

Experimental Setting: For each experiment, we split the data into training and testing. To emulate the real-world scenario, we maintain chronological (temporal) information. Thus from a data splitting point, we consider past data as training data and the rest as testing. We consider accuracy as the evaluation metric which is the ratio between the number of correct predictions and total testing instances. For our preliminary experiment, we use JIGSAWS suturing kinematic data. We leverage a multiclass SVM with 'rbf' kernel as our classifier.

Experiment 1: We split the data into three different ways as shown in Table 5. We evaluate each testing instance independently and calculate the accuracy leveraging the annotated JIGSAWS data. In this setting, we do not consider temporal or delay information. We observe that the classifier achieves the best result in the 90/10 (train/test) splitting with an accuracy of around 69%.

Experiment 2: Split data into train and test. Consider the previous step's ground truth label as a feature in the training instance. The dataset splitting is similar to experiment 1. During training and testing, we consider the previous step's ground truth label (not predicted) as an additional feature. In all the settings, we observe high accuracy in detecting surgemes with the best accuracy is around 98% in the 70/30 splitting set. This shows that the previous step's ground truth label is important in classifying new surgemes. Thus incorporating this historical label data would give a better

prediction accuracy. On the other hand, in this experiment, we assume that we would always get ground truth label. However, in case of any delay in getting the actual labels of the previous surgemes, we would not have the ground truth data for all the testing instances. To emulate this scenario we design the next experiment which we describe below.

Experiment 3: In this experiment, we consider both the temporal data and delay in getting the actual label of each surgeme from the surgeon. We design several settings corresponding to different delay length. Firstly, we consider that we will miss ground truth labels for alternating surgemes which is represented as "1:1 GT:MISS" in Table 6 . We observe that the accuracy remains high despite the missing labels and the classifier achieves and accuracy of 97%. Note that, here the number of missing labels correspond to the length of the delay. Later, we increase delay and consider that we will miss three previous surgemes labels out of four surgemes. Despite this delay (indicated by "1:3 GT:MISS" in Table 6 we observe that the classifier's accuracy remain almost the same. Similarly, later we introduce more delay and thus increase the missing labels. We observe that the performances decrease gradually. Table 6 shows that when the missing labels go above 50, the accuracy decreases drastically. Thus varying the missing labels, we can achieve a trade-off between surgemes detection accuracy and delay.

Table 6. Experimental Design Emulating Delay

Exp.	**Settings**		**Train/Test (%)**	**Accuracy (%)**
1	No temporal information No Delay		70/30	69.06
			80/20	63.50
			90/10	69.26
2	With Temporal information No Delay		70/30	98.31
			80/20	97.84
			90/10	97.49
3	With temporal information and delay	1:1 GT:MISS	80/20	97.53
		1:3 GT:MISS	80/20	97.22
		1:10 GT:MISS	80/20	95.40
		1:50 GT:MISS	80/20	85.30
		1:100 GT:MISS	80/20	73.08

In the setting of delayed information there is very limited work ever done. One of the methods proposed in literature is to augment the underlying Markov Decision Process such that it the agent state becomes the last known environment state and the actions taken by the agent from the last known state to current time step. However, this approach is highly inefficient to implement because of the following reasons:

1. The increase in state space complexity is exponential.

2. Solution is not scalable for stochastic delays.

Further literature survey shows that algorithms take action based on the most likely current state. This scheme is scalable for stochastic delays and doesn't cause an exponential increase in state space. However, this approach is not robust under the regimes where the most likely state itself has low probability. For example, consider a system with five states A, B, C, D, E. At a given instant the probabilities of being in state A is 40%, and being in any other state is 15%. The most likely state is A, but not being in A is much more likely than being in A.

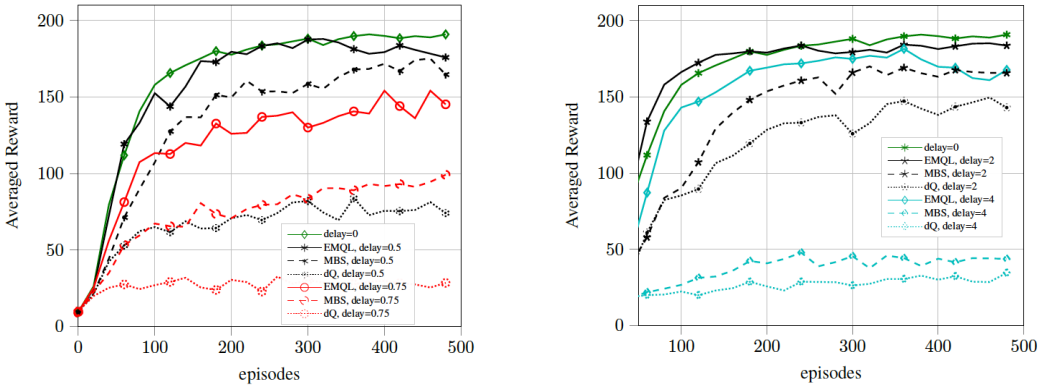
To mitigate the above-mentioned issues, an algorithm was developed to take actions which maximize according to the expected equality of the action, and not the expected state. Quality of an action is defined as the expected total discounted reward, on taking action a in state s. The value of an state is defined as the total discounted reward agent can collect starting from state s.

$$Q(s,a) = E[\sum_{t=0}^{\infty} \gamma^t r \,|s_t = s, a_t = a], V(s) = max_a Q(s,a)]$$

The algorithm selects an action which maximizes the expected value of the current unknown state:

$$a = \ argmax\ E_s[Q(s,a)|s_{t-d}, a_{t-d}, \ldots, a_{t-1}]$$

With the proposed solution (called EMQL), the agent performs better than current algorithms available. It is state efficient, scalable to stochastic delays, and robust under deviation from most likely state.



(a) Performance under stochastic delays. (b) Performance under constant delays.

The figures above compare the following algorithms:

1. EMQL: Proposed and developed algorithm
2. MBS: Algorithm which takes action for the most likely state
3. dQ: Algorithm which is not optimizing for the delays

For stochastic delays, delay probabilities are mentioned. Averaged delay thus becomes $E[d] = 1/(probability\ of\ delay)$.Two settings were compared with average delay of 2 timesteps and average delay of 4 timesteps. Corresponding constant delays were also considered. Performance for stochastic delays where average delay is same as constant delay is better. This is expected as the agent has more probability to take correct decisions.

Proposed EMQL algorithm beats the other two algorithm in performance. Also, when the delays are large, the deviation is from the optimal converging point under no delays is smaller for EMQL

3.2.2.2 Stateful Long Short Term Memory Netork (LSTM)

We created a model using LSTM to use the temporal nature of the data. We used stateful LSTM to work with long sequences which can also have variable length. The input at each time consisted of the last available surgeme and the current kinematics of the robot. We found that using DESK data set we get only 25% accuracy using LSTMs. The performance doesn't improve much after hyperparameter tuning. Since the dataset size is small for a LSTM, we will attempt to use this model in an artificially augmented dataset in future iterations.

We tested the sensitivity of current surgeme on past surgemes or predictability of the data using a fully connected network by varying the input size to include previous h surgeme history. We found that the performance of the fully connected neural network does not significantly increase on increasing h. Detailed results of the experiments are provided in Figure 34.

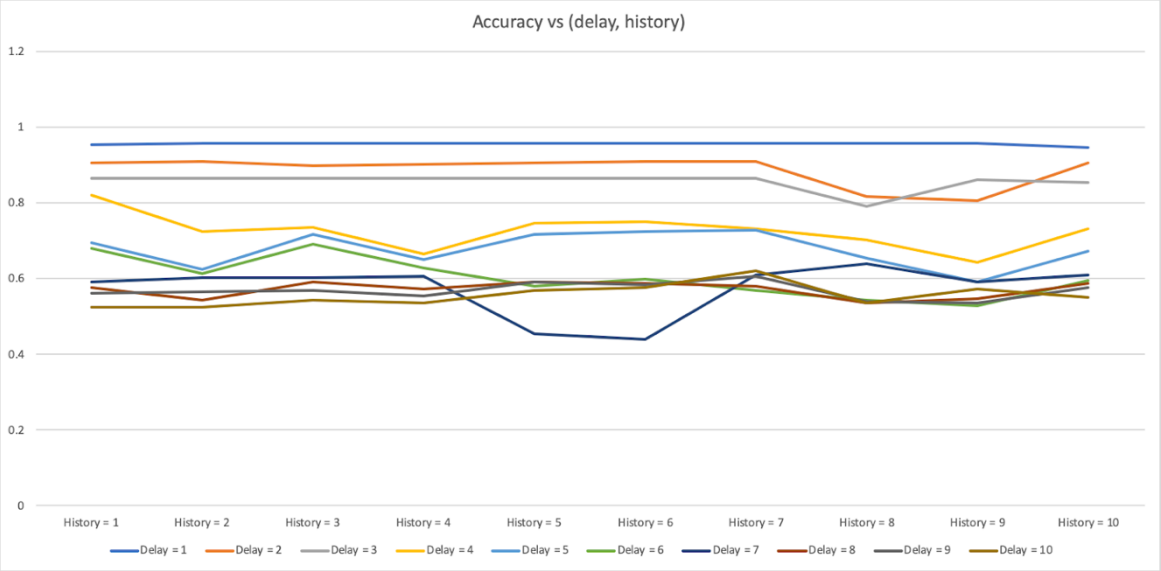


Figure 34: Chart showing the performance of fully connected neural network with varying history length

Table 7: Achieved accuracy of prediction for the current surgeme in presence of delays

Delay (in s)	Accuracy(in %)
0.000	100.00
0.167	95.28
0.333	90.50
0.500	71.00
0.667	83.00
0.833	68.50
1.000	65.43
1.167	62.40
1.333	57.37

We also present the achieved results in presence of varying delays using our fully connected neural network model. We note that the model can learn correctly as the case with no delays (delay = 0 s) gives 100% accuracy. This was not observed in LSTMs. Further, the accuracy of the model remains high (> 90%) with added delays. The accuracy of the model decreases as the delay increases from practical delay measurements (up to 500ms). Table 7 shows the prediction accuracy for the NN model for different delays.

3.2.3 Experiment 3: Metrics to Assess the Accuracy of High-Level Surgical Descriptors in
Transfer learning between surgical domains is paramount when dealing with austere conditions, since there is limited availability of surgical data for surgeries in remote environments. For this experimental setup two different conditions were explored: 1) Transfer learning between robots, for the same surgical task. 2) Transfer learning between surgical tasks.

3.2.3.1 Transfer learning between robots:
The goal of this phase was to transfer knowledge for surgeme recognition using a diverse set of robots: Simulated Taurus, Real SRI Taurus, Real YuMi and the da Vinci surgical robot. We initially trained our previous framework using data from a simulated SRI Taurus and tested on a real Taurus robot, a real ABB YuMi and the da Vinci surgical robot. The surgeme classification accuracy obtained for both YuMi and da Vinci was below 30%. This low recognition can be traced to the kinematic and workspace between the simulated Taurus and the real Yumi and da Vinci. To improve the recognition accuracy, we used the FFT algorithm as a method to map the kinematic data from different robots to the same space. The FFT algorithm can map a time series to a frequency domain. FFT outputs a set of frequency component bins that can represent spatial or geometric features of the signal. Therefore, our current framework replaces the feature reduction done with PCA with a feature extraction using a histogram obtained from the FFT frequency bins.

