ASAP: A Semi-Autonomous Precise System for Telesurgery During Communication Delays

Glebys Gonzalez, Mythra Balakuntala, Mridul Agarwal, Tomas Low, Bruce Knoth, Andrew W. Kirkpatrick, Jessica Mckee, Gregory Hager, *Fellow, IEEE*, Vaneet Aggarwal, Yexiang Xue, Richard Voyles, *Fellow, IEEE*, and Juan Wachs, *Senior Member, IEEE*

***Abstract*—In remote, rural, and disadvantaged areas, telesurgery can be severely hindered by limitations of communication infrastructure. In conventional telesurgery, delays as small as 300ms can produce fatal surgical errors. To mitigate the effect of communication delays during telesurgery, we introduce a semi-autonomous system that decouples the user interaction from the robot execution. This system uses a physics-based simulator where a surgeon can demonstrate individual surgical subtasks, with immediate graphical feedback. Each subtask is performed asynchronously, unaffected by communication latency, jitter, and packet loss. A surgical step recognition module extracts the intended actions from the observed surgeon-simulation interaction. The remote robot can perform each one of these actions autonomously. The action recognition system leveraged a transfer learning approach that minimized the data needed during training, and most of the learning is obtained from simulated data. We tested this system in two tasks: fluid-submerged peg transfer (resembling bleeding events) and surgical debridement. The system showed robustness to delays of up to 5 seconds, maintaining a performance rate of 87% for peg transfer and 88% for debridement. Also, the framework reduced the completion time under delays by 45% and 11% during peg transfer and debridement, respectively.**

***Index Terms*—Medical robotics, telesurgical robotics, human robot interaction, deep learning, transfer learning.**

Manuscript received 30 June 2022; revised 15 November 2022; accepted 13 January 2023. Date of publication 25 January 2023; date of current version 23 February 2023. This article was recommended for publication by Associate Editor A. Menciassi and Editor P. Dario upon evaluation of the reviewers' comments. This work was supported in part by NSF FMitF Program under Award CCF-1918327; in part by the Office of the Assistant Secretary of Defense for Health Affairs under Award W81XWH-18-1-0769; in part by NSF Center for Robots and Sensors for the Human WellBeing under Award CNS-143971; and in part by the CR-II Program under Award IIS-1850243. *(Corresponding author: Juan Wachs.)*

Glebys Gonzalez and Juan Wachs are with the School of Industrial Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: gonza337@purdue.edu; jpwachs@purdue.edu).

Mythra Balakuntala and Richard Voyles are with the School of Engineering Technology, Purdue University, West Lafayette, IN 47907 USA (e-mail: mbalakun@purdue.edu; rvoyles@purdue.edu).

Mridul Agarwal and Vaneet Aggarwal are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: agarw180@purdue.edu; vaneet@purdue.edu).

Tomas Low and Bruce Knoth are with the Robotic Systems Department, SRI Robotics Laboratory, Menlo Park, CA 94025 USA (e-mail: thomas.low@sri.com; bruce.knoth@sri.com).

Andrew W. Kirkpatrick and Jessica Mckee are with the Robotic Systems Department, Foothills Medical Centre, Calgary, AB T2N 2T9, Canada (e-mail: Andrew.Kirkpatrick@albertahealthservices.ca; jlb9@ualberta.ca).

Gregory Hager is with the Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: hager@cs.jhu.edu).

Yexiang Xue is with the Department of Computer Science, Purdue University, West Lafayette, IN 47907 USA (e-mail: yexiang@purdue.edu).

Digital Object Identifier 10.1109/TMRB.2023.3239674

I. Introduction

IN MILITARY medicine, the wounded must receive skilled surgical attention to address their injuries as soon as possible, to minimize blood loss and improve the likelihood of survival. The pioneering telesurgery work of SRI International in the late 1980s was focused on developing such capabilities. However, practical considerations, such as the lack of reliable communications bandwidth in austere environments, posed a significant challenge to the deployment of such systems. If these limitations could be overcome, the resulting system could improve medical care, reduce the risks to forward deployed medical personnel and allow greater numbers to be treated with the limited number of surgeons in theatre.

Previous works show that delays can have a fatal effect in teleoperated robotic surgery [1], [2], [3], [4], [5]. It has been shown that delays as small as 200 milliseconds can significantly affect surgical performance [5]. Furthermore, task performance degradation caused by delays can lead to increased mortality risk during surgery [1]. Even when the fastest networks are in place, unpredictable delays are almost unavoidable and common for long-distance telecommunication [6], [7]. In addition, the low-quality communication in austere areas leads to jitters and interruptions, directly affecting telesurgery. Therefore, to mitigate these problems, we introduce a ASAP (A Semi-Autonomous Precise) robotic framework. This frame- work allows the surgeon to perform the surgery without experiencing delays in visual feedback while the procedure is performed remotely in a semi-autonomous manner. Moreover, this framework aims to reduce the need to constantly query for the user's decision, which is necessary for remote operation in areas of no connectivity or space exploration.

This work tackles the communication challenges by decoupling the operator interface from robot manipulation. This is achieved using a virtual representation of the real surgical scenario, where the user can operate without experiencing communication interruptions or latency delays. The surgeon's actions are recognized and encoded into unit surgical routines (surgemes), which in turn are sent to the remote robot, where such surgemes are performed semi-autonomously. This architecture eliminates the need for a constant stream of information (visual and kinematic), making it possible to teleoperate through unreliable networks and greatly reducing the bandwidth necessary to perform such procedures. The framework comprises four modules: 1) Recognition, 2) Communication,

3) Scene Interpretation, and 4) Execution. The first module is recognition. This module takes the motions indicated by the physician in the simulator and recognizes the surgical action, which is then encoded into a surgeme. Then, the classified surgeme is sent through the communication module to the remote robot. At the remote robot side, a vision module is used to segment, recognize, and track the elements in the environment that are relevant to the surgery. The execution module relies on the environment information from the vision module and the recognized surgeme to execute it autonomously. This high-level information exchange significantly reduces the data needed for teleoperation and allows safe, semi-autonomous execution of surgical maneuvers. Finally, to increase the surgeon's situational awareness, the execution module sends back the environment objects' positions and the robot pose to the user, which in turn are used to update the simulator state.

We evaluated the framework's performance using two surgical tasks: 1) A peg transfer task, which includes grasping, dual- arm coordination, and obstacle avoidance. The peg transfer setup includes immersed objects in a bleeding simulator. The presence of bleeding introduces additional challenges to object recognition algorithms and motor control. 2) A surgical debridement, which is a dual-hand procedure that deals with soft tissue manipulation and arm coordination.

The performance of the recognition module was evaluated in terms of the percentage of surgeme recognition for two transfer learning settings: transfer learning between tasks and transfer learning between robots. During the evaluation, data from the target domain was gradually added to the training process to observe the effect on recognition accuracy. Then, the performance of the entire system was assessed in the presence of delays. We first studied the performance under no delay and then tested it under several increments up to 5s to simulate extreme communication challenges.

The contributions of this paper are:

- A framework for semi-autonomous operation that is robust to delays up to 5 seconds.
- A framework that reduces the information transferred during teleoperation by 99%.
- A developed model for data transfer between robots, that achieved an accuracy of 89% for surgeme recognition, using only 50% of the target domain data.
- A developed model for learning between tasks, which achieved an accuracy of 85% for surgeme recognition, and a superior convergence rate than the classification without transfer learning.
- Studied the effect of delays w.r.t teleoperation in two surgical scenarios, where the framework showed to mitigate the effect of delays by 45% (peg-transfer) and 11% (de-bridement).

II. Background

This section summarizes the main strategies in previous work that have been used to mitigate the effect of delays:

1) Information compression, which reduces the transmission time [8].
2) Incorporation of autonomy to reduce the communication frequency between the surgeon and the robot [9], [10].
3) Predictive displays [11], [12] to reduce the effect of delays on the user interface [13], [14], [15], [16], [17], [18].

The next subsections explain these strategies in more detail.

A. Information Compression

Researchers have developed data compression frameworks to address the problem of latency during teleoperation [8]. However, in many cases, these frameworks require specialized communication protocols and encoding algorithms that can efficiently process the information from robotic sensors (i.e., 3D point clouds, multiple view cameras, and haptic information) [19], [20].

