Autonomous Temporal Understanding and State Estimation during Robot-Assisted Surgery

Thesis by
Yidan Qin

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2022
Defended May 26, 2022

Yidan Qin
ORCID: 0000-0002-7766-1021

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Joel Burdick, who has been nothing but most supportive, considerate, kind, and caring. He gave me the academic freedom to explore my research interests, supported my research collaborations, and ultimately held my hand through this journey. He has been and will continue to be a role model for me. He is not only passionate about his students' research and academic achievement, but also caring about his students' well-being. I am forever grateful.

I would also like to thank my managers and colleagues at Intuitive Surgical. They are, without a doubt, one of the most pivotal reasons why I am able to pursue my research interests.

I would also like to thank my PhD Committee members: Prof. Richard Murray, Prof. Yisong Yue, and Prof. Yu-Chong Tai.

Last, but most importantly, my family for their love and support. Although I have not been able to visit my parents frequently, their words of love and care supported me through this process. My parents are my biggest supporters, and I love them.

ABSTRACT

Robot-Assisted Surgery (RAS) has become increasingly important in modern surgical practice for its many benefits and advantages for both the patient and the healthcare professionals, as compared to traditional open surgeries and minimally invasive surgeries such as laparoscopy. Artificial intelligence applications during RAS and post-operative analysis can provide various surgeon-assisting functionalities and could potentially achieve a better surgery outcome. These applications, ranging from providing surgeons with advisory information during RAS and post-operative analysis to virtual fixture and supervised autonomous surgical tasks, share a necessary prerequisite of a comprehensive understanding of the current surgical scene. This understanding should include the knowledge of the current surgical task being performed, the surgeon's actions and gestures, the state of the patient, etc. Currently, there is yet to be a unified effort to achieve the autonomous temporal understanding and perception of an RAS at the high accuracy and efficiency required in the highly safety-critical field of medicine.

This thesis develops novel modeling methodologies and deep learning-based models for the autonomous perception and temporal segmentation of the current surgical scene during an RAS. An RAS procedure is modeled as a hierarchical system consisting of discrete surgical states at multiple levels of temporal granularity. These surgical states take the form of surgical tasks, operational steps, fine-grained surgical actions, etc. A broad range of computational experiments were performed to develop methods that achieve an accurate, robust, and efficient estimation of these surgical states. Multiple novel deep learning-based models for feature extraction, noise elimination, and efficient training were proposed and tested. This thesis also shows the significant benefits of incorporating multiple types of data streams recorded by the surgical robotic system to a more accurate surgical state estimation effort.

Two new RAS datasets that contains real-world RAS procedures and diverse experimental settings were collected and annotated–filling a gap in the data sets available for the development and testing of of robust surgical state estimation models. The performance and robustness of models in this thesis work were showcased with these highly complex and dynamic real-world RAS datasets and compared against state-of-the-art methods. A significant model performance improvement was observed in both surgical state estimation accuracy and efficiency. The modeling methodolo-

gies and deep learning-based models developed in this work have diverse potential applications to the development of a next-generation surgical robotic systems.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Yidan Qin and Joel W Burdick. "Concurrent Hierarchical Autonomous Surgical State Estimation during Robot-assisted Surgery". In: (2022).
Y.Q. developed a novel algorithm that concurrently performs surgical state estimations at multiple levels of temporal granularity, performed training and model evaluations, and prepared the manuscript.

[2] Yidan Qin et al. "Autonomous Hierarchical Surgical State Estimation During Robot-Assisted Surgery Through Deep Neural Networks". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6220–6227. DOI: 10.1109/LRA.2021.3091728.
Y.Q. proposed a modeling strategy that describes a robot-assisted surgery as a hierarchical system with discrete surgical states, developed a deep learning-based methods for the estimation of such states, performed training and model evaluations, and prepared the manuscript.

[3] Yidan Qin et al. "Learning invariant representation of tasks for robust surgical state estimation". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3208–3215. DOI: 10.1109/LRA.2021.3063014.
Y.Q. developed a deep learning-based model for the invariant representation learning of time series data during robot-assisted surgeries, performed training and model evaluations, and prepared the manuscript.