Figure 35 shows the developed framework. First, we reduced the dataset to the common kinematic features in all the robots: grippers' position, orientation, and state (open or close). The positional features were measured in the cartesian coordinate system (x,y,z), while the orientation was represented by the angles roll, pitch and yaw. This gives a total of 14 distinct features for each arm (7 each). A single 560-dimensional vector per surgeme (40 × 7 × 2) was created by concatenating 14 features for every frame. Then, we mapped these features to a common space using FFT. Further, a machine learning classifier was used to recognize the surgemes. Using a robot-agnostic set of features allows us to leverage information coming from robots of multiple domains.

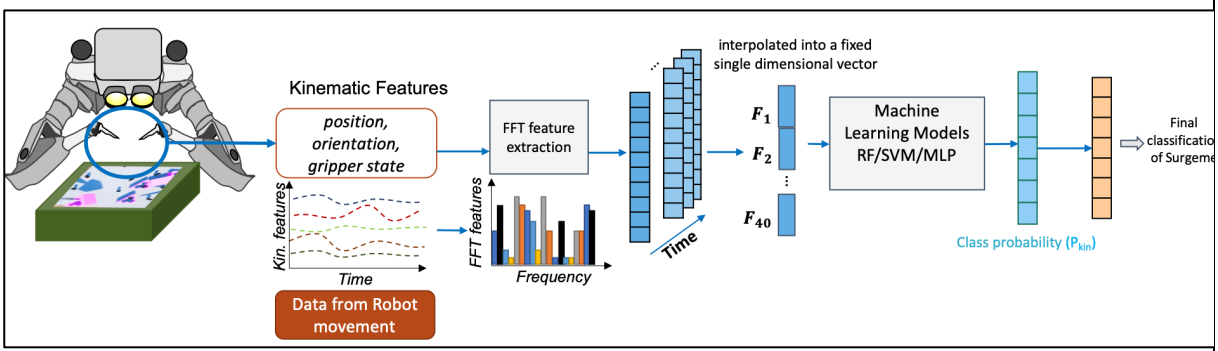


Figure 35. Architecture overview for Surgeme Recognition

For the transfer learning scenario, we started by training exclusively with simulation data and testing in the real robot using the architecture described in Figure 36. Then, we increasingly added data from the real scenario to the training set to simulate the effects of limited availability of the real data. We measured the presence of real data in the training model as a ratio between the real and simulated data. When the ratio value is zero, all the data comes from simulation. When the ratio value is 1, the data had a 50%-50% (50/50 = 1) distribution for real-simulated data. Figure 2 shows the classification accuracy of the models for all the real robots when they are trained using real data (orange line) against the performance when the training data is slowly added to the simulation data (blue line).

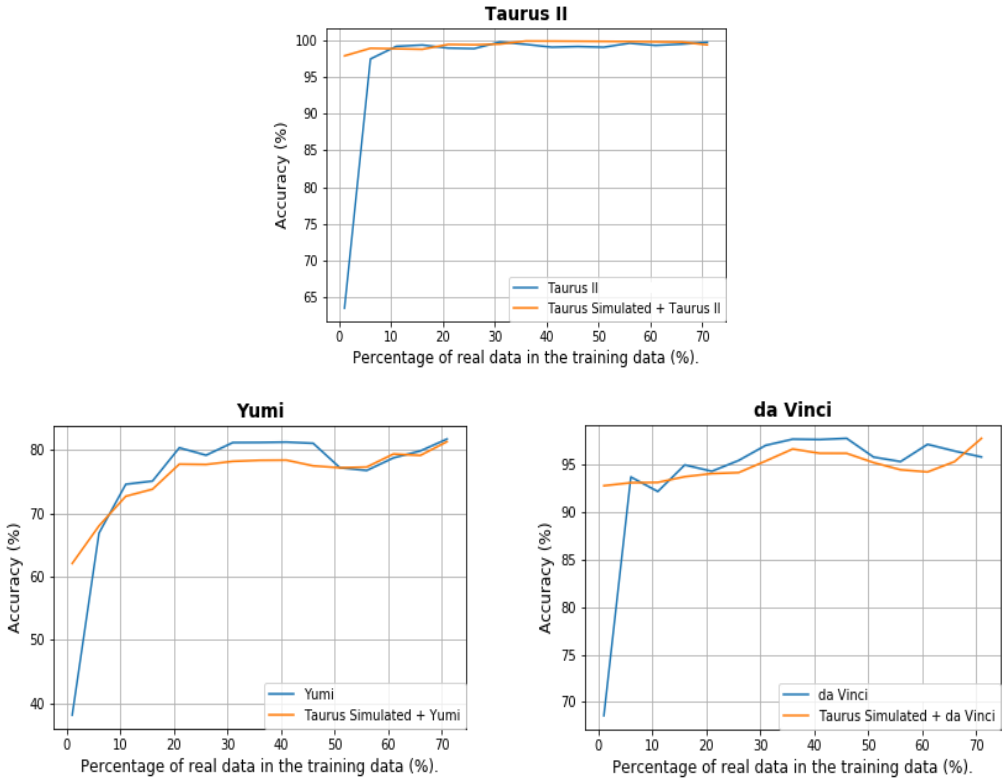


Figure 36. Performance comparison of training with the real data (no-transfer, shown in orange) vs training with only a percentage of the real data combined with simulation data (transfer learning scenario, shown in blue). The results are shown for the three real robots in the DESK dataset: the Taurus II robot (top) using SVM, the YuMi robot (bottom-left) using RF and the dVRK robot (bottom-right) using RF.

The Taurus II and da Vinci robots showed a classification accuracy of 97.5% and 93% respectively, even when there were no real examples included in the training set (data ratio=0). Adding a small number of real examples effectively improved the surgeme recognition accuracy, as shown in Figure 36. When the ratio of real to simulated data was 15%-85% (ratio=0.18), the classification accuracies went up to 99.7% for the Taurus II and 95.4% for the da Vinci. Therefore, these results show that surgeme classification on real environments can be achieved using a very small amount percentage of real data. The YuMi robot showed a slower convergence, with a ratio of 22%-78% of real to simulated data producing an accuracy of 81%. This accuracy discrepancy is likely due to the YuMi motions. The YuMi robot does not have three degrees of freedom at the gripper, making the orientation changes more abrupt. Thus, the teleoperators would choose a convenient orientation and default to translation

motions. In contrast, the gripper's position and orientation changed constantly for the da Vinci, the Taurus II and the simulated Taurus. This inconsistency in the teleoperation resulted in very different FFT features for the YuMi. Thus, it needed more of its own data during training to produce an accuracy over 80%.

Both da Vinci and Taurus II show a faster convergence to a classification accuracy (≥95%), needing little to no data in the transfer setup. The FFT features allow to describe the spatial properties (shapes) of the surgemes, while retaining scale and translation invariance. This allows to describe the same trajectory or gripper aperture in robots that have different workspaces, showing significant improvement over the previous approach.

3.2.3.2 Transfer learning between tasks:

We used a pre-trained network for classifying the surgemes in a peg transfer task to use as initial weights for the network described in previous section. We use the classifier model of [11] use for surgeme identification in the peg and pole task. The network structures for the two tasks are explained in Figure 5.

The peg transfer task and the debridement task involve similar surgemes such as approach skin or approach peg and align and grasp, however, the kinematic sequences may be drastically different. For example, the peg-and-pole task requires the peg manipulation using both robotic arms one after the other. However, in the debridement task, one arm only acts as the blade and the other arms holds the object for the entire duration of debridement. To demonstrate that the two sequences are different, we use dynamic time warping (DTW) distance. Table 8 shows the average DTW distance between any two sequence for the two surgical tasks.

TABLE 8: Average Dynamic Time Warp distances between the sequences of the two surgical task.

	Peg Transfer	Debridement
Peg Transfer	96.25	112.69
Debridement	112.69	106.24

First the results without transfer learning are presented, followed by the results from the different transfer learning setups. The proposed framework in Fig 37 is evaluated on a dataset of 88 surgeme sequences from 3 subjects with 12 trials each. The dataset is separated with a test-train split of 70% and 30% respectively. The surgeme recognition results are presented in Table 9.

TABLE 9: Surgeme classification accuracy of the baseline with non-transfer learning.

Accuracy	S1	S2	S3	S4	S5	S6	S7	**Average**
Testing	97	48	76	84	83	85	87	**83.7**
Training	97	59	83	93	88	88	89	**90.4**

As the proposed network few parameters to learn, 88 sequences are sufficient to achieve good classification. However, transfer learning achieves a similar performance with even lesser training sequences as shown in the following section.

A. Transfer learning results

First, a direct transfer learning approach is adopted. The network is trained with data from peg-transfer task. The peg transfer dataset was collected using 5 subjects with 12 trails per subject resulting in a total of 185 sequences. After a 70%–30% train-test split and training the network with the resulting 130 sequences, an average accuracy of 77% was obtained. This network was re-trained on the debridement data with the same split as the no-transfer scenario. The resulting surgeme classification accuracy is shown in Table 10.

TABLE 10: Surgeme classification accuracy on the test dataset for the baseline (no-transfer learning) and transfer learning methods.

Accuracy	S1	S2	S3	S4	S5	S6	S7	**Mean**
Baseline	97	48	76	84	83	85	87	**83.7**
Transfer Learning	94	63	74	86	84	79	91	**85.8**

When the entire dataset is used to train the proposed model, the transfer learning setup performed marginally better than the no-transfer setup. However, the transfer learning setup performed significantly better than the no-transfer setup as shown in the following section.

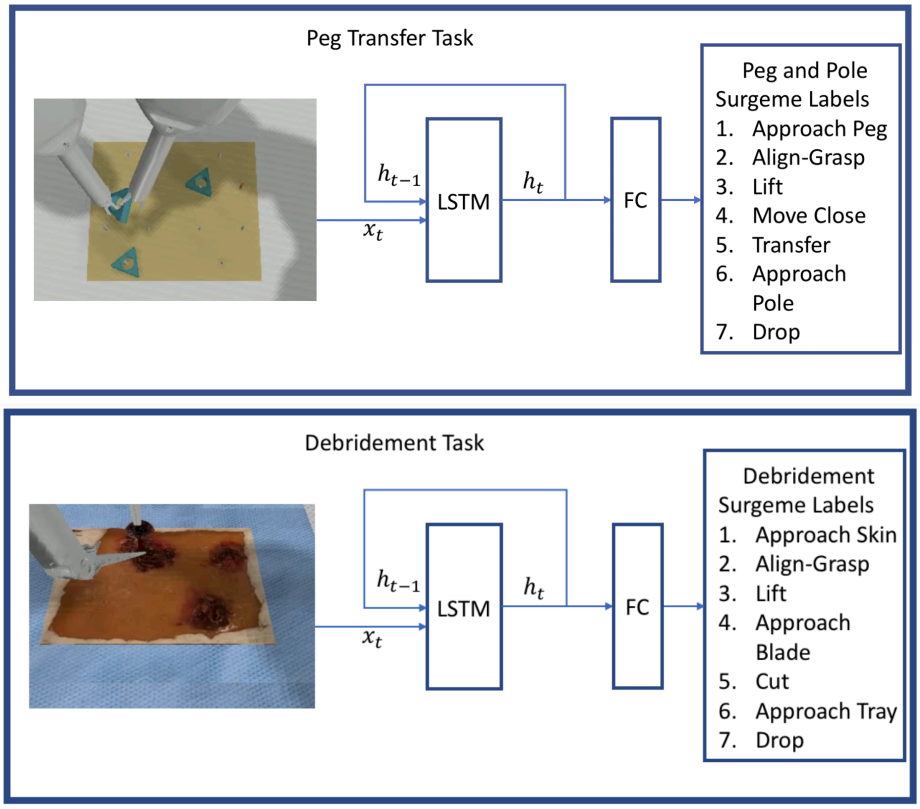


Fig. 37: The fundamental structure of the networks is kept same to allows for transfer of the knowledge between the two classifiers. First the network is trained on the Peg and Pole task where both the arms are used extensively. Pre-trained network is then used to transfer to the new debridement task. Since the number of classes are the same, we do not modify the structure of the last layer.