Stokto et al. proposed a method for compressing 3D point cloud streams that were used to reconstruct a virtual scene. The user interacts in the reconstructed virtual environment, allowing them to explore the space freely. To stream the point cloud, the data is divided into small blocks and compressed using a lossless real-time compressor [11], [21]. Naceri et al. [22] more recently addressed the challenge of streaming 3D information by reducing the live feed to a single camera on the robot's gripper while the rest of the environment is reproduced virtually. For the 3D stream, the framework compresses the color and the point-cloud frames separately, and the color- depth correspondence is found on the user side. Schimpe et al. [20] addresses the challenge of streaming simultaneous videos from multiple cameras for mobile robot teleoperation. The authors propose an adaptive video streaming that automatically allocates and reconfigures the bitrate according to the network latency [20]. Finally, Doniec et al. [19] addressed the issue of communication robustness for applications where information integrity is critical (i.e., submarine operation). This work uses an encoding scheme of two layers to prevent any errors during the live feed, working up to a latency of 100ms. Though research in data compression mitigates the effects of latency and bandwidth limitations, it cannot reliably deal with long network interruptions [8]. Thus, autonomy has been incorporated as part of teleoperation frameworks since it can reduce the communication frequency between the user and the robot. The following section discusses the incorporation of autonomy in telesurgery.

B. Autonomy in Surgery

Information compression systems require continuous communication due to the need for a surgeon to perform the entire surgery. However, it has been shown that automating small surgery segments can mitigate transmission delays [23], [24]. The feasibility of fully automated surgery has been demonstrated in limited cases such as mastoidectomy [25] and cochleostomy [26]. Nevertheless, full autonomy has not been generalizable to other procedures due to challenges associated with task complexity, soft tissue dynamics, and trust. Semi- autonomous surgical systems provide a middle ground where the surgeon maintains task control and performs the

decision- making while the robot automates small segments of the task [23], [27].

A popular approach to semi-autonomous surgery is to extract task segments from low-level teleoperation data [28], [29], [30], and then autonomously complete these segments. A system for Robotic Minimally Invasive Surgery (R-MIS) was presented in [31], where the surgeon's actions are segmented and further recognized. Then, a predictive controller model drives a robot assistant to automatically perform the recognized actions while the operator maintains supervisory control. A similar semi-autonomous system is presented in [29]. The robot automates the task segments (surgemes) using Gaussian Process Regression to generate trajectories based on demonstrations. However, the surgeon maintains task control to modify robot motion using supervisory functions based on Bayesian optimization. These systems demonstrate the advantage of breaking down the surgery into segments (surgemes) and locally delegating the execution of these surgemes to the machine. Such systems relax the need for continuous communication and provide supervisory control to the surgeon.

C. Augmented Reality and Predictive Displays

Predictive displays and virtual and augmented reality systems have been used for decades for mobile robot teleoperation [11], [32], manipulation using haptic devices [33], [34], [35], and more recently telesurgery [36], [37]. Some researchers have leveraged these technologies to mitigate the effect of delays and network interruptions. For example, Bejczy et al. implemented a control system for a phantom robot under delays [38]. The presented system used a display showing the predicted gripper pose of the phantom during delayed teleoperation, resulting in a 13% reduction of the completion time with respect to traditional teleoperation. Further, Xiong et al. proposed using a predictive display showing a virtual replica of the environment to explore independently from the image streaming frame rate. The teleoperation interface simultaneously controlled the real and virtual robot [12]. This strategy was effective as long as the task was simple enough to be completed without feedback from the real environment. The work in [37] reduced the effect delays at the user interface by integrating a system for surgery where the display shows the predicted position of the surgical grippers. The system concept was tested during a peg transfer laparoscopic task. Finally, The work in [39] proposes a framework called DESERTS that addresses the problem of delays during remote surgery by letting the operator perform the procedure in a virtual replica of the surgical environment. This virtual system automatically recognized the surgical steps carried out during the task. Concurrently, the remote robot received high-level surgical steps that were completed autonomously. Our work expands the DESERTS framework [39] by incorporating a debridement task and reducing the data requirements of surgical step recognition through transfer learning.

III. Methods

The ASAP framework replaces the live-stream video with a simulator-based interface to reduce the information traffic.

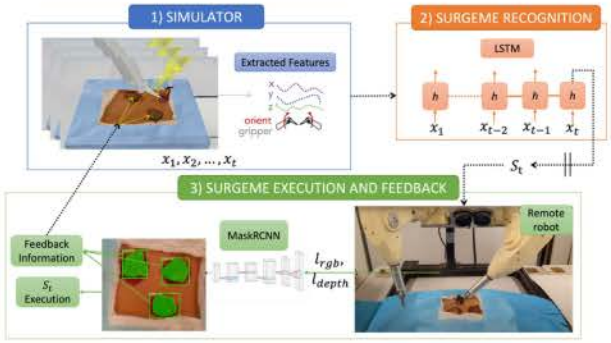

Fig. 1. Framework Architecture.

While the user operates a simulated robot in a replica of the surgical environment, they can also observe the pose of the remote robot and other task-related objects presented through an alpha-blended view in the simulator. At the remote site, an object recognition system identifies these objects of interest in the robot's visual field and forwards them to the simulator for reconstruction. We tested our framework using the two tasks from the DESK dataset [40]: a peg transfer and surgical debridement.

The use of surgemes allows for a high-level abstraction of the surgical procedure, hence reducing the size of the packages sent through the network. Our framework consists of four primary components: recognition, communication, scene interpretation, and execution modules (shown in Figure 1), which are described in detail as follows.

A. Recognition Module

The operator works directly on a realistic simulation that corresponds to the remote robot's environment. Here, a recognition system identifies high-level surgical actions (surgemes) from the end effector trajectories and video scenes. Then, these surgemes are sent to the remote robot, where they are executed semi-autonomously. To leverage the sequential nature of a surgical task, the recognition module predicts surgemes using an LSTM architecture [41]. 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 Δ^K is a probability simplex in K dimensions for K classes. The LSTM predicts class probability $\widehat{y_t} \in \Delta^K$ at time t. As our framework consists of 7 surgeme classes, we have $K = 7$. The seven classes for the peg transfer are: 1) Approach object, 2) Align and Grasp, 3) Lift, 4) Bring together (grippers), 5) Transfer object, 6) Approach peg, and 7) Drop. Alternatively, The seven surgeme classes for debridement are: 1) Approach skin, 2) Align and Grasp, 3) Lift, 4) Approach to cut, 5) Cut, 6) Approach to drop, and 7) Drop. We use a cross-entropy loss defined in Equation (1) to train the LSTM network. The literature recommends the cross-entropy loss for multi-class classification [42], such as our surgeme recognition task.

$$L(y, \widehat{y}) = \sum_{t=1}^{T} \sum_{k=1}^{7} y_{t,k} \log(\widehat{y}_{t,k}) \tag{1}$$

TABLE I
AVERAGE DTW DISTANCES BETWEEN THE TWO TASK SEQUENCES

	Peg Transfer	Debridement
Peg Transfer	96.25	112.69
Debridement	112.69	106.24

The LSTM was used to predict the surgeme at every time step. This frame-wise prediction allowed for proper surgeme identification before the operator completes the action, mitigating the impact of delays in the system. In this work, the recognition module framework was extended to learn in conditions of data scarcity. The LSTM infers the next surgeme through the hidden states based on the action history. Following the hidden layers, there are a series of fully connected dense layers that use the LSTM layer's outputs to generate features for classification. A transfer learning architecture was included to allow the recognition system to learn in a simulated domain, test in a real domain, and learn from surgemes obtained from different procedures.

1) Transfer Learning Between Tasks: Transfer learning was achieved by pre-training a network in the domain of one task (i.e., peg transfer) and then using the weights obtained in that training to initialize the network in the target task domain (i.e., surgical debridement).

Let W_0 be the parameter matrix for the LSTM. This W_o was optimized using stochastic gradient descent for the cross-entropy loss described in Equation (1). Once the training is complete in the source domain, we obtain W_o*. Then, W_o* is used as the initial weight matrix in the target network instead of using a randomized set of vectors.