[4] Yidan Qin et al. "davincinet: Joint prediction of motion and surgical state in robot-assisted surgery". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 2921–2928. DOI: 10.1109/IROS45743.2020.9340723.
Y.Q. developed a novel approach to the concurrent prediction of robotic surgical instruments' trajectory and surgical state predictions, performed training and model evaluations, and prepared the manuscript.

[5] Yidan Qin, Sahba Aghajani Pedram, Seyedshams Feyzabadi, Max Allan, A Jonathan McLeod, Joel W Burdick, and Mahdi Azizian. "Temporal segmentation of surgical sub-tasks through deep learning with multiple data sources". In: (2020), pp. 371–377. DOI: 10.1109/ICRA40945.2020.9196560.
Y.Q. developed a novel unified approach to the temporal segmentation of surgical sub-tasks during robot-assisted surgery, performed training and model evaluations, and prepared the manuscript.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter 1

INTRODUCTION

Robot-Assisted Surgeries (RAS) procedures usually consists of a relatively consistent series of standardized operational steps such as tissue dissections and suturing. These steps can, in turn, be further decomposed into finer segments of surgical states. These states could take the form of surgical actions, maneuvers, and observations. Thus, many RAS procedures can be viewed in a hierarchical manner. The autonomous awareness and recognition of the current operational step and the fine-grained surgical state is a cardinal prerequisite to many surgeon-assisting functionalities and artificial intelligence (AI) applications in the field of surgical robotics.

This thesis develops novel methods for the autonomous perception, and understanding of RAS using time series data recorded by a surgical robotic system (endoscopic video recordings, robot kinematics data, etc.). This new modeling methodology describes an RAS from a temporal perspective. Past efforts in this area have focused on the estimation of either the current surgical task/step or fine-grained surgical states separately and have mostly utilized a single type of time series data [24, 65, 99]. Additionally, state-of-the-art methods of surgical state estimation have mostly been evaluated only in a bench-top setting due to limitations in data availability, variability, and quality. Such limitations hindered their applications to the safety-critical field of medicine especially for learning-based methods.

This thesis proposes a new modeling strategy that describes an RAS as a hierarchical system of discrete surgical states. Using this concept, I develop unified approaches for a comprehensive temporal understanding of the surgical scene during RAS through the concurrent estimation of these surgical states at multiple levels of temporal granularity and with various types of data available from the surgical robotic system. Multiple deep learning-based models were developed and evaluated for this task.

Two new RAS datasets were also collected in this thesis that contain complex experimental settings and real-world RAS procedures. The developed surgical state estimation models were evaluated and compared against state-of-the-art methods with these more realistic and complex datasets. The performance improvements and

robustness of our models were shown to a fuller extent through the new datasets. An accurate and comprehensive understanding of the current surgical scene during and after an RAS has diverse applications ranging from surgeon skill evaluation and user interface integration to supervised semi-autonomous or autonomous surgical tasks. In the long term, the proposed unified surgical state estimation methods could aid the development of many surgeon-assisting functionalities in the field of surgical robotics research.

1.1 Motivations

The recovery from a surgical procedure involves various risks such as blood loss, pain, post-operative complications, and many others. Throughout the history of surgery, medical professionals and researchers continually strive to provide patients with better care and a smoother recovery by reducing these risks and complications. The first minimally invasive surgery (MIS) was performed in the early 19th century and has thereafter been widely adopted [72]. As compared to traditional open surgeries, MIS procedures are less traumatic to patients, and they reported less pain and blood loss [119]. During a laparoscopic procedure, the surgeons make small incisions in the patient's body. A laparoscope and various surgical instruments are inserted through these openings to perform the surgery (Fig. 2.2). The training of a laparoscopic surgeon, however, requires a steep learning curve [81]. As the surgeons operate with laparoscopic instruments and the laparoscope, the directions of the surgeon's hand movements are opposite to the directions of the laparoscopic instruments' movements inside the patient's body, which is counter-intuitive. Additionally, multiple surgeons are needed to operate various laparoscopic instruments and the laparoscope at the same time. The inevitable hand tremors of the surgeons during the operation also affect laparoscopic surgeries' outcomes.