B. Transfer Learning Analysis

The get more insight into the transfer learning setup, three different aspects of the method are analyzed in the following sub-sections, 1) Impact of Amount of Training Data: The debridement dataset has 88 sequences of the task out of which 61 sequences were separated as training set. The remaining 17 sequences are used as the test set. To study the impact of the amount of training data a fraction of the 61 sequences is used for training and tested with the same 17 sequences. Different fractions starting from 10% to 70% (61 sequences) in steps of 10% were used. For the transfer learning, the network was retrained on this training set starting from the peg-transfer weights.

The impact of having more training data in the target domain is presented in Figure 38. The comparison is against the baseline of no-transfer learning. As shown, even with as little as 8 training sequences the classification accuracy of the transfer learning setup is 76% which is 22.6% better than the baseline. This suggests that the amount of training data required in the new surgical domain is reduced using a transfer learning setup.

2) Use of each layer in classification: For object recognition networks, transfer learning is usually achieved by re-training the last few layers. This is because the object specific discriminating features are learnt in the last few layers, whereas the initial layers produce low level image features such as edges, shapes, or textures. To analyze the proposed network similarly, the effect of retraining the different layers is evaluated. Our model comprises of an LSTM layer which can infer using the hidden states based on the history. Following this is a series of fully connected dense layers which use the outputs of LSTM layer to generate features for classification.

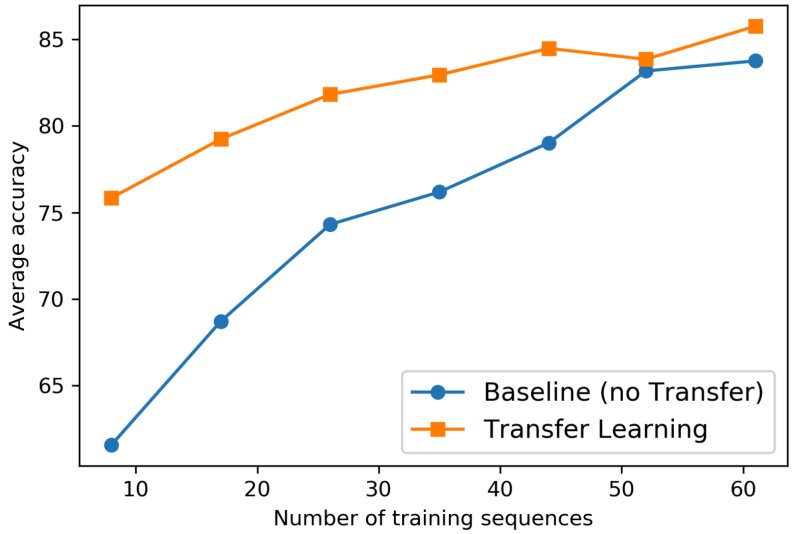


Fig. 38: Impact of increasing size of target dataset. The slow increase in accuracy of the transfer learning suggests that the network carries significant momentum from the weights trained using the peg transfer task.

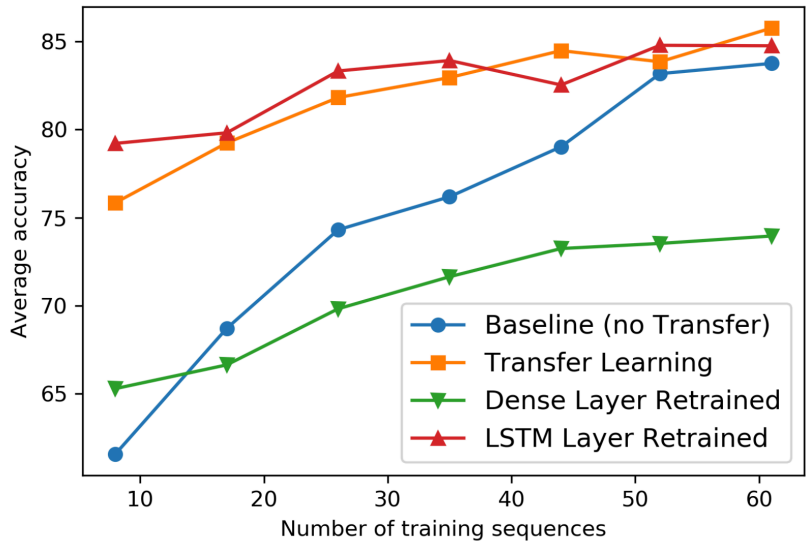


Fig. 39: Impact of re-training specific layers for transfer learning. Re-training dense layer only performs worse than the baseline, which suggests that most of the surgeme specific knowledge is capture by the LSTM layer.

Figure 39 presents the impact of re-training the different layers of the classification network. We observe that the retraining only LSTM layer achieves the same effect as retraining the entire network. This implies that the LSTM layer captures the domain-specific discriminating features required for surgeme classification. Further, retraining only the dense layers performs worse than the baseline. This signifies that re-training the LSTM layer is essential for transfer learning across sequential data such as surgemes.

3) Learning Rate of the transfer learning: All the networks are trained using stochastic gradient descent optimizer with a constant learning rate of 0.005. The transfer learning allows the model to train faster as compared to the baseline without transfer learning. To establish this fact, we compared the classification accuracy after each training step. We present the result for the networks are trained with 61 training sequences.

Figure 40 shows the effect of transfer learning on the convergence rates. The networks initialized with pre-trained weights already obtain an initial push to move towards the optima resulting in a faster convergence rate. The analysis of convergence rates shows that the transfer learning setup can not only provide advantages when there is little data in the target domain but also when the training time is limited. This is specially useful in cases where an emergency situation may allow only few hours for training the robot before deployment.

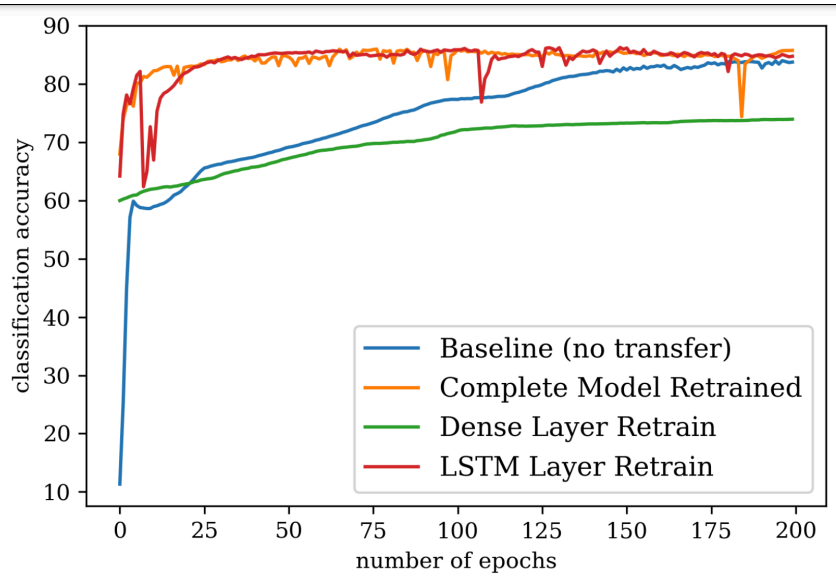


Fig. 40: Impact of transfer learning on the convergence of the neural network. The transfer learning setup converges after about 50 training steps as compared to the non-transfer learning setup which requires about 3 times more training steps.

In austere environments, such as battlefields, even performing the debridement task using a tele-surgical robotic system can be lifesaving in the absence of a medic. To eliminate severe impact of limited or no connectivity on the implementation of tele-surgery robotic systems execution using surgemes is considered. In this paper we considered developing a classifier for surgeme recognition for semiautonomous performance of the debridement task. We developed a simulator where a Taurus robot performs debridement task. Kinematic data for debridement task was collected and used to train an LSTM classifier. Such a classifier is then used to infer the current surgeme being performed. We showed that proposed LSTM framework can classify surgemes with an accuracy of 84.4%.

To reduce the data requirements and allow faster deployement, transfer learning is used. Data collected from peg transfer task was used to pre-train the network for transfer learning. By transferring the weights from surgeme recognition on the peg transfer task, the performance of the classifier increases by 10% when only 50% of the training data is available. It was shown that the non-transfer learning setup requires 50% more data to achieve the performance of the transfer learning setup. Ablation studies were conducted to understand the role of each layer in the transfer learning task. It was found that it is necessary to retrain the LSTM layer as the temporal structure of the kinematic sequences vary between the peg transfer task and the debridement task. Further, we found that the transfer learning models converges to the solution faster than the non-transfer learning method. We conclude that the proposed transfer learning allows the surgical system to be easily transferable to other surgical tasks.

3.2.4 Experiment 4: Metrics to Assess Connectivity Dropout and Recovery:

3.2.4.1 Peg Transfer Scenario

Task Setup: We set up a peg transfer task based on the DESK dataset that was previously developed [1]. We modified the environment to be monochrome (red) to simulate a real surgical scene. The triangular objects and poles were partially submerged on artificial blood to simulate the challenges

during surgery. In particular, the artificial blood adds scattering, and reflections to the vision system, as shown in Figure 41.

Results: The goal of the proposed framework is to mitigate the effect of delays over remote teleoperation. First, this section describes how each module on the framework was accessed. Then it outlines the task performance using different types of delays.



Figure 41. surgemes peg transfer.

To access the efficacy of the developed framework, we evaluated the system under three delay configurations: 1) No delay, 2) One second of delay, and 3) Five seconds of delay. Previous research has shown that the chosen delays severely hinder surgical teleoperation, making it effectively unusable for the surgeon.
Thus, we surgeme classification accuracy, completion time, and task success rate to measure the effects of delays on the system.

First, we measured the recognition accuracy for the surgemes performed by the
the remote robot during execution. In a successful trial with no mistakes, the robot should recognize the seven peg-transfer surgemes to execute. The results remote recognition accuracy using different

delays are shown in Figure 1. These results indicate that the recognition performance is not affected by delays. In particular, the no-delay setup obtained the same recognition accuracy as the 5 second-delay setup at 96%.

We also evaluated the performance of the system to complete the peg transfer task. The task completion rate is defined as the number of successfully completed trials by the number of trials. The completion rate at different delays is presented in Figure 42.