The peg transfer and debridement tasks involve similar surgemes such as **approach skin/peg**, **align**, and **grasp**. However, the same surgeme can have a very different kinematic appearance during different tasks. For example, the object grasp during peg-and-pole and the skin grasp during debridement look different, given that the former is a solid object and the latter is done on soft, deformable tissue. The dynamic time warping (DTW) distance metric was used to show that the two sequences are different. Table I shows the average DTW distance between every combination of two sequences for the two surgical tasks. These sequences were (x, y, z) comprised of gripper position vectors in inches.

Table I shows that surgeme sequences resemble other surgemes in the same task better than their surgeme counterparts in a different task. Thus, learning from another task presents the challenge of having the same surgemes looking different.

2) Transfer Learning Between Robots: The problem of transfer learning between robots was addressed using the Fast Fourier Transform (FFT) algorithm to map the kinematic data from different robots to the same space. The FFT algorithm allows mapping a time series to a frequency domain. Furthermore, 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 1 shows the developed framework. First, we reduced the dataset to the standard kinematic features in all the robots: grippers' position, orientation, and state (open or close). Next, 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 14 distinct features for each arm (7 each). Thus, for every frame t a 14-dimensional vector X_t was created by concatenating the kinematic features. Then the signal of X_t was mapped to a common space using the FFT discrete function F (see Equation (2)). Finally, a Multi-Layered Perceptron (MLP) was used to classify the new kinematic features.

$$F(X)_k = \sum_{t=1}^{t=T} X_t W_T^{(t-1)(k-1)}, \text{ where } W_T = e^{\frac{-2\pi i}{T}} \tag{2}$$

where T is the total number of frames in the surgeme sequence, X_t is the feature vector, and $1 \leq k < T$. Using a robot-agnostic set of features allows us to leverage information from robots of multiple domains.

B. Communication Module

The communication module was in charge of transmitting data between the operator and the remote robot by facilitating message passing, including high-level surgeme commands and feedback information. The communication module includes two components: (1) Operator to Robot: The identified surgeme in the simulator is sent to the remote robot. This module allows to deal with disruptions. When disruptions occur (delay or interruption), the missing surgeme information from the operator is transferred back to the robot as soon as the connection is re-established. The surgemes are communicated using a TCP protocol. (2) Robot to Operator: The operator state is updated with the latest feedback as early as possible. In case of disruption, the most recent feedback is sent to the operator after the network is re-established. During this communication, the priority is to send the latest data to avoid packet drop. Thus, a UDP protocol is adopted to alleviate the costly handshakes and to update the simulator with the latest feedback messages.

C. Scene Interpretation Module

The scene interpretation 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 3 shows the simulator interface. We first describe the object recognition unit running on the robot side. Then, we describe the simulator's display unit, which reproduces the remote robot scene in the simulator. For the Object Recognition Unit, a 3D camera (Intel Realsense) was mounted on a remote ABB YuMi teleoperated robot. The camera streams color (RGB) and Depth image frames. This image stream was 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. Since the YoloV3 network can only identify objects as regions of interest, an object tracker was added to track objects of the same class (for example,

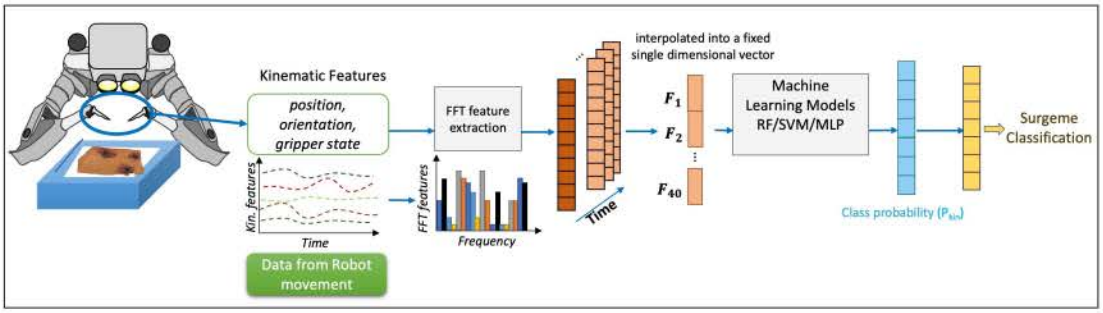

Fig. 2. Architecture overview for transfer learning between robots during surgeme recognition.

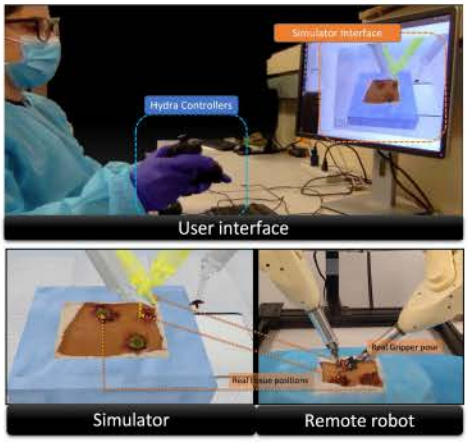

Fig. 3. System Interface for debridement.

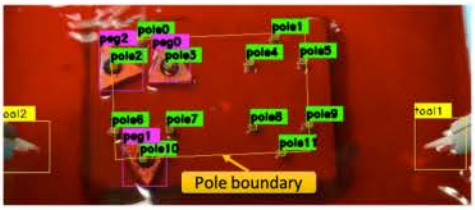

Fig. 4. Detected objects in the peg transfer setup.

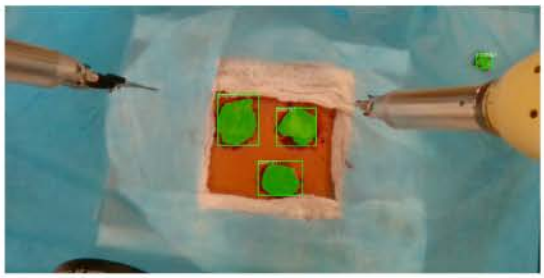

Fig. 5. Debridement skin tracking.

Fig. 6. Peg transfer setup with simulated bleeding.

three triangular objects). Figure 4 shows the tracking and segmentation of the peg transfer objects, and Figure 5 shows the result of the same vision processing on the debridement skin.

The rapid changes during simulated blood turbulence in the environment led to an increased ratio of false positives from 0.01 to 0.5. The false positives led to inconsistencies in the object tracker, making it difficult to estimate the real position of the objects in the scene. The false positives were filtered out using the attributes listed below. For reference, Figure 6 shows the implementation of the turbulence setup.

1) Area Filtering: We included knowledge about the peg board to further filter false positives with 'pole' labels. We defined the pole boundary as an ROI (Region of Interest) for pole detection. The corners of the boundary were defined as $[max(x_i),\ max(y_i)]$, $[max(x_i),\ min(y_i)]$, $[min(x_i),\ min(y_i)]$ and $[min(x_i),\ max(y_i)]$ where x_i and y_i are the coordinates of pole i, with $i = 0, \ldots, 11$ for a total of 12 poles. Figure 4 shows the detected boundary.

2) Shape Filtering: Since the dimensions of the detected objects have similar widths and heights, only detections with an aspect ratio, 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' positions. These estimates are then used to keep track of objects' positions. The tracking system uses the object detection information of the last ten frames and estimates the confidence of the objects' presence, reducing false positives.

3) 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 3. 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. Figure 13 shows the alpha blended objects in yellow in the simulator.

D. Execution Module

After receiving the surgeme from the user simulator, the robot executes it in the real setting. The message the robot receives contains the surgeme label, the prediction confidence, and the surgeme parameters (e.g., for approach, the parameter is the position in x, y, z to approach). Then, the robot executes the surgemes only when sufficient confidence exists in the classification result. The confidence threshold for surgeme execution was set empirically after several evaluations. Once the recognition module classifies the surgeme with high confidence, the remote robot begins performing the surgeme. This occurs before the operator finishes the task, leading to a reduced lag between the operator and robot execution.