The first robotic surgical system was cleared by the Food and Drug Administration in 2000, and has since been widely adopted by hospitals and healthcare providers worldwide for the treatment of a wide range of conditions [11]. Extensive studies have shown that RAS patients report less pain, less blood loss, lower complication rates, and quicker recoveries [112] than open surgery and laparoscopic surgery patients. Additionally, RAS surgeons enjoy the convenience of more precise and intuitive instrument manipulations [21]. The da Vinci® surgical system is capable of executing precise surgical actions and gestures teleoperated by RAS surgeons while eliminating the inevitable hand tremors commonly found in laparoscopic surgeries [87]. Additionally, robotic surgical systems are controlled more intuitively and allow

Figure 1.1: The da Vinci® Xi surgical system. From left to right: surgeon-side console, vision-side cart, patient-side cart.

a faster learning process compared to laparoscopy [66], as the surgeon's movements are reflected as-is from the surgeon-side console to the instruments. More than six million robotic surgeries have been performed worldwide as of 2019 by da Vinci surgical systems, according to its developer Intuitive Surgical Inc.; however, RAS currently still only occupy roughly 4% of MIS procedures such as cholecystectomy, abdominal hysterectomy, hernia repair, etc. [53].

In the past decade, the rapid development in AI, computer vision, and deep learning has allowed many models and algorithms to be implemented in various practical applications. Medicine and healthcare is one of the fields that has benefited greatly from AI applications, including computer vision-aided diagnostics, predictive analysis, genetics, and many others [80]. Since robotic surgical systems have the capability to house hardware with high computing power, RAS procedures have the potential to deploy and greatly benefit from advanced AI applications. Currently, RAS procedures are performed in a teleoperative manner.

The da Vinci® Xi surgical system (Fig. 1.1), for example, consists of three major components: the surgeon-side console (SSC), the vision-side cart (VSC), and the patient-side cart (PSC). During an RAS, the surgeon performs the surgery on the SSC, and their action movements are mirrored exactly by the manipulators on the PSC. Details and mechanisms of the da Vinci® Xi surgical system will be discussed in more detail in the following chapters. All of the data used in this thesis came

from some generation of a da Vinci surgical robot.

AI applications have the potential of assisting the surgeons in various ways beyond teleoperation. Surgeon-assisting functionalities include: providing advisory information, user interface (UI) integration, executing supervised autonomous surgical tasks, and many others [15, 21]. Recently, Yang et al. proposed six levels of autonomy in medical robotics that ranges from robot assistance while the human has the continuous control of the surgical system to fully autonomous surgical procedures [125]. While some preliminary efforts have been made towards robot assistance and task autonomy, AI methodology will play an indispensable role in such technology advancements.

An important prerequisite for AI applications in surgical robotics is the awareness of the current stage and status of the surgery being performed [125]. Such awareness should be accurate, comprehensive, and detailed, as the safety of the patient is paramount. The current stage and status of the surgery include many aspects, such as the current surgery step, the current task within that step, and the current action performed by the surgeon, etc. For example, during a prostatectomy RAS procedure, the current surgical step may be an anastomosis. The current task may be two-handed suturing during anastomosis, and the current action is the process of passing the needle from one surgical manipulator to another. A comprehensive understanding of the current surgical scene should include information at multiple levels of temporal granularity for various applications. Knowing the current surgical step of anastomosis is useful for surgeon skill evaluation and post-operative analysis. The awareness of two-handed suturing being performed at its early stage could be used to initiate supervised autonomous suturing. Recognizing fine-grained actions by the surgeon permits surgeon-assisting functionalities such as virtual fixtures and collision prevention.

From a temporal perspective, an RAS can thus be modeled in a *hierarchical* manner. A surgical procedure is commonly performed following a consistent and standard series of steps that were researched, studied, and proven to meet the standard of care for tor the patient [46]. Each of these steps is completed by performing various surgical tasks and maneuvers such as suturing and tissue dissection. Many surgical tasks can in turn be further divided into fine-grained surgical states such as surgeon actions (pushing the needle through the tissue, passing the needle from one surgical instrument to another, etc.) and environmental observations (bleeding, etc.) [32, 96]. Details of such hierarchical system and its formal definition will

be discussed further with examples in Chapter 3. The determination of the current step, task, fine-grained surgical states during RAS has found numerous and diverse AI applications in both intraoperative and post-operative settings. The real-time autonomous awareness of the current step of the surgical procedure, for instance, aids the operating room scheduling staff about the current stage in the surgery, which will allow them to better estimate the remaining time of the surgery [73, 116] and better manage the scheduling of operating rooms. The post-operative temporal segmentation of surgical tasks has found wide applications in surgical workflow analysis [86] and surgeon skill evaluation [32]. The real-time recognition of the current fine-grained surgeon actions and surgical states is a prerequisite for many surgeon-assisting functionalities, such as providing advisory information to the automation of surgical tasks [104].