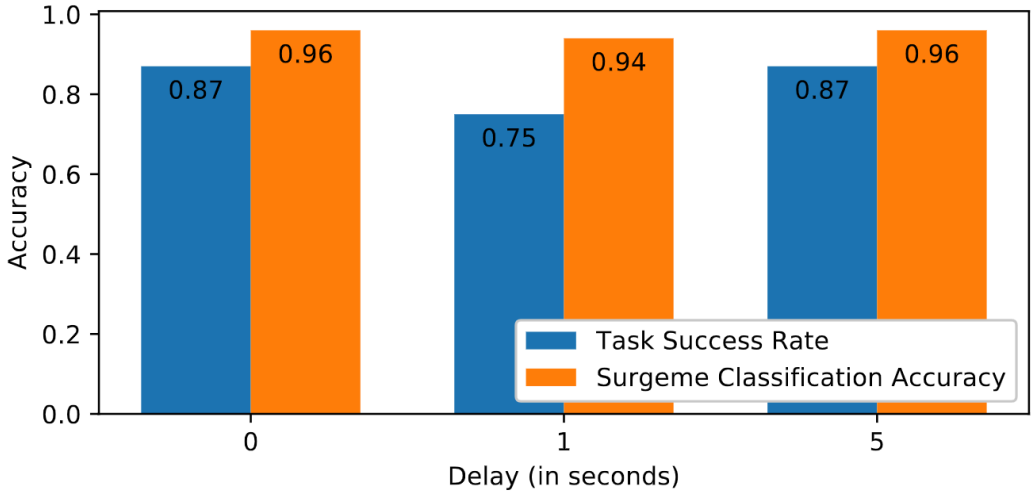


Figure 42: Surgeme detection accuracy and task completion rate under delays. Since surgemes are recognized from the operators simulated robot state, the recognition accuracy does not suffer degradation

We also compared the task completion time in the presence of delays. We defined the user completion time as the duration of a user trial in simulation. The trial starts when the user moves the robot for the first time and ends when the last peg transfer step is completed. Then, we measured the task completion time for the robot. The robot completion time is defined as the elapsed duration between the robot finishing execution of the last surgeme and the robot beginning the execution of the first surgeme. The task completion times at different delays are presented in Figure 43.

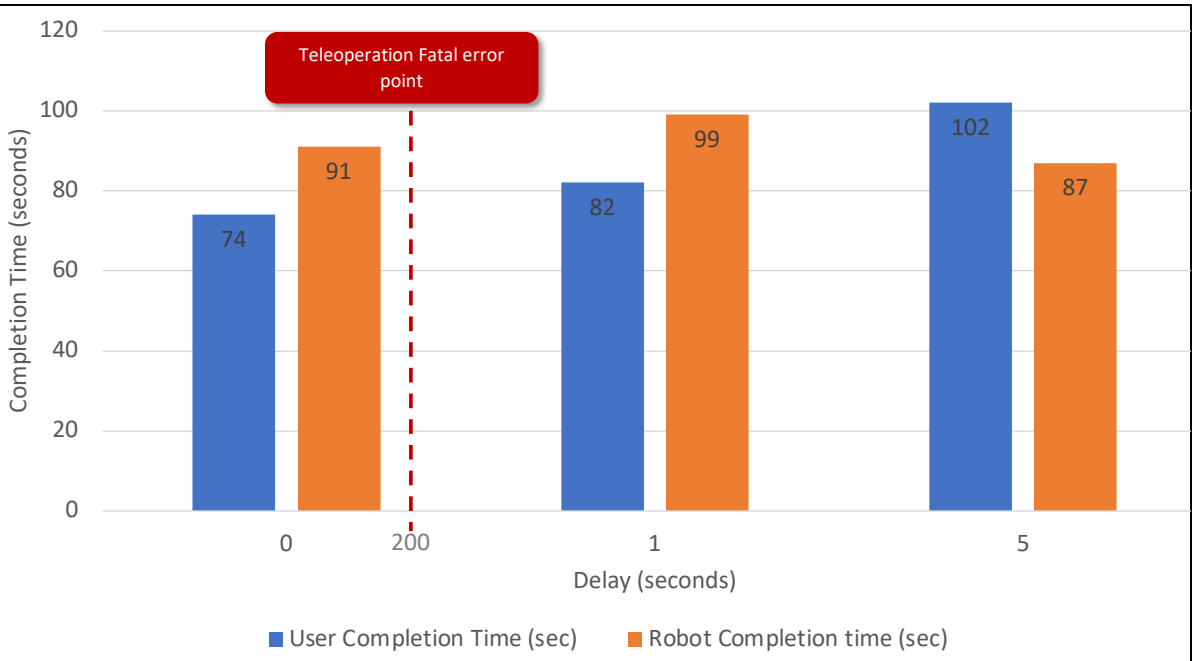


Figure 43: Completion time under delays. Because of a simulator-based interface, the operator can continue completing the task without increasing the completion time.

We analyzed the transmission delays in the surgeme transfer in absence of artificial delays. For the surgeme transfer from the simulator to the robot, the average delay was 0.057 seconds and a jitter of 0.048 seconds. Additionally, we observed that the simulator takes approximately 200 ms to update and the surgeme messages are generated at a frequency of 5 Hz. The resultant bandwidth requirement for sending surgemes is only 1 KBps.

Finally, we evaluated the packet drop rate of the feedback module running on the UDP protocol for different camera frame rates (see Figure 44). We noted that although the delays were 4 times lesser for a frame rate of 20 fps compared to the frame rate of 5 fps, the packet drop rate is 25 times higher. For 30 fps camera video rate, without the presence of any artificial network delays, the natural delay of our system is bounded by 0.36 seconds with 99% probability. In contrast, the commercial live streaming services provide the lowest delay guarantee of 8 seconds. We also compared the bandwidth requirement for the feedback module. A color image from a camera is a tensor of size 640x480x3 bytes for 8-bit encoding. At 5 fps, the camera generates a stream of about ~6 MBps. However, our proposed framework only generates 10 KBps, resulting in a total bandwidth requirement of 11 KBps.

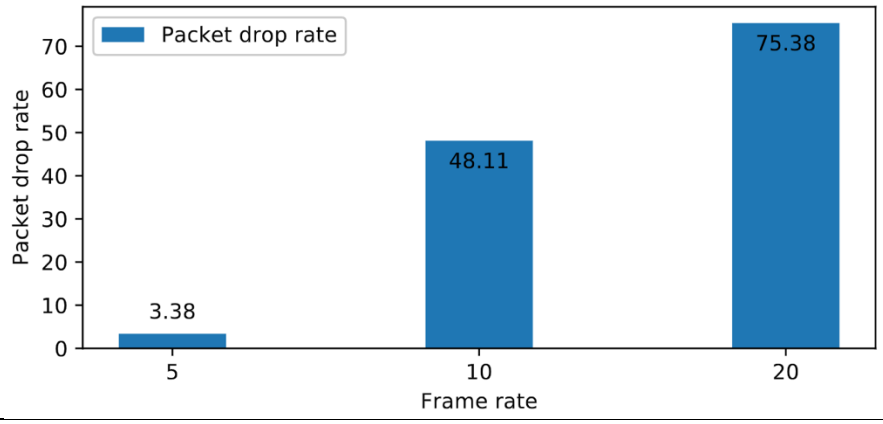


Figure 44: Packet drop rate for different camera frame rates.

3.2.4.2 Debridement Scenario

During debridement where each piece of simulated skin had three patches of burned skin. The goal of debridement is to remove the damaged regions. The skin and tissue was recreated with elastic properties to reproduce the challenges that the deformable tissue introduces during surgery. A total of 120 trials was collected from three subjects. Each subject performed 8 trials for each delay type, completing 40 trials at the end of the session. Figure 45 shows the debridement surgemes and Figure 46 shows the developed framework.

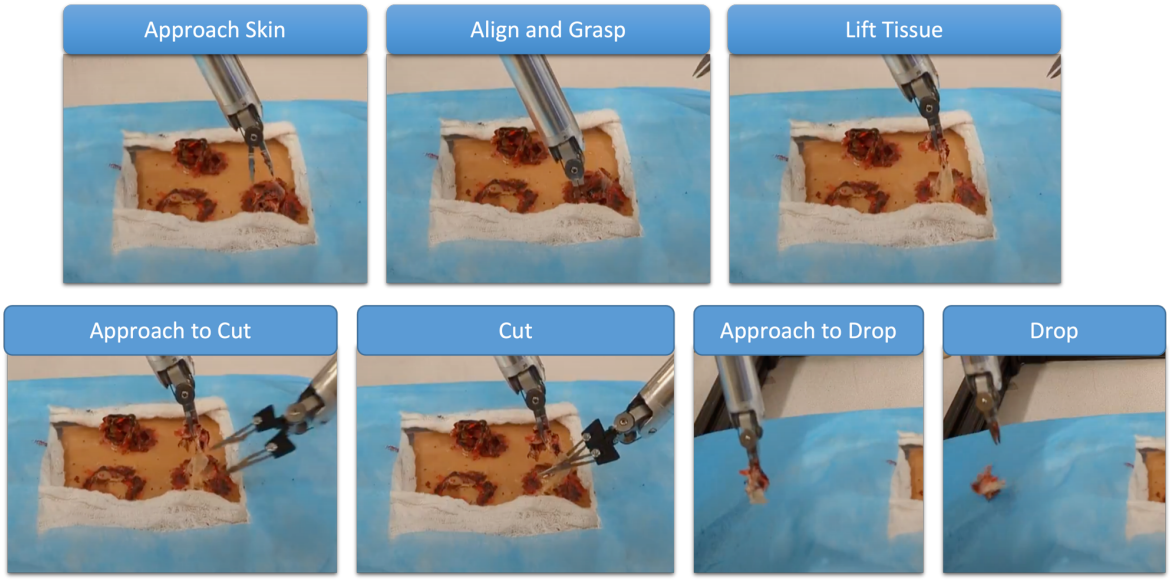


Figure 45. Surgemes to complete a trial.

To assess the efficacy of the developed framework, we evaluated the system under five delay configurations: 1) No delay, 2) 250ms of delay 3) 500ms of delay 2) One second of delay, and 3) Five seconds of delay. Previous research has shown that the chosen delays severely hinder surgical tele-operation, making it effectively unusable for the surgeon [2-6]. Thus, we measured the surgeme classification accuracy, the completion time, and task success rate to evaluate the effects of delays on the developed system. Figure X. shows a snapshot of the framework working in real time.

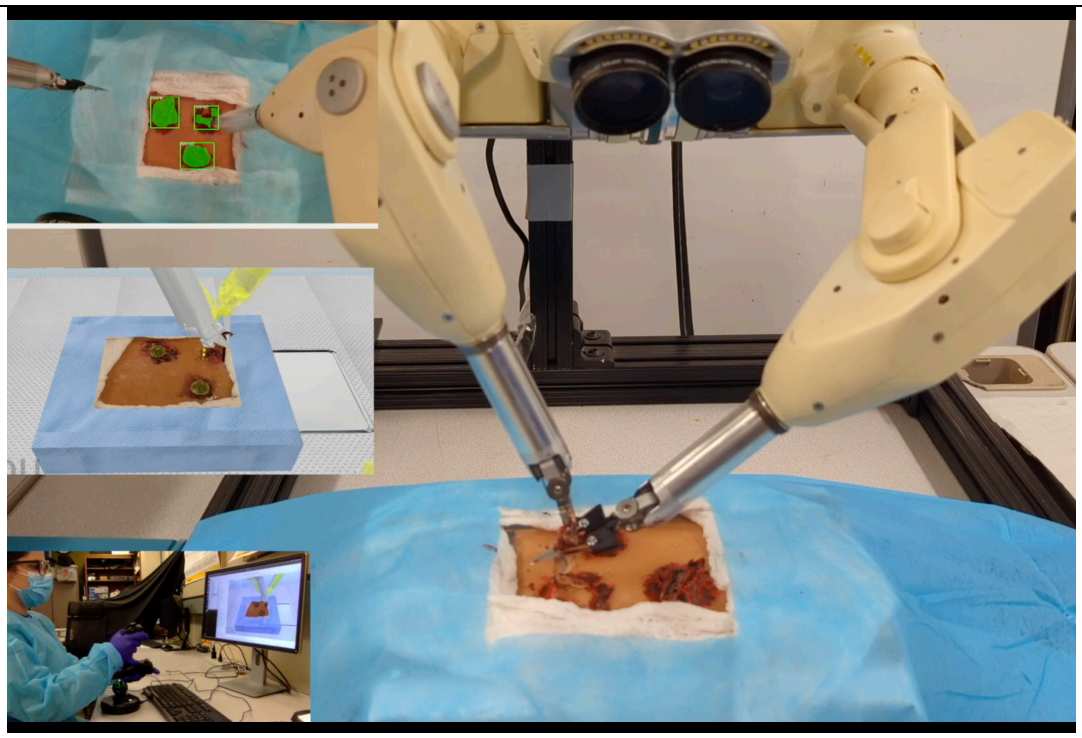


Figure 46. DESERT Framework for debridement

First, we measured the recognition accuracy for the surgemes performed by the remote robot during execution. In a successful trial with no mistakes, the robot should recognize the seven debridement surgemes for a successful trial. The results for remote recognition accuracy using different delays are shown in Figure 47 next to the frame-wise surgeme recognition accuracy of on the simulator side. These results show that the surgical step recognition accuracy is maintained over 90% until delays of 1s. In particular, the surgeme prediction with no delays achieved a classification accuracy of 98%. With the increase of delays, the user modifies it's execution behavior in the simulator. This changes the recognition accuracy to 88% when the delay is 5 seconds. Figure 47 shows a softer drop in the remote surgeme recognition, when compared to the frame-wise recognition in the simulator. This result indicates that a drop in frame-wise accuracy during simulation can be mitigated in the remote side, because the robot waits until there is enough confidence (threshold > 0.7) in the predicted surgeme to execute it.