E. Surgeme Execution Unit

This unit performs a model-based execution of the surgemes required to complete the peg transfer task. It requires two inputs from the recognition module; surgeme label and parameter. The surgeme parameters represented a target location in the environment (e.g., location of peg). To recognize these parameters, we first utilized a CNN for object detection [43], followed by a Mask-RCNN network for instance segmentation [44]. The execution of the surgemes in the peg transfer task was particularly challenging in the simulated blood-occluded setting [36], because the foreground and background in the environment were uniform, and the liquid surface produced reflections, leading to depth image inaccuracies. To mitigate this issue, morphological dilation was applied to the object mask to fill holes in the depth image, leading to a coherent and complete view.

IV. Experiments and Results

The framework evaluation was done in three steps. First, we assessed the accuracy of the surgeme recognition module, as shown in Section IV-A. Then, we measured the effectiveness of two transfer learning models: 1) Transfer learning between tasks (see Section IV-B, and 2) Transfer learning between robots (see Section IV-C). Finally, we evaluated the frame- work's performance under delays, as discussed in Section IV-D. Throughout the evaluation, two different tasks were used. The first was a peg transfer, which is a standard training task for laparoscopic surgery. The second was a surgical debridement task, which requires bi-manual manipulation of soft tissue with a surgical blade.

A. Live Surgeme Recognition

This section summarizes the surgeme recognition performance during the peg transfer task and debridement trials. The frame-wise recognition accuracy for peg transfer and debridement is shown in Tables II and III, respectively. The system shows a live surgeme recognition accuracy of 70% for peg transfer and 90% for debridement. The table also shows the percentage of surgeme history

TABLE II
Live Surgeme Recognition Performance. (History %) Refers to the Percentage of the Surgeme Used to Cross the Confidence Threshold (%)

	S1	S2	S3	S4	S5	S6	S7	Avg
Confidence Threshold	90	40	40	80	80	70	70	-
History %	54	71	23	24	53	68	51	**49**
Accuracy	88	25	63	81	100	50	81	**70**

TABLE III
Live Surgeme Recognition Performance During Debridement. (History %) Refers to the Percentage of the Surgeme Used to Cross the Confidence Threshold ($\tau = 0.7$)

	S1	S2	S3	S4	S5	S6	S7	Avg
Confidence Threshold	70	70	70	70	70	70	70	-
Accuracy	100	80	100	58	85	100	100	**90**
History %	4	55	13	5	3	17	33	**12**

TABLE IV
Framework Performance Using Surgeme Recognition vs Using the Surgeme's Ground Truth

Task	Recognition System	Ground Truth
Debridement	0.875	0.92
Peg Transfer	1	0.95

(third row in the table) that the system required to reach the recognition threshold. The percentage of surgeme history is defined as $S_i^\tau / S_i, s_i^\tau$ being the number of frames of the surgeme i that the LSTM used in the input to reach a confidence threshold τ, and S_i is the total number of frames of surgeme i (i.e., the length of the surgeme). The results show that for the peg transfer task only 49% of the history is required, while for debridement, only 12% of the surgeme frames were necessary.

Also, we evaluated the recognition module's effect on the system's performance. Table IV shows the task completion rate of the framework using the surgeme classification model vs. using the surgeme ground truth labels (i.e., recognition accuracy of 100%). These results show that, compared to a perfect system, the recognition module reduces the task completion rate by 0.13 and 0.04 for peg transfer and debridement, respectively. In addition, we performed two unpaired, two-sided T-tests, where H0 is defined as "There is no difference between the means of the task performance rate using the surgeme's ground truth and the surgeme recognition module" The H0 was accepted for both tasks ($p = 0.56$ and $p = 0.3$). Meaning there is no significant difference between the means of the two groups (ground truth vs. surgeme recognition).

Finally, Table V shows the task time introduced by each system component. First, the recognition time (RT) was studied in isolation. The system uses an average of 5.6 seconds to recognize the peg transfer surgemes and only 1.2 seconds to classify the debridement surgemes. Then, we studied the performance time, where the results show that the robot executes the task faster than the operator(RE vs. SE) for peg transfer and debridement. Finally, we assessed the recognition system's effect on the user's waiting time (WT). It is important to note that the wait time for recognition overlaps with the user execution time (UE), meaning the user waits only when the remote robot finishes the surgeme after them. In particular,

TABLE V
SYSTEM RECOGNITION TIME VS USER EXECUTION TIME (S). *RT*: RECOGNITION TIME, *RE*: ROBOT EXECUTION TIME, *SE*: SYSTEM EXECUTION TIME ($RT + RE$), *UE*: USER EXECUTION TIME, *W T*: WAIT TIME (SE - UE)

Surgeme	RT	RE	SE $RT+RE$	UE	WT $SE-UE$
Peg Transfer (time in seconds)					
S1	8.2	6.1	14.5	15.3	-0.9
S2	7.2	6.5	13.7	10.1	3.5
S3	1.4	6.3	7.7	6.3	1.4
S4	3.6	5.8	9.4	15	-5.6
S5	9.6	8.5	18.1	18.2	-0.1
S6	5.8	8.7	14.5	8.5	5.9
S7	3.5	9.2	12.7	6.9	5.8
Average	5.6	7.3	12.9	11.5	1.5
Debridement (time in seconds)					
S1	0.7	2.4	3.1	6.4	-3.3
S2	1.7	3.0	4.7	3.1	1.6
S3	0.3	4.1	4.4	2.0	2.4
S4	0.6	4.5	5.1	10.3	-5.2
S5	0.4	18.2	18.6	12.8	5.8
S6	1.1	10.3	11.4	5.9	5.5
S7	3.8	3.4	7.2	11.5	-4.3
Average	1.2	6.6	7.8	7.4	0.4

TABLE VI
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**

Table V shows that the user waits an average of 1.5 seconds during peg transfer and 0.4 seconds during debridement

B. Transfer Learning Between Tasks

Transfer learning is used to reduce the data requirements and allow faster deployment. We used a previously developed simulator where a YuMi robot performs a transfer-learning task [36] and a Taurus robot (SRI International, Menlo Park, CA) performs debridement task [45]. The objective was to train a network in the source domain (peg-transfer) and test it on the target domain (debridement). We used a simulation dataset previously collected in [45]. This dataset contained 185 surgeme sequences from 5 subjects for peg transfer and 88 surgeme sequences from 3 subjects for debridement.

1) Transfer Learning vs No-Transfer Baseline: First, the baseline (no-transfer learning) is presented, followed by the results of different transfer learning setups. The evaluation used a test-train split of 70% and 30%, respectively. The surgeme recognition results for this transfer learning modality are presented in Table VI.

Then, the network was trained with the data from the peg transfer task, producing an average accuracy of 77%. This network was re-trained on the debridement data with the same split as the baseline. The resulting surgeme classification accuracy is shown in Table VII.

2) Effect of Adding Training Data: To better understand the transfer learning setup, we analyzed the effect of adding debridement data to the training. To test this concept, we used only 10% of the debridement data. This amount was increased by 10% until 70% of the data was used. Since the other 30% is used for testing, the number 70% represents all the training data available. Then, the network was initialized with the training weights from the peg-transfer task (source domain) and was retrained on the available debridement data (target domain).

TABLE VII
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**

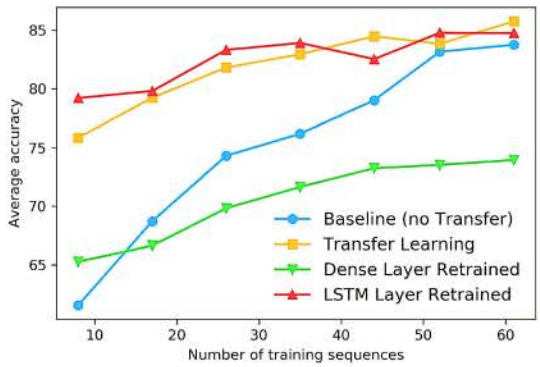

Fig. 7. Impact of increasing the size of the target dataset. The slow increase in accuracy of the transfer learning suggests that the network carries significant momentum from the weights trained using the peg transfer task.