Prior work in the field of surgical state estimation and temporal segmentation during RAS has only focused on the fine-grained state estimation within a surgical task or the surgical phase recognition in separate efforts. They have mostly only used a single data source (the surgical robot's kinematics time series data or the endoscopic video data as seen by the user) to perform the estimation. The recognition of fine-grained surgical states is particularly challenging due to their short duration and frequent state transitions. The kinematics data of the surgical robot in RAS datasets such as the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) [115] usually takes the form of the 3-dimensional Cartesian positions and velocities of the surgical robot manipulators' end-effectors. Using the robot kinematics data, fine-grained surgical state estimation methods have deployed hidden Markov Models (HMM) [99, 113], in which a surgical task was modeled as a stochastic process. Conditional Random Fields (CRF)[114] was also applied to extract motion primitives of the surgical tasks. Zappella et al. proposed multiple models of modeling surgical video clips for single-action recognition [127]; however this approach does not perform action segmentation, requiring each video clip to contain one and only one action. Other authors have also used a Gaussian Mixture Models (GMMs) to model the time series and segment the series when two consecutive frames belong to different Gaussian distributions, using the Expectation Maximization algorithm to estimate the parameters [135]. Based on GMM, van Amsterdam et al. proposed a weakly-supervised method for action segmentation [6]. The Transition State Clustering (TSC), which uses a Dirichlet process GMM, provided an unsupervised method for surgical trajectory segmentation [58].

More recently, deep learning methods have come to define the state-of-the-art in fine-grained surgical state estimation. Learning-based methods that exploit the temporal correlations among adjacent entries in time series data have been proposed to infer the current fine-grained surgical states and have shown their superiority over traditional probabilistic methods [24, 63, 86]. These methods aim to capture temporal features in RAS time series data streams through popular neural network architectures. Lea et al. proposed the Temporal Convolutional Networks (TCN), which is a convolutional encoder-decoder network along the temporal axis that captures the temporal features of the surgical robot's trajectory when performing certain actions [65]. Recurrent Neural Network (RNN) methods - a widely-used learning model for time series data - have also been explored extensively. Due to its ability to process sequences of data and capture its temporal correlations, Long-Short Term Memory (LSTM) has been widely used in natural language processing and time series prediction [3, 33, 38, 45]. DiPietro et al. applied LSTM to the segmentation of surgical gestures and maneuvers through the learning of the temporal dependencies within the series of robot kinematics data [24, 25]. Menegozzo et al. proposed the Time Delay Neural Network (TDNN), which has a pyramid structure on the temporal axis [78].

With the recent improvement of computing power and the rapid development in computer vision research, applying deep neural network models to analyze video data for various applications has been one of the most popular topics in the field. Vision-based action recognition models have been widely used in the classification of human motions [60, 131] as well as realistic everyday actions [51, 107]. A popular framework for vision-based frame-wise action recognition models is a Convolutional Neural Network (CNN). Some models directly use fully-connected layers for action classification [51] while others use CNNs for feature extractions and other layers such as LSTM to model the temporal dependencies in a video [117]. Such efforts have also been made for surgical fine-grained state estimation. In an RAS procedure, video data is readily available from the endoscopic view as seen by the surgeon during the operation. The endoscopic view provides rich information about the current surgical scene and has gained more attention in the development of surgical state estimation models. The segmentation of fine-grained surgical actions has also utilized similar vision-based action recognition methods as mentioned before. One possible way of modeling the video sequence is by concatenating spatial features at each frame on the temporal axis with Spatio-Temporal CNN (ST-CNN) [64]. Methods introduced in [64] and [49] use a fine-tuned deep CNN model to extract feature vectors from