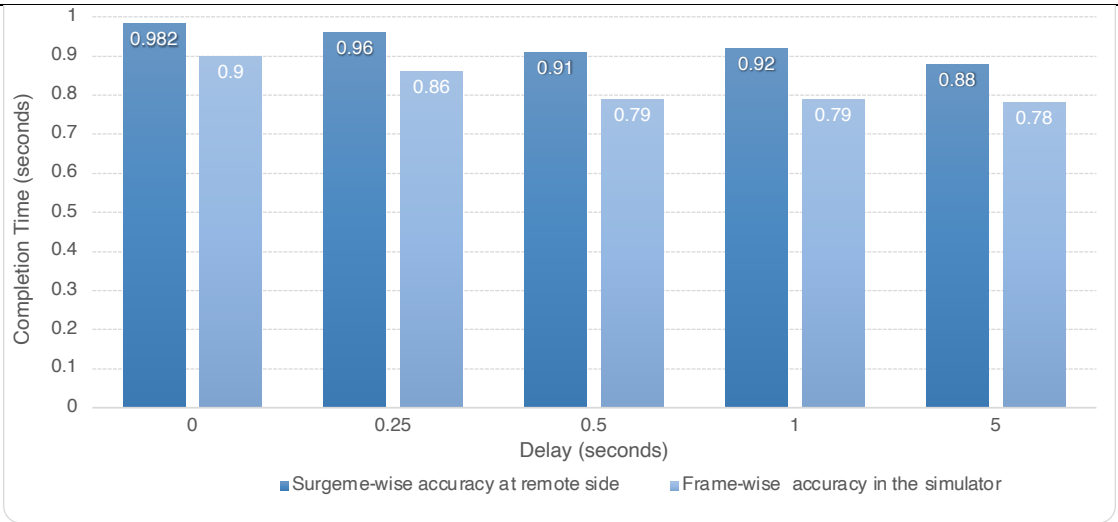


Figure 47. Surgeme recognition accuracy

Table 11. Surgeme recognition accuracy using partial history.

Surgeme	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**Avg**
Accuracy at threshold >0.7	1	0.8	1	0.58	0.85	1	1	-
% of surgeme history	0.04	0.55	0.13	0.05	0.03	0.17	0.33	0.12

In addition, an experiment was performed to understand the amount of surgeme that the network uses to achieve the required classification confidence (threshold > 0.7). Table 11 shows the surgeme-wise accuracy achieved at 0.7 confidence and the percentage of the surgeme history that the network used. A weighted mean of the percentage of surgeme history used by the network was calculated, indicating that the network needs only an average 12% of the total sugeme history is needed to produce an accuracy of 98% on the remote side. This means the system can reduce 88% of the elapsed surgeme recognition time through early prediction.

In addition, we compared the task completion time in the presence of delays. We defined the user completion time as the duration of a user trial in simulation. The trial starts when the user moves the robot for the first time and ends when the last peg transfer is completed. Then, we measured the end-to-end completion time. The later completion time is defined as the elapsed duration between the user moving the robot in the simulator for the first time and the remote robot finishing the execution of the last surgeme. The task duration results are presented in Figure 48. We also obtained the completion time for direct teleoperation using the same robot under no delays.

We performed and unpaired t-test between the delay types and the baseline. There was no significant difference between the teleoperation baseline and our system end-to-end completion time performance for 0 and 250ms. At 250ms of delays, traditional teleoperation presents disruptions to the user(cite). These results show that our system may mitigate the effect of delays such as 250ms.

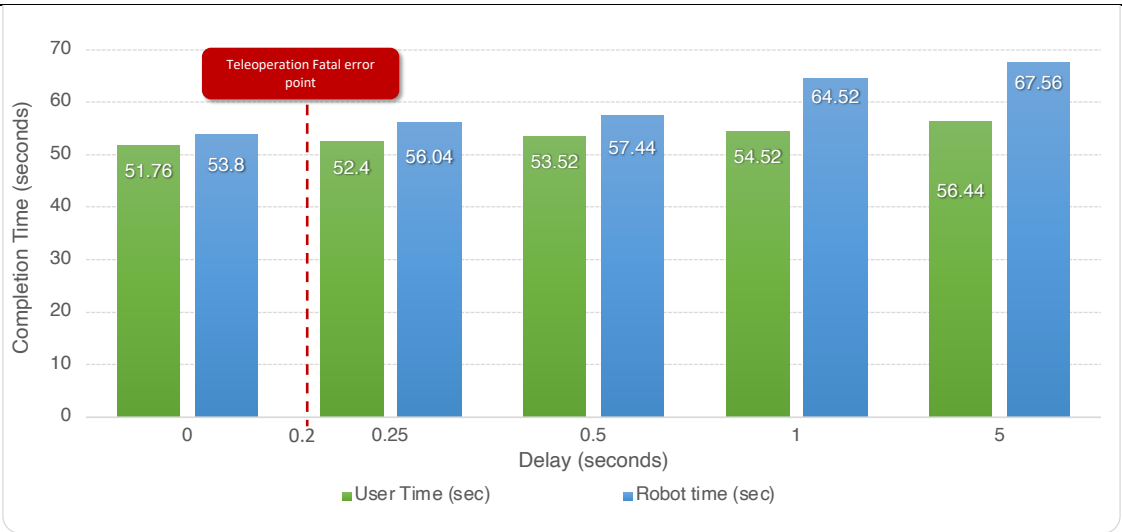


Figure 48. Completion Time under delays

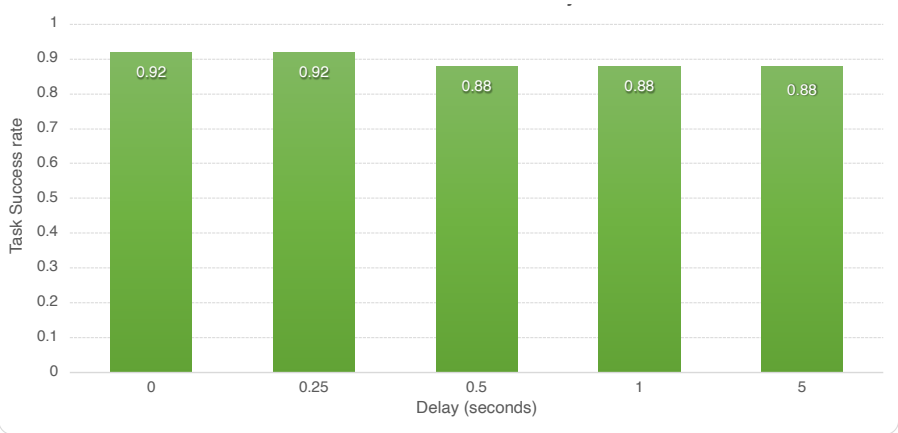


Figure 49. Task success rate under delays.

We also evaluated the performance of DESSERTS to complete the peg transfer task. The task completion rate is defined as the number of successfully completed trials by the number of trials. The completion rate at different delays is presented in Figure 49. The performance was maintained at 88% from 0.5s to 5s and for the no-delay and 0.25s delay setup the performance was 92%, We performed a paired t-test between the different delay setups, that showed there was no significant difference between any of the delay conditions. Thus, the performance of the proposed system is robust to delays as high as 5 seconds, which is a delay condition that state of the art teleoperation cannot withstand.

3.2.5: Real-World Application: Fail-Safe and Hybrid Human-Robot Team Recovery (key personnel Kirkpatrick leads)

The TeleMentored Ultrasound Supported Medical Interventions (TMUSMI) Research Group has long focused its efforts on supporting and empowering point of care providers to perform out of scope and to deliver life saving interventions (LSIs). Providing the earliest hemorrhage control is now recognized as a shared responsibility of all members of society, including both the lay public and professionals, consistent with the Stop the Bleed campaign. However, providing early hemorrhage control in a hostile environment, such as the scene of a mass shooting, is extremely challenging. In such settings, the first access to a bleeding victim may be robotic, a non-medical first responder or a first responder that the skill set required is out of their scope of practice. Over the course of this project we conducted a campaign of research examining different paradigms to improve and empower hemorrhage control in austere and extreme settings. Different arms or paradigms of this campaign included robotic application, telementoring and video modeling. The results of each of these areas can be found in the publications list below.

Robotic Hemorrhage Control
The TMUSMI group partnered with the Arapanoe County Sheriff's Office and the Denver South Metro Fire Rescue Service to initiate the worlds first studies examining the use of Bomb Robot's the place wound clamps on exsanguination simulators(1, 2). The investigations showed that a variety of Bomb robots, from heavy duty "workhorse" robots to light weight "surveillance" robots were capable of wound clamp application. Further, not unexpectant, the robot operators became more proficient with increased practice with wound clamp application. TMUSMI recommends that all bomb robots in North America should be capable of providing hemorrhage control in a hot zone if required.

Just-in-Time Telementored Hemorrhage Control
TMUSMI has partnered with both the Israeli defence for and the Canadian Forces Medical services to examine the attributes and challenges of using simple informatics to guide less experienced or even procedure naïve medics to perform various resuscitation techniques ranging from those they have been trained on to completely out of scope. High level lessons learned from these ongoing projects include the fact that with cooperation and mutual understanding medics seem to be able to perform much better than expected. However, when there is unfamiliarity and the remote expert "mentor" is not in tune to the point of care providers needs and work culture, then remote mentoring can be counter-productive(3-5).

Video-Modelled (VM) Procedural Guidance
Video-modelling is a paradigm that while ubiquitous in "normal" life, has be unexplored to potentially empower life saving procedures in austere locations. This is also a paradigm that may lend itself to Space Exploration when the spacecraft venture beyond low Earth's orbit and real time communications become impractical. TMUSMI thus has initiated inaugural studies comparing VM versus telementored simulated life saving procedures conducted by Canadian Search and Rescue Technicians. While the analysis is ongoing, both techniques have merits that may be complimentary(6). In summary however, while multiple supporting technologies exist, the human factors aspects of delivering any lifesaving interventions in austere environments are largely still unknown and in the opinion of TMUSMI require urgent detailed studies to bring these techniques to operational reality.