The effect of adding more training data in the target domain is presented in Figure 7. The results show that even when retraining with as little as eight samples, the classification accuracy of the transfer learning setup was 76% which is 22.6% better than the no-transfer setup.

3) Effect of Retraining Network Layers: In object recognition approaches, transfer learning is generally achieved by retraining the last layers. This is because object-specific discriminating features are learned in the last few layers of the network. In contrast, the initial layers produce low-level image features such as texture, edges, or shapes [46]. To analyze the effect that the different LSTM layers have in surgical task classification, we proposed different model variations where we only retrained specific segments of the LSTM: 1) The densely connected layer (only the last layers) 2) The hidden states (referred in the tables as the LSTM layer) and 3) Retraining the entire network.

Figure 7 presents the effect of retraining the different layers of the proposed model. We observe that retraining only the hidden layers (the LSTM layers) achieves the same effect as retraining the entire model.

Figure 8 shows the effect of transfer learning on the convergence rates. All the networks were trained using a learning rate of 0.005 with a stochastic gradient descent optimizer. The transfer learning model allows the model to train faster than the baseline. This is because the networks initialized with pre-trained weights already obtained an initial advantage towards the optima resulting in a faster convergence rate. Moreover, the results show that our transfer learning model only needs

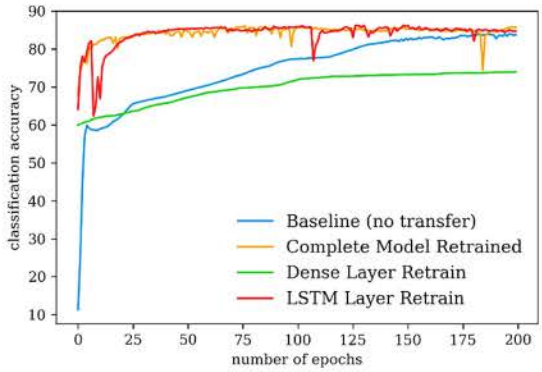

Fig. 8. Impact of retraining different layers in the model in the accuracy.

a third of the epochs to reach convergence compared to the no-transfer baseline.

C. Transfer Learning Between Robots

The goal of this study was to transfer knowledge for surgeme recognition using a diverse set of robots: Simulated Taurus, Real Taurus, Real YuMi (ABB Robotics, Västeraos, Sweden), and the dvRK surgical robot (Intuitive Surgical, Sunnyvale CA.). First, to create surgeme features outside of the robot space, we mapped the kinematic features to a simpler geometric space using an FFT. Then, the FFT features were taken as input to a classifier trained for surgeme recognition. Figure 2 shows the transfer knowledge architecture. Finally, the classification models were trained using simulated data (simulated SRI Taurus) and tested on real robot data to evaluate the system.

1) No-Transfer Baseline: First, three learning models were evaluated using the FFT mapping described at the beginning of this section: 1) A Random Forest (RF), 2) A Support Vector Machine (SVM), and 3) A Multilayered Perceptron (MLP)) [39]. Table VIII shows the recognition accuracy for the no- transfer scenario using all the available data. In addition, the classification was assessed in two modalities: classifying the segmented surgeme and classifying every frame. The first modality requires the surgeme to be segmented in advance, and the second modality is suited for live surgeme recognition. The framework produced models that can accurately classify surgemes. The Taurus II simulator and the real Taurus obtained a maximum accuracy of 88% and 94%, respectively, when using Random Forest (RF). For the da Vinci and YuMi robots, the maximum accuracy was achieved using MLP, with a 95% accuracy for YuMi and a 97% accuracy for da Vinci.

2) Transfer Learning Setup: The training was done exclusively with simulation data and testing in the real robot for the transfer learning scenario. Then, we increasingly added data from the real scenario to the training set to simulate the effects of the 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, defined as $\frac{|X_r^i|}{|X_s|}$ Where $X_s = X_1, \ldots, X_N$ is a set of simulated surgemes of size $|X_s| = N$ and $X_r^i = X_1, \ldots, X_i$ is a set of real surgemes of size $i = |X_r^i|$. When the ratio value is zero, all the data comes from the simulation. When the ratio value is 1, the data had a 50%-50% (50/50 = 1) distribution for real-simulated data. Figure 9 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 gradually added to the simulation data (blue line).

TABLE VIII
ACCURACY FOR THE NO-TRANSFER SCENARIO USING DIFFERENT CLASSIFIERS. SURGEME SEGMENT CLASSIFICATION (LEFT) AND FRAME-WISE CLASSIFICATION (RIGHT)

Robot	Segment			Frame		
	RF	SVM	MLP	RF	SVM	MLP
Taurus sim	88	87	78	86	58	73
Taurus	94	92	92	95	60	92
YuMi	91	93	95	88	48	86
dVRK	88	83	89	90	96	97

3) Transfer Learning Evaluation: The Taurus II and da Vinci robots showed a classification accuracy of 97.5% and 93%, respectively, even when no real examples were included in the training set (data ratio = 0). Also, adding a few real examples improved the surgeme recognition accuracy, as shown in Figure 10. When the real to simulated data ratio 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. The YuMi robot showed a slower convergence, with a ratio of 22%-78% of real to simulated data producing an accuracy of 81%.

D. System Evaluation Under Delays

The system was evaluated under several delay configurations. Then, the task completion rate, completion time, and recognition accuracy were measured. The first two metrics were also used to assess the performance of the same tasks using a standard teleoperation system.

1) Peg Transfer Semi-Submerged in Artificial Blood: We set our peg transfer task based on the DESK dataset that was previously developed [40]. The pegs and poles were partially submerged in 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 10.

To assess the efficacy of the developed framework, we evaluated both the ASAP framework and a standard teleoperation system under three delay modalities: no delay, one second of delay, and five seconds of delay.

Figure 11 shows the results for the framework's completion rate. The task completion rate was maintained in the presence of delays. In particular, the framework shows the same completion rate of 88% for 1 second and 5 seconds of delay. In contrast, the teleoperation performance dropped substantially during the evaluation. First, it dropped to 56% at 0.5 seconds and found its breaking point at 1 second delay, where the completion rate dropped below 20%. Higher delays led to a completion rate of 0%.

The task completion time was also evaluated. The results are summarized in Figure 12. With no delays, the teleoperation showed an advantage over our system: 72 seconds vs.

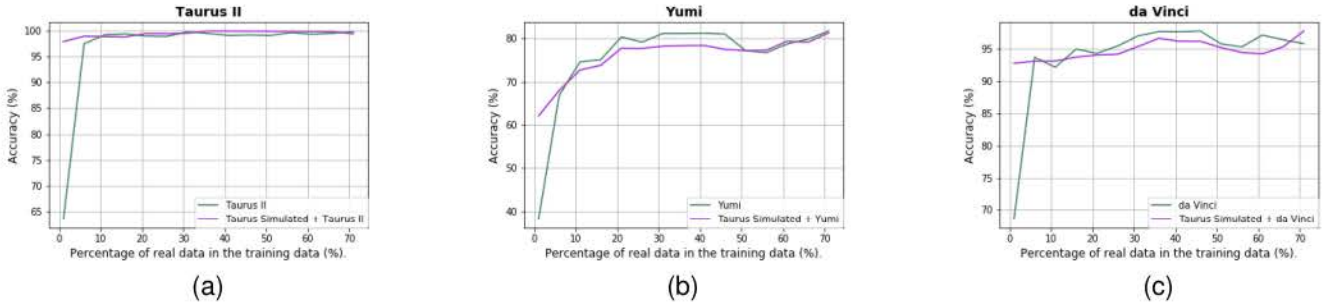

Fig. 9. Performance comparison of training with the real data (no-transfer, shown in green) vs training with only a percentage of the real data combined with simulation data (transfer learning scenario, shown in purple).