frames in the RAS video. The sequences of feature vectors are then concatenated and serves as the input of an LSTM model for temporal segmentation. TCN can also be applied to vision data for fine-grained action segmentation, taking the encoding of a spatial CNN as the input [65]. Ding et al. proposed a hybrid TCN-BiLSTM network that incorporates the encoder component of TCN and bidirectional LSTM for improved performances [22]. Similar methods as the ones mentioned above have been applied where the annotated endoscopic videos of surgeries were used to train a CNN-LSTM model in an end-to-end manner [115]. Jin et al. introduced the post-processing of predictions using prior knowledge inference [49]. Since the fully-annotated RAS videos are expensive to obtain, a teacher-student approach of surgical phase segmentation was proposed using a computationally expensive CNN-BiLSTM-CRF model as the teacher network and a more light-weight CNN-LSTM model as the student network that is able to perform real- time inference [126].

As described previously, RAS procedures can often be viewed as the concatenation of a series of operational steps and surgical phases. Similar to fine-grained surgical state estimation, the recognition and temporal segmentation of these tasks and phases also has wide applications. These applications range from surgical workflow analysis and post-operative analysis to surgeon skill evaluation. Comparing to fine-grained surgical state estimation, surgical task or phase recognition requires the model to capture longer-term temporal dependencies in data. LSTM-based models have therefore been shown to outperform their CNN-based counterparts with fixed-size sliding windows for surgical task recognition [24]. Multiple CNN-LSTM models have also been proposed for vision-based surgical phase segmentation [85, 86, 126]. However, these methods also share the weakness of only utilizing one type of input data, vision or kinematics, for temporal segmentation, which inevitably limits their performances.

The limitation shared by the aforementioned single-input surgical state estimation models is the large discrepancy among states' representative vision and kinematics features, making them distinguishable through certain types of input data but not others. For example, the fine-grained surgical action of transferring a needle from one surgical instrument to another is highly distinguishable through the sequential opening and closing of instrument grippers (kinematics data). Although several attempts have been made to incorporate multiple types of input data for surgical state estimation, they have only focused on using the derived values from one data source in addition to the other type of input data. JIGSAWS contains synchronized

endoscopic video and robot kinematics data. Lea et al. measured two scene-based features from the JIGSAWS suturing videos, which were then used as additional variables together with the robot kinematics data [62]. Their LC-SC-CRF model introduces the notion of action primitives, which models each action in JIGSAWS as a sequence of class-specific temporal filters [63]. The addition of visual features (the distance to the closest object part from each tool and the relative position of each tool to the closest object part) offers a richer representation to the relative position of the surgical tools to the needle points, which leads to a higher frame-wise action recognition accuracy. Using derived values, however, saddles the development and training of these methods with additional annotation burdens.

Similar methods have also been implemented for surgical phase segmentation. Multiple temporal clustering methods have been implemented to perform RAS phase segmentation, including GMM, aligned cluster analysis, and hierarchical aligned cluster analysis [132]. The hierarchical aligned cluster analysis aims to decompose a time series into different segments in a manner similar to k-means clustering [131]. These methods have previously been widely used in human motion clustering [26]. A multi-stage temporal convolutional network [19] and the integration of 2D and 3D CNNs [23] were proposed for richer temporal feature learning. Zia et al. collected the robot kinematics and system events data from da Vinci surgeries and fed both data streams through the temporal clustering methods to perform surgical phase recognition [135]. It was shown that the frame-wise temporal segmentation accuracy is considerably improved by incorporating system events data in addition to robot kinematics data.

In addition to robot actions, a fine-grained surgical state could also be the environmental changes observed by the robot. The non-action states were omitted in popular surgical action segmentation datasets such as JIGSAWS [2] and Cholec80 [115]; however, they are important for applications such as autonomous procedures. They are also challenging to recognize as some non-action states may not be well-reflected in a single-source dataset. For instance, bleeding is not as well-represented by the robot kinematics as the endoscopic vision. Hence, the kinematics-based state estimation models may not be able to accurately recognize this state.