References

[1] Madapana, N., Rahman, M.M., Sanchez-Tamayo, N., Balakuntala, M.V., Gonzalez, G., Bindu, J.P., Venkatesh, L.V., Zhang, X., Noguera, J.B., Low, T. and Voyles, R.M., 2019, November. Desk: A robotic activity dataset for dexterous surgical skills transfer to medical robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 6928-6934). IEEE.

[2] T. Kim, P. Zimmerman, M. Wade, and C. Weiss, "The effect of delayed visual feedback on telerobotic surgery," Surgical Endoscopy and Other Interventional Techniques, vol. 19, no. 5, pp. 683–686, 2005.

[3] M. D. FABRLZIO, B. R. Lee, D. Y. Chan, D. Stoianovici, T. W. Jarrett, C. Yang, and L. R. Kavoussi, "Effect of time delay on surgical performance during telesurgical manipulation," Journal of endourology, vol. 14, no. 2, pp. 133–138, 2000.

[4] M. Anvari, T. Broderick, H. Stein, T. Chapman, M. Ghodoussi, D. W. Birch, C. Mckinley, P. Trudeau, S. Dutta, and C. H. Goldsmith, "The impact of latency on surgical precision and task completion during robotic-assisted remote telepresence surgery," Computer Aided Surgery, vol. 10, no. 2, pp. 93–99, 2005.

[5] S. Xu, M. Perez, K. Yang, C. Perrenot, J. Felblinger, and J. Hubert, "Determination of the latency effects on surgical performance and the acceptable latency levels in telesurgery using the dv-trainer® simulator," Surgical endoscopy, vol. 28, no. 9, pp. 2569–2576, 2014.

[6] A. Kumcu, L. Vermeulen, S. A. Elprama, P. Duysburgh, L. Platisa, ˇ Y. Van Nieuwenhove, N. Van De Winkel, A. Jacobs, J. Van Looy, and W. Philips, "Effect of video lag on laparoscopic surgery: correlation between performance and usability at low latencies," The International Journal of Medical Robotics and Computer Assisted Surgery, vol. 13, no. 2, p. e1758, 2017.

[7]Goodfellow, I., Bengio, Y. and Courville, A., 2016. *Deep learning*. MIT press.

[8] Williams, C.K. and Rasmussen, C.E., 2006. *Gaussian processes for machine learning* (Vol. 2, No. 3, p. 4). Cambridge, MA: MIT press.

[8] Duvenaud, D., 2014. *Automatic model construction with Gaussian processes* (Doctoral dissertation, University of Cambridge).

[9] M. Hwang, D. Seita, B. Thananjeyan, J. Ichnowski, S. Paradis, D. Fer, T. Low, and K. Goldberg, "Applying Depth-Sensing to Automated Surgical Manipulation with a da Vinci Robot," in International Symposium on Medical Robotics (ISMR), 2020.

[10] Reiley, C.E. and Hager, G.D., 2009, September. Task versus subtask surgical skill evaluation of robotic minimally invasive surgery. In *International conference on medical image computing and computer-assisted intervention* (pp. 435-442). Springer, Berlin, Heidelberg.

[11] Targ, S., Almeida, D. and Lyman, K., 2016. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*.

What opportunities for training and professional development has the project provided?
If the project was not intended to provide training and professional development opportunities or there is nothing significant to report during this reporting period, state "Nothing to Report."

Describe opportunities for training and professional development provided to anyone who worked on the project or anyone who was involved in the activities supported by the project. "Training" activities are those in which individuals with advanced professional skills and experience assist others in attaining greater proficiency. Training activities may include, for example, courses or one-on-one work with a mentor. "Professional development" activities result in increased knowledge or skill in one's area of expertise and may include workshops, conferences, seminars, study groups, and individual study. Include participation in conferences, workshops, and seminars not listed under major activities.

Training:

The PI Wachs and co-PI Voyles, Aggarwal, Voyles and Xue are mentoring 4 graduate students: (1) through a research assistantship, computer science, polytechnic, and engineering students are working on all the problems related to computer vision, robotics and machine learning under the mentorship of co-PI Yexiang Xie; (2) through a research assistantship, industrial engineering PhD student Glebys Gonzales is working in the problems related to robot teaching and experimental design, under the mentorship of PI Wachs.

How were the results disseminated to communities of interest?
If there is nothing significant to report during this reporting period, state "Nothing to Report."

Describe how the results were disseminated to communities of interest. Include any outreach activities that were undertaken to reach members of communities who are not usually aware of these project activities, for the purpose of enhancing public understanding and increasing interest in learning and careers in science, technology, and the humanities.

Madapana, N., Rahman, M. M., Sanchez-Tamayo, N., Balakuntala, M. V., Gonzalez, G., Bindu, Venkatesh, L., Zhang X., Noguera, J., Low, T., Voyles, R. M., Wachs, J. (2019, November). Desk: A robotic activity dataset for dexterous surgical skills transfer to medical robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 6928-6934). IEEE. – Oral Presentation

Rahman, M.M., Balakuntala, M.V., Gonzalez, G., Agarwal, M., Kaur, U., Venkatesh, V.L., Sanchez-Tamayo, N., Xue, Y., Voyles, R.M., Aggarwal, V. and Wachs, J., 2021. SARTRES: a semi-autonomous robot teleoperation environment for surgery. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, *9*(4), pp.376-383.

Gonzalez, G.T., Kaur, U., Rahman, M., Venkatesh, V., Sanchez, N., Hager, G., Xue, Y., Voyles, R. and Wachs, J., 2021. From the Dexterous Surgical Skill to the Battlefield—A Robotics Exploratory Study. *Military medicine*, *186*(Supplement_1), pp.288-294.

Gonzalez, G., Agarwal, M., Balakuntala, M.V., Rahman, M.M., Kaur, U., Voyles, R.M., Aggarwal, V., Xue, Y. and Wachs, J., 2021, May. Deserts: Delay-tolerant semi-autonomous robot teleoperation for surgery. In *2021 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 12693-12700). IEEE. Oral Presentation

Agarwal, M., Gonzalez, G., Balakuntala, M.V., Rahman, M.M., Aggarwal, V., Voyles, R.M., Xue, Y. and Wachs, J., 2021, August. Dexterous Skill Transfer between Surgical Procedures for Teleoperated Robotic Surgery. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)* (pp. 1236-1242). IEEE. Interactive session presentation

Rahman, M.M., Sanchez-Tamayo, N., Gonzalez, G., Agarwal, M., Aggarwal, V., Voyles, R.M., Xue, Y. and Wachs, J., 2019, October. Transferring dexterous surgical skill knowledge between robots for semi-autonomous teleoperation. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)* (pp. 1-6). IEEE. Oral Presentation

Rahman, M.M., Voyles, R.M., Wachs, J. and Xue, Y., 2021, August. Sequential prediction with logic constraints for surgical robotic activity recognition. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)* (pp. 468-475). IEEE.

Balakuntala, M.V., Kaur, U., Ma, X., Wachs, J. and Voyles, R.M., 2021. Learning Multimodal Contact-Rich Skills from Demonstrations Without Reward Engineering. *arXiv preprint arXiv:2103.01296*.

Malpani, Anand, S. Swaroop Vedula, Henry C. Lin, Gregory D. Hager, and Russell H. Taylor. "Effect of real-time virtual reality-based teaching cues on learning needle passing for robot-assisted minimally invasive surgery: a randomized controlled trial." International journal of computer assisted radiology and surgery (2020).

Balakuntala, M.V., Venkatesh, V.L., Bindu, J.P., Voyles, R.M. and Wachs, J., 2019, October. Extending policy from one-shot learning through coaching. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)* (pp. 1-7). IEEE.

Glebys Gonzalez, Masudur Rahman, Mridul Agarwal, Mythra Balakuntala, Vishnu Venkatesh, Vaneet Aggarwal, Yexiang Xue, Richard Voyles, Gregory Hager, Andrew W Kirkpatrick, Steve Overholser, Juan Wachs. “ASTRO: A Semi-Autonomous Telemedicine Robot for Operative Surgery”. Military Health System Research Symposium (Abstract accepted at MHSRS-2020). Poster Presentation

Glebys Gonzalez, Mythra Balakuntala, Mridul Agarwal, Masudur Rahman, Richard Voyles,Vaneet Aggarwal, Yexiang Xue and Juan Wachs. “A semi-autonomous robotic framework for remote surgery under delays”. Military Health System Research Symposium (Abstract accepted at MHSRS-2021).

Nothing to Report.

4. **IMPACT:** Describe distinctive contributions, major accomplishments, innovations, successes, or any change in practice or behavior that has come about as a result of the project relative to:

What was the impact on the development of the principal discipline(s) of the project?
If there is nothing significant to report during this reporting period, state “Nothing to Report.”

Describe how findings, results, techniques that were developed or extended, or other products from the project made an impact or are likely to make an impact on the base of knowledge, theory, and research in the principal disciplinary field(s) of the project. Summarize using language that an intelligent lay audience can understand (Scientific American style).

In dangerous austere environments, teleoperated robots can provide medical assistance for critical injuries, without risking the lives of medics in the field. However, network constraints and connectivity issues have restricted the use of remote teleoperation. We address the limited bandwidth and latency problem and their impact in teleoperation by introducing a novel closed-loop tele-surgical framework. This framework works by abstracting low-level image data into high-level command data. First, the objects in the scene are recognized, which are tracked, and in turn, displayed at the simulator. At the operator side, only high-level instructions (surgemes) are recognized and sent to the remote robot. Representing surgical maneuvers with high-level abstraction, in the form of surgemes, not only reduce the bandwidth requirements, but also allow the robot to operate semi-autonomously for short segments during the procedure, thereby making the whole system robust to delays. The validation of this framework includes the execution of a peg transfer task, a fundamental laparoscopic surgery skill, and a surgical debridement. We evaluated the overall and individual modules' performance on this task for a variety of delay settings. Notably, the framework achieves a task success rate of 88% with delays up to 5 seconds. These results show the effectiveness of the framework even in extreme adversarial settings. In the future, this framework can be extended to incorporate increasingly complex surgical procedures such as debridement, suturing, and incisions to be more useful to the field of telesurgery.

In addition, to further eliminate severe impact of limited or no connectivity on tele-surgical framework, we considered developing a classifier for surgeme recognition for semiautonomous performance of the debridement task. We developed a simulator where a Taurus robot performs debridement task. Kinematic data for debridement task was collected and used to train an LSTM classifier. Such a classifier is then used to infer the current surgeme being performed. We showed that proposed LSTM framework can classify surgemes with an accuracy of 84.4%.

What was the impact on other disciplines?
If there is nothing significant to report during this reporting period, state "Nothing to Report."

Describe how the findings, results, or techniques that were developed or improved, or other products from the project made an impact or are likely to make an impact on other disciplines.

An initial formative experiment conducted during the telerobotic surgical skill training for a variety of surgical tasks (i.e. peg transfer, debridement and cricothyrotomy), provided some initial understanding about the role of high level and low level instructions affect performance. Improved understanding of the factors affecting design and use of explicit commands as well as the physical and cognitive requirements for this interaction will be crucial to introduce physical interaction with devices in teleoperation. It is expected that the use of high-level interfaces and the surgeme learned through deep learning frameworks will increase the understanding about the different uses of autonomous control in the operating setting, with extensions to other high-risk/high-stakes scenarios.

What was the impact on technology transfer?

If there is nothing significant to report during this reporting period, state "Nothing to Report."