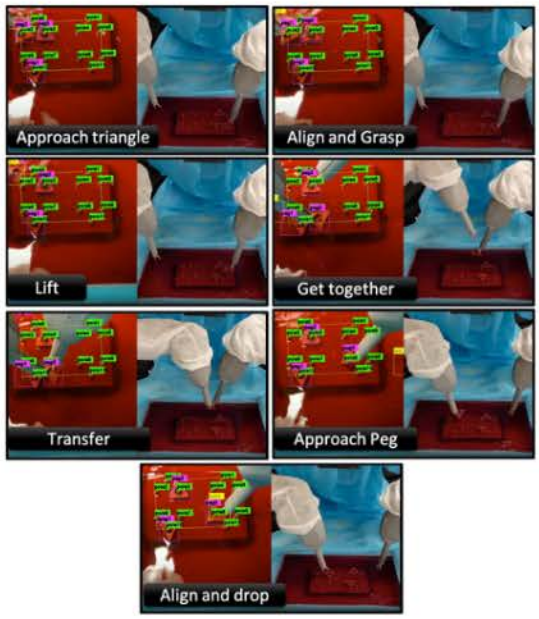

Fig. 10. Surgemes for peg transfer.

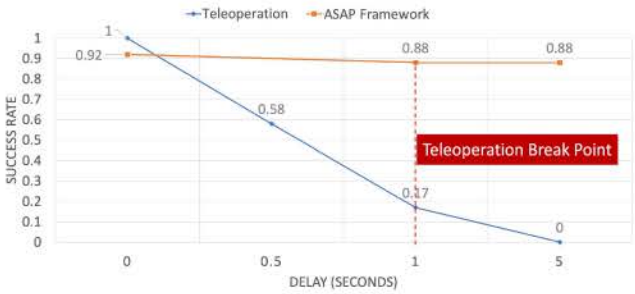

Fig. 11. Task completion rate under delays for Peg Transfer.

91 seconds. Nevertheless, in the presence of delays, the completion time of the proposed framework increases at a much slower rate.

We conducted unpaired t-tests comparing the performance of ASAP vs the teleoperation baseline. The results show that the completion time of the proposed framework was significantly different from the baseline at every delay step. This indicates that from 0.5 seconds to 5 seconds of delay the ASAP time performance was significantly better than the baseline (see Figure 12).

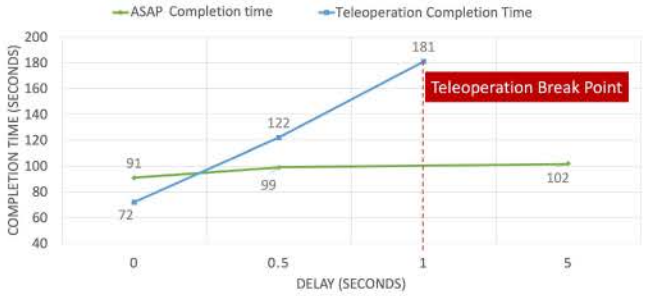

Fig. 12. Completion time under delays for Peg Transfer.

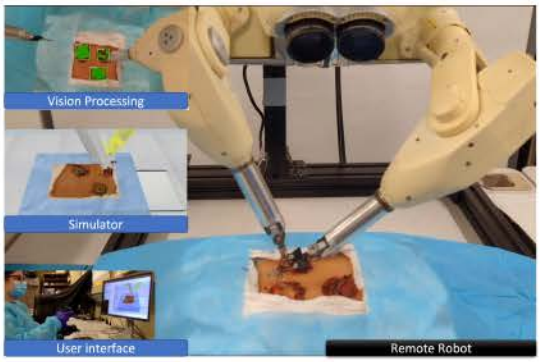

Fig. 13. Framework interface for debridement.

2) Surgical Debridement: The debridement included a piece of simulated skin with three patches of burned skin. The goal of debridement is to remove the necrosis affected regions from the rest of the skin. The skin and tissue were created with elastic properties to reproduce the challenges that the deformable tissue introduces during surgery. A total of 120 trials were collected from three subjects. Each subject performed eight trials for each delay type, completing 40 trials at the end of the session. Figure 10 shows the debridement surgemes.

Five delay configurations were adopted: 1) No delay, 2) 250ms of delay 3) 500ms of delay 2) One second of delay, and 3) Five seconds of delay. Analogous to the previous task, we measured the surgeme classification accuracy, the completion time, and the task success rate. Figure 13 shows a snapshot of the framework working in real-time.

The results for the recognition accuracy at the remote side using different delays are shown in Figure 14. These results

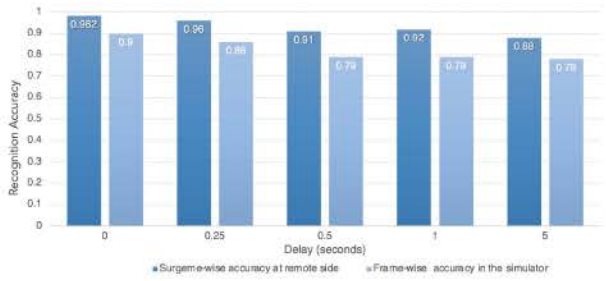

Fig. 14. Surgeme recognition accuracy.

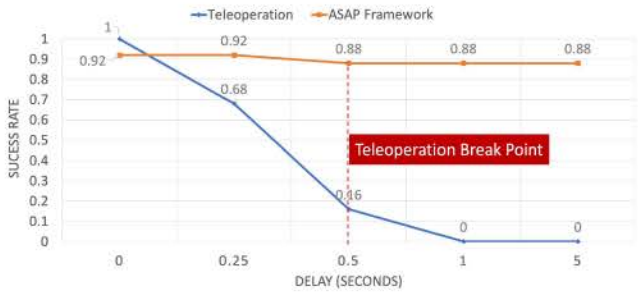

Fig. 15. Task success rate under delays during Debridement.

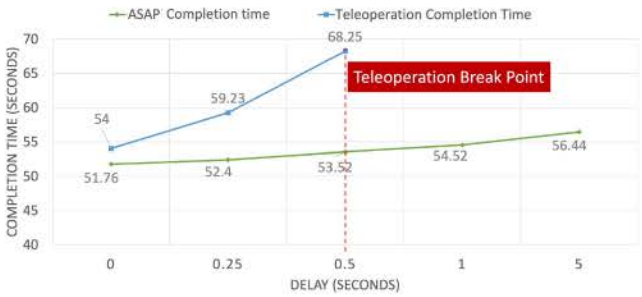

Fig. 16. Completion time under delays during Debridement.

are displayed next to the frame-wise recognition accuracy on the simulator side. The figure shows that the surgeme recognition accuracy is maintained at over 90% until delays of 1s. In particular, the surgeme prediction with no delays achieved a classification accuracy of 98%. With the increase in delays, the user adapts its execution behavior in the simulator. This shift in the user changes the recognition accuracy to 88% when the delay is 5 seconds. Figure 14 shows a softer drop in the remote surgeme recognition when compared to the frame- wise recognition in the simulator. In addition, we evaluated the task completion time in the presence of delays (see Figure 16). At 0.25 seconds, the framework reduces the teleoperation completion time by 11%.

We also performed unpaired t-tests to compare the performance of our framework at every delay point w.r.t the teleoperation with no delay. The test showed no significant differences at any delay setup with respect to the zero-delay teleoperation baseline. This means that at 5 seconds of delay our system showed a completion time that was comparable to the performance of teleoperation with no delays (no significant difference found). Moreover, adding 5 seconds of delay to the system produced an average increase in the completion time of 4.56 seconds, which is slightly lower than the added delay.

3) Transmission Delays: Finally, we analyzed the transmission delays in the surgeme transfer in the 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.

V. Discussion

This section summarizes the findings for each one of the framework components: A) The live surgeme recognition, B) The transfer learning between tasks, and C) Transfer learning between robots, D) The system evaluation over delays, and E) Future work.

A. Live Surgeme Recognition

Results from the live recognition system experiments show that the framework requires a fraction of the surgeme history to achieve an accurate surgeme classification. The peg transfer task required only 49% of the surgeme frames, while the debridement required only 12% of the frames. This result indicates that each instruction can be sent to the remote site before the user finishes performing the instructions, giving the remote robot time to "catch up" with the user even when the instructions are sent asynchronously. The surgeme completion time results discussed in Section V-D confirm this observation. In addition, the evaluation of the effect of surgeme recognition on task performance (see Table IV) shows no significant difference between using the ground truth labels and the surgeme recognition system. This result indicates that the error introduced by the surgeme recognition system does not significantly impact the task completion rate of the system.