To the best of my knowledge, there has yet to be an attempt at hierarchical surgical state estimation in the surgical setting. Existing fine-grained state estimation methods model a surgical task as a set of states; however, there is not yet an accepted definition of an RAS procedure from an estimation perspective. Although

remaining a less explored area comparing to hierarchical semantic or instance segmentation [35, 122], various methods have been developed for hierarchical video temporal segmentation and content characterization [40, 69]. Gunsel et al. divided video characteristics such as motion vectors and color histograms into different categories to segment a video sequence into shots [40]. De Menthon et al. proposed a spatio-temporal segmentation method based on mean shift analysis by mapping each frame in the video sequence to a feature vector describing the color and motion characteristics [20]. Recently, more learning-based hierarchical temporal segmentation methods, either supervised or unsupervised, have been developed. Lan et al. proposed an unsupervised spatio-temporal segmentation method that proposes action-related spatial regions with CRF [27], tracks the segments over time, and clusters fine-grained temporal segments into higher-level segments [61]. The method uses a linear Support Vector Machine (SVM) classifier as the discriminative algorithm for classification.

The highly complex and diverse real-world RAS environment calls for a robust hierarchical surgical state estimation model, which is especially crucial in the field of medicine. The development of such model, however, is not a trivial task, and various challenges remain. Among real-world RAS procedures, the endoscope lighting and viewing angles, the patient's anatomical structure and health condition, surgical backgrounds vary considerably. These variations are considered as *nuisance factors* in RAS data. Additionally, different surgeons may employ diverse surgical techniques to perform the same surgical task or fine-grained surgical actions based on their personal preference, their training, their natural handedness (e.g., right-handed vs. left-handed), and the patient condition. A robust hierarchical surgical state estimation model needs to be able to combat such nuisance factors and technique variations and remains accurate in the highly complex and diverse real-world RAS environment. While the adverse effect of nuisance factors and surgical technique variations can be alleviated by a large and diverse annotated realistic RAS dataset during model training, such datasets are extremely costly to acquire due to factors such as patient privacy concerns, annotation costs, resource limitations, etc.

Improving model robustness against a highly dynamic and complex environment remains one of the most challenging obstacles in the field of computer vision and machine learning that prevents some models from being implemented in real-world applications. Model robustness is especially important in AI applications for RAS, as safety is of great importance. The robustness of hierarchical surgical state estimation

could be boosted from many aspects, however has not been extensively investigated prior to our work. AI applications in RAS are mostly based on machine learning techniques, which rely heavily on the dataset used for training and evaluation. Prior surgical state estimation efforts have suffered from dataset limitations. Additionally, surgical state estimation robustness could be boosted through model architecture design.

1.2 Contributions and thesis outline

The remainder of this thesis is structured as follows: Chapter 2 presents my efforts on the development of two new RAS datasets, which include the curation, processing, and feature extraction of surgical activities and real-world RAS procedures using two types of da Vinci® surgical systems. The hardware and software components of the surgical systems used in this thesis work are described along with their data-recording capability. Comparing to time series datasets in action recognition or speech recognition such as ActivityNet [14], RAS data enjoys the luxury of having synchronized endoscopic vision, robot kinematics, and system events data. Past RAS datasets such as JIGSAWS and Cholec80, however, contained limited data sources. Additionally, these datasets suffer from a lack of variety in environmental settings, activity elements, endoscope movements, etc. These disadvantages hindered a thorough examination of surgical state estimation models' performance and robustness against a complex surgical environment.

Part of the work described in this thesis therefore curated, annotated, and processed two new RAS datasets that contains various experimental settings and real-world RAS procedures for the development, training, and evaluation of our surgical state estimation methods. The Robotic Intra-Operative Ultrasound (RIOUS+) dataset contains trials of an ultrasound scanning task that is common among RAS for the understanding of patient anatomical structures. The HERNIA-40 dataset contains complete real-world inguinal hernia repair procedures, which is extremely valuable as it captures the complexity of the real-world surgical environment. These datasets allowed us to train and evaluate our models to the fullest extent. The details of these datasets are described in Chapter 2 as well. As this thesis work presents multiple deep learning-based surgical state estimation efforts, a review of machine learning models and architectures used was also given in Chapter 2. Due to the highly complex and dynamic nature of RAS time series data, I developed various data processing and feature extraction methods prior to the estimation of surgical states. Different feature extraction strategies were deployed to accommodate the

diverse features in the endoscopic video, robot kinematics, and system events data. These strategies are also discussed in details in this chapter.