Describe ways in which the project made an impact, or is likely to make an impact, on commercial technology or public use, including:

- *transfer of results to entities in government or industry;*
- *instances where the research has led to the initiation of a start-up company; or*
- *adoption of new practices.*

Nothing to report.

What was the impact on society beyond science and technology?
If there is nothing significant to report during this reporting period, state "Nothing to Report."

Describe how results from the project made an impact, or are likely to make an impact, beyond the bounds of science, engineering, and the academic world on areas such as:

- *improving public knowledge, attitudes, skills, and abilities;*
- *changing behavior, practices, decision making, policies (including regulatory policies), or social actions; or*
- *improving social, economic, civic, or environmental conditions.*

The new fundamental scientific theories and algorithms developed through the three years of funding through this FORwARD Award will have applications to the broad field of communications and machine learning, but specially will be suitable to the robotic surgical domain in the austere setting. At the fundamental level, our work will address challenges such as "what is the tradeoff between specificity and generality of data for maximum knowledge transfer between domains", or "how to measure diversity of actions/objects among a dataset" (e.g. entropy would be a good starting point), "how to compress the information so the semantic content is not affected", and "how to deal with imperfect and randomly delayed information to the robot". Our research strategy will also suggest a potential plan to implement these principles into a semi-autonomous robot capable of performing common surgical tasks associated with the battlefield. The direct outcome of this will be an improvement in public knowledge about semi-autonomous robotics.

5. **CHANGES/PROBLEMS:** The Project Director/Principal Investigator (PD/PI) is reminded that the recipient organization is required to obtain prior written approval from the awarding agency Grants Officer whenever there are significant changes in the project or its direction. If not previously reported in writing, provide the following additional information or state, "Nothing to Report," if applicable:

Changes in approach and reasons for change
Describe any changes in approach during the reporting period and reasons for these changes. Remember that significant changes in objectives and scope require prior approval of the agency.

No changes

Actual or anticipated problems or delays and actions or plans to resolve them
Describe problems or delays encountered during the reporting period and actions or plans to resolve them.

No delays.

Changes that had a significant impact on expenditures
Describe changes during the reporting period that may have had a significant impact on expenditures, for example, delays in hiring staff or favorable developments that enable meeting objectives at less cost than anticipated.

No changes.

Significant changes in use or care of human subjects, vertebrate animals, biohazards, and/or select agents
Describe significant deviations, unexpected outcomes, or changes in approved protocols for the use or care of human subjects, vertebrate animals, biohazards, and/or select agents during the reporting period. If required, were these changes approved by the applicable institution committee (or equivalent) and reported to the agency? Also specify the applicable Institutional Review Board/Institutional Animal Care and Use Committee approval dates.

Significant changes in use or care of human subjects

No changes.

Significant changes in use or care of vertebrate animals.

No changes

Significant changes in use of biohazards and/or select agents

No changes.

6. **PRODUCTS:** List any products resulting from the project during the reporting period. If there is nothing to report under a particular item, state "Nothing to Report."

- **Publications, conference papers, and presentations**
  Report only the major publication(s) resulting from the work under this award.

  Journal publications. *List peer-reviewed articles or papers appearing in scientific, technical, or professional journals. Identify for each publication: Author(s); title; journal; volume: year; page numbers; status of publication (published; accepted, awaiting publication; submitted, under review; other); acknowledgement of federal support (yes/no).*

Rahman, M.M., Balakuntala, M.V., Gonzalez, G., Agarwal, M., Kaur, U., Venkatesh, V.L., Sanchez-Tamayo, N., Xue, Y., Voyles, R.M., Aggarwal, V. and Wachs, J., 2021. SARTRES: a semi-autonomous robot teleoperation environment for surgery. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, *9*(4), pp.376-383.

Gonzalez, G.T., Kaur, U., Rahman, M., Venkatesh, V., Sanchez, N., Hager, G., Xue, Y., Voyles, R. and Wachs, J., 2021. From the Dexterous Surgical Skill to the Battlefield—A Robotics Exploratory Study. *Military medicine*, *186*(Supplement_1), pp.288-294.

Malpani, Anand, S. Swaroop Vedula, Henry C. Lin, Gregory D. Hager, and Russell H. Taylor. "Effect of real-time virtual reality-based teaching cues on learning needle passing for robot-assisted minimally invasive surgery: a randomized controlled trial." International journal of computer assisted radiology and surgery (2020).

Kirkpatrick AW, Mckee JL, Netzer I, Mckee IA, McBeth P, Wachs JP, Ball CG, Glassberg E. A Randomized Trial of Mentored vs Nonmentored Military Medics Compared in the Application of a Wound Clamp Without Prior Training: When to Shut Up and Just Watch! Mil Med. 2020 Jan 7;185(Suppl 1):67-72. doi: 10.1093/milmed/usz251. PMID: 32074324.

McKee IA, McKee JL, Knudsen BE, Shelton R, LaPorta T, Wachs J, Kirkpatrick AW. A “human-proof pointy-end”: a robotically applied hemostatic clamp for care-under-fire. Can J Surg. 2019 Dec 1;62(6):E13-E15. doi: 10.1503/cjs.002619. PMID: 31782650; PMCID: PMC6877384.

Jessica Mckee JL, McBeth P, Wachs J, Hamilton D, Ball C, Gillman L, Kirkpatrick A. Creating a physical telementored ultrasound supported medical interventions (TMUSMI) “box.” Can J Surg, 2019 Vol. 62 (3 Suppl 2); S18-19

Books or other non-periodical, one-time publications. *Report any book, monograph, dissertation, abstract, or the like published as or in a separate publication, rather than a periodical or series. Include any significant publication in the proceedings of a one-time conference or in the report of a one-time study, commission, or the like. Identify for each one-time publication: Author(s); title; editor; title of collection, if applicable; bibliographic information; year; type of publication (e.g., book, thesis or dissertation); status of publication (published; accepted, awaiting publication; submitted, under review; other); acknowledgement of federal support (yes/no).*

No changes.

Other publications, conference papers, and presentations. *Identify any other publications, conference papers and/or presentations not reported above. Specify the status of the publication as noted above. List presentations made during the last year (international, national, local societies, military meetings, etc.). Use an asterisk (*) if presentation produced a manuscript.*

1 Madapana, N., Rahman, M. M., Sanchez-Tamayo, N., Balakuntala, M. V., Gonzalez, G., Bindu, Venkatesh, L., Zhang X., Noguera, J., Low, T., Voyles, R. M., Wachs, J. (2019, November). Desk: A robotic activity dataset for dexterous surgical skills transfer to medical robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 6928-6934). IEEE.

Rahman, M.M., Balakuntala, M.V., Gonzalez, G., Agarwal, M., Kaur, U., Venkatesh, V.L., Sanchez-Tamayo, N., Xue, Y., Voyles, R.M., Aggarwal, V. and Wachs, J., 2021. SARTRES: a semi-autonomous robot teleoperation environment for surgery. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, *9*(4), pp.376-383.

Gonzalez, G.T., Kaur, U., Rahman, M., Venkatesh, V., Sanchez, N., Hager, G., Xue, Y., Voyles, R. and Wachs, J., 2021. From the Dexterous Surgical Skill to the Battlefield—A Robotics Exploratory Study. *Military medicine*, *186*(Supplement_1), pp.288-294.

Gonzalez, G., Agarwal, M., Balakuntala, M.V., Rahman, M.M., Kaur, U., Voyles, R.M., Aggarwal, V., Xue, Y. and Wachs, J., 2021, May. Deserts: Delay-tolerant semi-autonomous robot teleoperation for surgery. In *2021 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 12693-12700). IEEE.

Agarwal, M., Gonzalez, G., Balakuntala, M.V., Rahman, M.M., Aggarwal, V., Voyles, R.M., Xue, Y. and Wachs, J., 2021, August. Dexterous Skill Transfer between Surgical Procedures for Teleoperated Robotic Surgery. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)* (pp. 1236-1242). IEEE.

Rahman, M.M., Sanchez-Tamayo, N., Gonzalez, G., Agarwal, M., Aggarwal, V., Voyles, R.M., Xue, Y. and Wachs, J., 2019, October. Transferring dexterous surgical skill knowledge between robots for semi-autonomous teleoperation. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)* (pp. 1-6). IEEE.

Rahman, M.M., Voyles, R.M., Wachs, J. and Xue, Y., 2021, August. Sequential prediction with logic constraints for surgical robotic activity recognition. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)* (pp. 468-475). IEEE.

Balakuntala, M.V., Kaur, U., Ma, X., Wachs, J. and Voyles, R.M., 2021. Learning Multimodal Contact-Rich Skills from Demonstrations Without Reward Engineering. *arXiv preprint arXiv:2103.01296*.

Malpani, Anand, S. Swaroop Vedula, Henry C. Lin, Gregory D. Hager, and Russell H. Taylor. "Effect of real-time virtual reality-based teaching cues on learning needle passing for robot-assisted minimally invasive surgery: a randomized controlled trial." International journal of computer assisted radiology and surgery (2020).

Balakuntala, M.V., Venkatesh, V.L., Bindu, J.P., Voyles, R.M. and Wachs, J., 2019, October. Extending policy from one-shot learning through coaching. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)* (pp. 1-7). IEEE.

Glebys Gonzalez, Masudur Rahman, Mridul Agarwal, Mythra Balakuntala, Vishnu Venkatesh, Vaneet Aggarwal, Yexiang Xue, Richard Voyles, Gregory Hager, Andrew W Kirkpatrick, Steve Overholser, Juan Wachs. “ASTRO: A Semi-Autonomous Telemedicine Robot for Operative Surgery”. Military Health System Research Symposium (Abstract accepted at MHSRS-2020).

Glebys Gonzalez, Mythra Balakuntala, Mridul Agarwal, Masudur Rahman, Richard Voyles,Vaneet Aggarwal, Yexiang Xue and Juan Wachs. “A semi-autonomous robotic framework for remote surgery under delays”. Military Health System Research Symposium (Abstract accepted at MHSRS-2021).

Kirkpatrick AW, McKee J, McKee I, Knudsen B, Shelton R, Netzer I, LaPorta A, Ball C, McBeth P, Wachs J. Hemorrhage Control in the “Killing Zone”: A Randomized Controlled Trial of

Humanoid versus Robotic First Response in Care Under Fire Situations, presented by McKee JL, and discussed by Kirkpatrick AW at The Sharpest Knife Competition, Surgeons Day, University of Calgary, Calgary, Alberta, June 14 2019.

McKee JL, McBeth PB, McKee IA, Gillman L, Wachs JP, Barrett R, Leeper WR, Kirkpatrick AW. Remote just-in-time telementored tourniquet application: A double-factorial randomized controlled trial examining two tactical tourniquets and scripted and unscripted mentor-mentee guidance – a pilot study. Presented in poster format by McKee JL at the Military Health System Research Symposium. August 22 2018, Kissimmee, Florida.

Kirkpatrick AW, McKee JL, McBeth P, Hamilton D, Wachs J, Ball CG, Gillman L. Creating a physical tele-mentored ultrasound supported medical interventions (TMUSMI) Box. Presented in Oral format by AW Kirkpatrick at the 2019 Annual Scientific Meeting of the Trauma Association of Canada, Calgary, Alberta March 1 2019

Kirkpatrick AW, McKee I, Knudsen B, Clark C, Shelton R, LaPorta AJ, Wachs J, McKee JL. Hot Zone Hemorrhage Control: Technical Notes on Robotically Applied Hemostatic Clamping for Care-Under-Fire: Presented in oral format by Kirkpatrick AW at the Military Health System Research Symposium, Kissimmee, Florida, Aug 21 2019. MHSRS-19-01038.