Finally, the user wait time results show that the time introduced by the surgeme recognition system (5.6s and 1.2s, for peg transfer and debridement respectively) is compensated in two ways: the user performance overlaps with the recognition time and the robot performs the surgemes faster than the human. Thus, this produces a maximum waiting time of 1.4 seconds for peg transfer and 0.4 seconds for debridement.

B. Transfer Learning Between Tasks

Surgical data in austere environments is scarce. Thus, the transfer learning setup is particularly relevant to the proposed framework since it is crucial to have a system that can learn from data generated during simulation or from another surgical task. By transferring the weights from surgeme recognition on the peg transfer task to the debridement task, the classifier's performance increased by 10% when only 50% of the training data was used. It was also shown that the non-transfer learning setup requires 50% more data to achieve the same performance as the transfer learning setup. To understand the role of each layer in the transfer learning task, each layer was separately retrained. It was found that it is necessary to retrain the hidden layers because the temporal pattern of the kinematic sequences can vary between the debridement and the peg transfer task. Further, we found that the transfer learning models converged to the solution faster than the non-transfer learning method.

We also evaluated the effect of retraining different layers of the network. This assessment showed that retraining the hidden layers of the LSTM produced a higher accuracy and convergence rate than retraining only the last layers. Overall the performance increase indicates that the sequential information of the target domain is relevant to improve the knowledge transfer between surgical tasks.

C. Transfer Learning Between Robots

The results for transfer learning between robots also confirm that the sequential information contributes to classification accuracy. Table VI shows that regardless of the classification algorithm, encoding sequential information into FFT features produced higher accuracy than using the information of a single frame. Furthermore, results for the simulated Taurus showed an accuracy of 99.7% using a ratio of real to simulated data of only 15% to 85%. This performance indicates that surgeme classification in real environments can be achieved using a very small percentage of real data. On the other hand, the YuMi recognition accuracy converged slower, achieving an accuracy of 81% at a real-to-simulated data ratio of 22% to 78%. The discrepancy in accuracy between the Taurus and the YuMi is likely due to the YuMi kinematics. The YuMi robot does not have three degrees of freedom at the gripper, as opposed to the other robots used, making the orientation changes more abrupt. Thus, the teleoperators must choose a convenient orientation and primarily rely on translation motions to generate smoother trajectories. In contrast, the gripper's position and orientation constantly changed 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, when compared to the FFT features generated using the other robots. Thus, it required more of its data during training to produce an accuracy of over 80%.

Finally, the da Vinci and Taurus II showed a faster convergence to a classification accuracy of 95%, requiring little to no data in the transfer learning setup. This indicates that the FFT features describe the spatial properties (shapes) of the surgemes adequately while retaining scale and translation invariance.

D. System Evaluation Under Delays

To simulate the lack of communication infrastructure in an austere environment, The ASAP framework was evaluated under communication delays of up to 5 seconds. Two surgical tasks were performed during the evaluation: 1) A peg transfer and 2) A surgical debridement.

The system performance metrics show that the ASAP framework is robust to delays as high as 5 seconds. For debridement, there was no significant difference between the success rate at 250 milliseconds and the success rate at 5 seconds. In addition, the peg transfer task showed no significant difference in the task performance between no delays and 5 seconds of delay.

Conversely, the teleoperation system started to fail at delays as small as 250ms. Particularly, the teleoperation setup at 250ms showed a completion rate of 58% for the peg transfer task and 68% for debridement. Moreover, teleoperation was unusable for delays longer than 1 second. These results replicate similar values found in literature [1], [2], [3], [4], [5].

The ASAP framework reduced the teleoperation completion time by 45% for the peg transfer task at a delay of 0.5s and 11% for debridement at a delay of 0.25s. This shows that our system successfully mitigates the effect that delays have on performance time. Two factors contributed to these results. The first one is that during teleoperation, the user has to stop every other frame to wait for the feedback image to synchronize with the system controls. In comparison, when operating using the ASAP framework the user only waits for feedback when they wanted to check the remote robot state, which mainly happened between surgical steps and not during them. In addition, early surgeme recognition allowed sending instructions after the user had performed an average of 49% of the surgeme, for the peg transfer task and 12% for debridement. These results show that the robot can start completing the surgeme before the user finishes issuing it, further reducing the idle time.

At 5 seconds of delay, the robot maintained a performance rate of 88%, even when the surgeme recognition accuracy at the simulator dropped to 78%. The task performance was robust to drops in recognition because the surgeme prediction accuracy at the robot side was an average of 0.1 higher than the frame-wise accuracy at the simulator. This is because the surgeme accuracy at the robot side does not solely depend on the frame-wise classification of each surgeme. The accuracy also depends on classification confidence. Thus, the robot did not execute the surgeme until the prediction confidence was reached ($\tau > 0.7$, where τ is the confidence threshold).

Finally, when the delay was 5 seconds, it increased the completion time by 4.56 seconds, for the surgical task. This result implies that the system was not only able to predict the surgeme early but that the robot was executing the surgemes faster than the operator. Thus, the proposed system can mitigate the effects of delays as high as 5s and finish in a time that was not significantly different than the standard undisturbed teleoperation, showing that the framework was robust to delays while keeping a satisfactory performance accuracy.

E. Future Work

One limitation of the current framework is that it was tested on visually-driven tasks. While vision is an essential component of telesurgery, force feedback has also been shown to be crucial to the success of many procedures [47]. Thus, future work will include force feedback, allowing for contact-rich procedures such as bleeding compression or cricothyrotomy. These procedures require a hybrid controller to combine position and force control modes along the different axes. Further, the feedback from both vision and force sensors must be meaningfully combined to estimate the environment state. Future research will include creating a hybrid force-position control scheme with multi-modal feedback [48]. Moreover, future research will also explore error-correcting methods for the current framework. These methods can be addressed autonomously on the robot side when the risk of performing a corrective strategy is low.

VI. Conclusion

In this paper, we proposed a framework to tackle the effects of connectivity associated with telerobotic surgeries. This framework provides a novel simulator interface where the surgeon can operate directly on a virtual reality simulation and the activities are mirrored on a remote robot, almost simultaneously. Thus, the surgeon can perform the surgery while experiencing minimal latency in visual feedback. At the same time, high-level commands are extracted from the operators' 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 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 showed that the system produced a completion time that was faster than teleoperation for the debridement task. With delays as small as 0.25s, the system reduced the completion time by 45% with respect to teleoperation. Moreover, transfer learning between surgical tasks was explored. Our results demonstrate that our framework can boost the performance of surgeme recognition across surgical tasks. When using a pre-trained network, it was found that the classification accuracy achieved 76% with only 8 sequences in the target domain, which is 22.5% better than a no-transfer scenario. To conclude, the presented semi-autonomous system decouples the user interaction from the robot execution, allowing effective teleoperation suitable for remote, rural, and disadvantaged areas.

Acknowledgment

The Computational infrastructure was partially supported by Microsoft AI for Earth Program. The authors wish to acknowledge the U.S. Army Telemedicine and Advanced Technology Research Center (TATRC) and the Telerobotic Operative Network (TRON) project collaborators for their support of this research. The views, opinions, and/or findings contained in this article are those of the authors and should not be construed as an official position, policy, or decision of any of the mentioned institutions, unless so designated by other documentation. TRON project website: https://www.tatrc.org/www/divisions/medras/news/archive/2021-q1-medras-meet-TRON.html

References

[1] T. Kim, P. Zimmerman, M. Wade, and C. Weiss, "The effect of delayed visual feedback on telerobotic surgery," *Surg. Endosc. Other Interv. Techn.*, vol. 19, no. 5, pp. 683–686, 2005.

[2] M. D. Fabrlzio et al., "Effect of time delay on surgical performance during telesurgical manipulation," *J. Endourol.*, vol. 14, no. 2, pp. 133–138, 2000.

[3] M. Anvari et al., "The impact of latency on surgical precision and task completion during robotic-assisted remote telepresence surgery," *Comput.-Aided Surg.*, vol. 10, no. 2, pp. 93–99, 2005.

[4] 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," *Surg. Endosc.*, vol. 28, no. 9, pp. 2569–2576, 2014.

[5] A. Kumcu et al., "Effect of video lag on laparoscopic surgery: Correlation between performance and usability at low latencies," *Int. J. Med. Robot. Comput. Assist. Surg.*, vol. 13, no. 2, p. e1758, 2017.

[6] S. Kassing, D. Bhattacherjee, A. B. Águas, J. E. Saethre, and A. Singla, "Exploring the 'Internet from space' with Hypatia," in *Proc. ACM Internet Meas. Conf.*, 2020, pp. 214–229.

[7] M. Handley, "Using ground relays for low-latency wide-area routing in megaconstellations," in *Proc. 18th ACM Workshop Hot Topics Netw.*, 2019, pp. 125–132.

[8] L. Lévêque, W. Zhang, C. Cavaro-Ménard, P. Le Callet, and H. Liu, "Study of video quality assessment for telesurgery," *IEEE Access*, vol. 5, pp. 9990–9999, 2017.

[9] Z. Wang, I. Reed, and A M. Fey, "Toward intuitive teleoperation in surgery: Human-centric evaluation of teleoperation algorithms for robotic needle steering," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 1–8.

[10] M. Moniruzzaman, A. Rassau, D. Chai, and S. M. S. Islam, "Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey," *Robot. Auton. Syst.*, vol. 150, Apr. 2022, Art. no. 103973.

[11] P. Stotko et al., "A VR system for immersive teleoperation and live exploration with a mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 3630–3637.

[12] Y. Xiong, S. Li, and M. Xie, "Predictive display and interaction of telerobots based on augmented reality," *Robotica*, vol. 24, no. 4, pp. 447–453, 2006.

[13] M. Yip and N. Das, "Robot autonomy for surgery," 2017, *arXiv:1707.03080*.

[14] S. A. Pedram, P. Ferguson, J. Ma, E. Dutson, and J. Rosen, "Autonomous suturing via surgical robot: An algorithm for optimal selection of needle diameter, shape, and path," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 2391–2398.

[15] B. Kehoe et al., "Autonomous multilateral debridement with the Raven surgical robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 1432–1439.

[16] J. Rosen, B. Hannaford, and R. M. Satava, *Surgical Robotics: Systems Applications and Visions*. New York, NY, USA: Springer, 2011.

[17] G. S. Guthart and J. K. Salisbury, "The intuitive/sup TM/ telesurgery system: Overview and application," in *Proc. ICRA Millennium Conf. IEEE Int. Conf. Robot. Autom. Symposia*, vol. 1, 2000, pp. 618–621.

[18] R. Taylor et al., "A steady-hand robotic system for microsurgical augmentation," *Int. J. Robot. Res.*, vol. 18, no. 12, pp. 1201–1210, 1999.

[19] M. Doniec, A. Xu, and D. Rus, "Robust real-time underwater digital video streaming using optical communication," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 5117–5124.

[20] A. Schimpe, S. Hoffmann, and F. Diermeyer, "Adaptive video configuration and bitrate allocation for teleoperated vehicles," in *Proc. IEEE Intell. Veh. Symp. Workshops (IV Workshops)*, 2021, pp. 148–153.

[21] Y. Collet and C. Turner, *Smaller and Faster Data Compression with Zstandard*, vol. 1, Facebook Code, Menlo Park, CA, USA, 2016.

[22] A. Naceri et al., "The vicarios virtual reality interface for remote robotic teleoperation," *J. Intell. Robot. Syst.*, vol. 101, no. 4, pp. 1–16, 2021.

[23] S. Manoharan and N. Ponraj, "Precision improvement and delay reduction in surgical telerobotics," *J. Artif. Intell. Capsule Netw.*, vol. 1, no. 1, pp. 28–36, 2019.

[24] C. M. Korte, "A preliminary investigation into using artificial neural networks to generate surgical trajectories to enable semi-autonomous surgery in space," Ph.D. dissertation, Dept. Aerosp. Eng. Eng. Mech. Coll. Eng., Univ. Cincinnati, Cincinnati, OH, USA, 2020.

[25] A. Danilchenko et al., "Robotic mastoidectomy," *Otol. Neurotol. Off. Publ. Amer. Otol. Soc. Amer. Neurotol. Soc. Eur. Acad. Otol. Neurotol.*, vol. 32, no. 1, pp. 11–16, 2011.

[26] C. J. Coulson, R. P. Taylor, A. P. Reid, M. V. Griffiths, D. W. Proops, and P. N. Brett, "An autonomous surgical robot for drilling a cochleostomy: Preliminary porcine trial," *Clin. Otolaryngol.*, vol. 33, no. 4, pp. 343–347, 2008.

[27] L. Cheng, J. Fong, and M. Tavakoli, "Semi-autonomous surgical robot control for beating-heart surgery," in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, 2019, pp. 1774–1781.

[28] F. Falezza et al., "Modeling of surgical procedures using statecharts for semi-autonomous robotic surgery," *IEEE Trans. Med. Robot. Bionics*, vol. 3, no. 4, pp. 888–899, Nov. 2021.

[29] J. Chen et al., "Supervised semi-autonomous control for surgical robot based on Banoian optimization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2020, pp. 2943–2949.

[30] J. Bohren, C. Paxton, R. Howarth, G. D. Hager, and L. L. Whitcomb, "Semi-autonomous telerobotic assembly over high-latency networks," in *Proc. 11th ACM/IEEE Int. Conf. Human-Robot Interact. (HRI)*, 2016, pp. 149–156.

[31] G. De Rossi et al., "A first evaluation of a multi-modal learning system to control surgical assistant robots via action segmentation," *IEEE Trans. Med. Robot. Bionics*, vol. 3, no. 3, pp. 714–724, Aug. 2021.

[32] D. Lovi, N. Birkbeck, A. H. Herdocia, A. Rachmielowski, M. Jägersand, and D. Cobzaş, "Predictive display for mobile manipulators in unknown environments using online vision-based monocular modeling and localization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 5792–5798.

[33] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, vol. 1, 1988, pp. 131–138.

[34] W. R. Ferrell, "Delayed force feedback," *Human Factors*, vol. 8, no. 5, pp. 449–455, 1966.

[35] Z. Zhao, P. Huang, Z. Lu, and Z. Liu, "Augmented reality for enhancing tele-robotic system with force feedback," *Robot. Auton. Syst.*, vol. 96, pp. 93–101, Oct. 2017.

[36] G. Gonzalez et al., "DESERTS: DElay-tolerant semi-autonomous robot teleoperation for surgery," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 12693–12700.

[37] F. Richter, Y. Zhang, Y. Zhi, R. K. Orosco, and M. C. Yip, "Augmented reality predictive displays to help mitigate the effects of delayed telesurgery," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 444–450.

[38] A. K. Bejczy, W. S. Kim, and S. C. Venema, "The phantom robot: Predictive displays for teleoperation with time delay," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, 1990, pp. 546–551.

[39] G. T. Gonzalez et al., "From the dexterous surgical skill to the battlefield—A robotics exploratory study," *Mil. Med.*, vol. 186, no. S_1, pp. 288–294, 2021.

[40] N. Madapana et al., "DESK: A robotic activity dataset for dexterous surgical skills transfer to medical robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 6928–6934.

[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] S. Pincus and B. H. Singer, "Randomness and degrees of irregularity," *Proc. Nat. Acad. Sci.*, vol. 93, no. 5, pp. 2083–2088, 1996.

[43] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[44] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[45] M. Agarwal et al., "Dexterous skill transfer between surgical procedures for teleoperated robotic surgery," in *Proc. 30th IEEE Int. Conf. Robot Human Interact. Commun. (RO-MAN)*, 2021, pp. 1236–1242.

[46] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[47] A. M. Okamura, "Haptics in robot-assisted minimally invasive surgery," *The Encyclopedia of Medical Robotics: Volume 1 Minimally Invasive Surgical Robotics*. Singapore: World Sci., 2019, pp. 317–339.

[48] M. V. Balakuntala, U. Kaur, X. Ma, J. Wachs, and R. M. Voyles, "Learning multimodal contact-rich skills from demonstrations without reward engineering," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 4679–4685.