Kirkpatrick AW, McKee I, Knudsen B, Clark C, Shelton R, LaPorta AJ, Wachs J, McKee JL. What do you do when a Tension Pneumothorax Needs Treatment and the Last Man Standing is Wearing a Bomb Disposal Suit? Remote Mentoring of In-transport Tube Thoracostomy by Procedure Naïve Personnel. Presented in oral format by McKee JL at the Military Health System Research Symposium, Kissimmee, Florida, Aug 19 2019. MHSRS-19-2525.

McKee JL, McBeth PB, JP Wachs, Hamilton DR, Kirkpatrick AW. Creating a Physical Tele-Mentored Ultrasound Supported Medical Interventions (TMUSMI) “Box”. Presented in Poster Format at Presented in poster format by AW Kirkpatrick at the Military Health System Research Symposium, Kissimmee, Florida, Aug 21 2019. MHSRS-19-02552.

Kirkpatrick AW, McKee IA, Shelton R, McBeth P, Gillman L, Barrett R, Wachs J, McKee JL. It’s okay to move your Lips when you Read: A Randomized Trial of low-Bandwidth Mentoring versus Reading the Manufacturer Instructions Regarding Tourniquet Application. Presented in Poster Format at Presented in oral format by JL McKee at the Military Health System Research Symposium, Kissimmee, Florida, Aug 21 2019. MHSRS-19-02535.

Mckee JL, McBeth P, Wachs J, Hamilton D, Ball CG, Gillman LM, Kirkpatrick AW. Building a "Real" Telemedicine Ultrasound Supported Medical Intervention (TMUSMI) Mousetrap: Creating a Practical Multi-Modality Telementoring Capacity for Far Forward Resuscitation. Presented in Poster Format at Presented in poster format by JL McKee at the Military Health System Research Symposium, Kissimmee, Florida, Aug 21 2019. MHSRS-19-02537.

Tomlinson C, Donley N, McKee JL, Kay M, Wachs J, Kirkpatrick AW, editors. Is it better to watch or listen before doing? A Randomized Trial of Video-Modelling versus Telementoring for In- and Out-of-Scope Prehospital Life-Saving Interventions Performed by Search and Rescue Medics. 2020 Military Health System Research Symposium; 2020 - Cancelled by COVID; Kissimme, Florida.

Kirkpatrick AW, McKee JL, Tomlinson C, Donley, Ball CG, Wachs J. Testing and Comparative Evaluation of Video-Modelling to Allow Out-of-Scope Life-Saving Medical Interventions Randomly Compared to Real-Time Telementoring. Presented virtually by AW Kirkpatrick at the 2021 Canadian Space Agency- National Research Council of Canada Industrial Research Assistance Program Colloquium “Healthcare without Boundaries”, June 2 2021.

- **Website(s) or other Internet site(s)**
  List the URL for any Internet site(s) that disseminates the results of the research activities. A short description of each site should be provided. It is not necessary to include the publications already specified above in this section.

Forward Website
Project dissemination website, summarizing the goals and accomplishments of the grant.
Desk Dataset:
Library of surgemes collected during a peg transfer procedure. The data was collected using four different robots: A da Vinci Research Kit, An SRI Taurus, a simulated SRI Taurus and an ABB YuMi robot.

Data collection Protocol:
Reference document that explains in detail the collection protocol for the Desk Dataset. The information includes the pegboard setup for every subject, direction of the transfer (left-to-right or right-to-left) and the camera configuration with respect to the robot.

Development Repository:
This repository contains all the surgeme execution development.

- **Technologies or techniques**
  Identify technologies or techniques that resulted from the research activities. In addition to a description of the technologies or techniques, describe how they will be shared

As a result of our research, we have developed and implemented an annotation the DESK (Dexterous Surgical Skill) dataset. It comprises a set of surgical robotic skills collected during a surgical training task using four robotic platforms (the Taurus II robot, Taurus II simulated robot, the YuMi robot and the da Vinci Research Kit dVRK) and two surgical tasks (laparoscopic peg-transfer , surgical debridement). This dataset was used to test the idea of transferring knowledge across different domains and different surgical tasks.

In addition we created a framework (DESERTS) that addresses the challenges of remote teleoperation by enabling the surgeon to perform procedures in a simulated environment, completely uninterrupted. High-level surgical maneuvers are inferred from the surgeon's work and sent to the robot. At the same time, the simulator displays the remote environment as an alpha-blended scene, allowing the surgeon to maintain situational awareness of the conditions at the remote site. The framework eliminates the need of low-level motion information and high-quality video data transfers.

- **Inventions, patent applications, and/or licenses**
  Identify inventions, patent applications with date, and/or licenses that have resulted from the research. State whether an application is provisional or non-provisional and indicate the application number. Submission of this information as part of an interim research performance progress report is not a substitute for any other invention reporting required under the terms and conditions of an award.

Nothing to report

- **Other Products**

DESK surgical skill database**:** **link** DESERTS framework for surgical teleoperation under delays: • Publication: 10.1109/ICRA48506.2021.9561399 • Development code: https://github.com/glebysg/surgeme_ros.git

7. PARTICIPANTS & OTHER COLLABORATING ORGANIZATIONS

What individuals have worked on the project?
Provide the following information for: (1) PDs/PIs; and (2) each person who has worked at least one person month per year on the project during the reporting period, regardless of the source of compensation (a person month equals approximately 160 hours of effort). If information is unchanged from a previous submission, provide the name only and indicate "no change."

Name:	*Juan P Wachs*
Project Role:	*Principal Investigator*
Researcher Identifier (e.g. ORCID ID): 0000-0002-6425-5745	
Nearest person month worked:	*1.12 month*
Contribution to Project:	*Supervising the overall performance of the project. Coordinated visits to IUSM. Working with Maria Eugenia in all the aspects of gesture recognition and one shot learning. Working with Aditya Shanghavi for the design of the large interaction table. Helping with the journal publication.*
Name:	*Richard Voyles*
Project Role:	*Co-Investigator*
Researcher Identifier (e.g. ORCID ID):	
Nearest person month worked:	*1.12 month*

Contribution to Project: *Actively participated in and advised research assistant Mythra Balakuntala in the research and development of the first prototype of the execution by Reinforcement Learning; in designing, conducting, and analyzing the results of user studies aimed at assessing the robotic system; in disseminating the project results in a journal paper.*
Name: *Vaneet Aggarwal* *Project Role:* *Co-Investigator* *Researcher Identifier (e.g. ORCID ID):* *Nearest person month worked:* *1.12 month* *Contribution to Project:* *Supervised and actively participated in research for theoretical limits for performance of reinforcement learning algorithms under delays. Guided development of experiments for performance of learning algorithms under delays. Supervised research for algorithms for detecting and predicting surgemes.*
Name: *Yexiang Xue* *Project Role:* *Co-Investigator* *Researcher Identifier (e.g. ORCID ID):* *Nearest person month worked:* *Contribution to Project:* *Actively participated in and advised research assistant Md Masudur Rahman and others in the research and development of the recognition module and the first prototype of the overall pipeline; in designing, conducting, and analyzing the results of the recognition module and the overall pipeline; in disseminating the project results in conference papers.*
Name: *Gregory Hager* *Project Role:* *Co-Investigator* *Researcher Identifier (e.g. ORCID ID):* *Nearest person month worked:* *Contribution to Project:* *Actively participated in and advised research assistant Yotam Barnoy in the development of automatic execution of surgical tasks in the da Vinci simulatpr*
Name: *Glebys Gonzalez* *Project Role:* *Research Assistant* *Researcher Identifier (e.g. ORCID ID): 0000-0002-0181-006* *Nearest person month worked:* *6 months* *Contribution to Project:* *Coordinating the development of the surgeme recognition and execution with all the Research Assistants. Actively participated in the development of the automatic execution system of the remote robot and the*

feedback simulated system. Actively contributed in the data collection of the Taurus Simulator and development of scene semantic methods with Vision.

Name: *Md Masudur Rahman*
Project Role: *Research Assistant*
Researcher Identifier (e.g. ORCID ID): 0000-0002-3633-0621
Nearest person month worked: *6 months*

Contribution to Project: *Developed the learning algorithms for surgeme classification on the live. the simulated system. In addition Masud Rahman actively participated in the development of the feedback system*

Name: *Mythra V. Balakuntala*
Project Role: *Research Assistant*
Researcher Identifier (e.g. ORCID ID): 0000-0003-2551-4780
Nearest person month worked: *6 months*

Contribution to Project: *Formulated the methods for the perception module of the robot. Designed and developed the object detection, tracking and instance segmentation models for scene understanding. Developed the blood peg transfer setup and the modifications to the perception system for the same. Formulated the theoretical aspects of the Reinforcement learning models for learning the surgemes. Developed the simulated environment to deploy RL algorithms to learn the surgemes and trained surgemes using different DRL models. Designed and developed the cricothyroidotomy setup. Responsible for creation of the dataset and surgeme modelling for the cricothyroidotomy task. Setup the feedback from the robot to the simulator to update the simulator. Contributed to the data collection for peg-transfer data collected in the Yumi simulator.*

Name: *Mridul Agarwal*
Project Role: *Research Assistant*
Researcher Identifier (e.g. ORCID ID)
Nearest person month worked: *6 months*

Contribution to Project: *Responsible for the development of the communication sytem and the surgeme prediction with delays. Actively contributed in the data collection of the Taurus Simulator and development of scene semantic methods with Vision. Actively participated in publications in robotic conferences.*

Has there been a change in the active other support of the PD/PI(s) or senior/key personnel since the last reporting period?
If there is nothing significant to report during this reporting period, state "Nothing to Report."

If the active support has changed for the PD/PI(s) or senior/key personnel, then describe what the change has been. Changes may occur, for example, if a previously active grant has closed and/or if a previously pending grant is now active. Annotate this information so it is clear what has changed from the previous submission. Submission of other support information is not necessary for pending changes or for changes in the level of effort for active support reported previously. The awarding agency may require prior written approval if a change in active other support significantly impacts the effort on the project that is the subject of the project report.

No changes.

What other organizations were involved as partners?
If there is nothing significant to report during this reporting period, state "Nothing to Report."

Describe partner organizations – academic institutions, other nonprofits, industrial or commercial firms, state or local governments, schools or school systems, or other organizations (foreign or domestic) – that were involved with the project. Partner organizations may have provided financial or in-kind support, supplied facilities or equipment, collaborated in the research, exchanged personnel, or otherwise contributed.
Provide the following information for each partnership:

Organization Name: Dr. Andrew Kirkpatrick, University of Calgary
Location of Organization: Calgary, Canada,
Partner's contribution to the project: Over the course of this project, the co-PIs conducted a campaign of research examining different paradigms to improve and empower hemorrhage control in austere and extreme settings. Different arms or paradigms of this campaign included robotic application, telementoring and video modeling.

8. SPECIAL REPORTING REQUIREMENTS

COLLABORATIVE AWARDS: For collaborative awards, independent reports are required from BOTH the Initiating PI and the Collaborating/Partnering PI. A duplicative report is acceptable; however, tasks shall be clearly marked with the responsible PI and research site. A report shall be submitted to **https://ers.amedd.army.mil** for each unique award.

QUAD CHARTS: If applicable, the Quad Chart (available on https://www.usamraa.army.mil) should be updated and submitted with attachments.

9. **APPENDICES:** Attach all appendices that contain information that supplements, clarifies or supports the text. Examples include original copies of journal articles, reprints of manuscripts and abstracts, a curriculum vitae, patent applications, study questionnaires, and surveys, etc.