Although efforts in standardizing the description of an RAS procedure has been made, they are far from ideal or suitable for the task of surgical state estimation. Chapter 3 proposes a strategy that models an RAS procedure from the temporal awareness perspective, in which an RAS is modeled as a hierarchical system of discrete surgical states. This system is referred to as the surgical Hierarchical Finite State Model (HFSM). The formal definitions of surgical states, HFSM, and its components are given. Additionally, the key differences between our definitions and a traditional Finite State Machine (FSM) as defined in the field of automata theory are discussed. Chapter 3 also reviews surgical ontology - the mainstream modeling strategy for RAS description - and its limitations for surgical state estimation and AI applications in RAS.

In Chapter 4, a deep learning-based unified approach for fine-grained surgical state estimation that incorporates multiple types of input data sources is described. To the best of our knowledge, this was the first surgical state estimation model that utilizes the endoscopic video, robot kinematics, and system events data available during an RAS procedure. The proposed model (denoted as Fusion-KVE) improved state-of-the-art fine-grained surgical state estimation performance by up to 7.3%. Fusion-KVE was evaluated and compared against prior work using the real-world RAS datasets that I developed. When a more complex and realistic dataset was used, our proposed unified model was able to show its robustness to the fullest extend. Additionally, this chapter discussed the necessity and advantages of incorporating multiple types of input data. Different types of RAS time series data (endoscopic video, robot kinematics, system events) represents an RAS procedure from their respective perspectives and contains different types of features associated with the current fine-grained surgical state. They therefore have their respective strengths and weaknesses in the identification of different surgical states. By incorporating multiple types of RAS data, the proposed unified surgical state estimation approach was able to learn and utilize features embedded in each type of input data. Richer information about the current surgical state can therefore be extracted for more accurate state estimation results.

Many surgeon-assisting functionalities during RAS, ranging from virtual fixture of the surgical instruments to the supervised autonomy of surgical actions and tasks, share the prerequisite of surgical instrument trajectory and surgical state predictions.

Chapter 5 proposed a joint surgical instrument path and state prediction model during RAS (denoted as daVinciNet). daVinciNet performs multi-step predictions of the surgical instruments' end-effectors' 3D Cartesian trajectories in the endoscope frame along with the future fine-grained surgical states. During the evaluation of model performance, daVinciNet accurately predicted the end-effector trajectories and surgical states for up to 2 seconds into the future. Additionally, an ablation study was performed to confirm the necessity and effectiveness of the feature extraction methods I developed in Chapter 2 to the surgical state prediction performance. daVinciNet implements a novel endoscopic video feature extraction method that goes beyond feature extraction directly from the raw endoscopic video frames. As the end-effectors' trajectories and future surgical states are usually associated with the user's current action, the area surrounding the surgical instruments' end-effectors in an endoscopic video frame should contain more information about their future trajectories and the procedure's future surgical states. The visual features within this region were therefore separately extracted in addition to global visual features from the entire endoscopic video frame. By incorporating localized endoscopic vision features to both the trajectory and surgical state predictions, the prediction performances of both end-effectors' trajectories and future fine-grained surgical states were improved. Specifically, the end-effectors' trajectory prediction accuracy was improved by up to 8.1%. Fusion-KVE from Chapter 4 was used to produce the historic surgical state sequence in addition to features extracted from input time series data to perform fine-grained surgical state prediction. The state prediction contributions of both the features from the current surgical scene and the historic surgical state sequence was also shown.

Chapter 6 proposes a method for the invariant representation learning of surgical states. One of the major roadblocks in the development of an accurate and robust surgical state estimator is the complex nature of the current surgical scene during an RAS due to the high variability of surgical background, patient conditions, anatomical structures, lighting conditions, etc. These variations are nuisance factors in RAS data and hinders an effective feature extraction and state estimation. This adverse effect is especially troublesome with limited data availability. While the negative effects of these nuisance factors could be reduced by training the surgical state estimation model with a large and diverse RAS dataset, such dataset is costly to acquire. Additionally, in real-world RAS procedures, different surgeons may employ various surgical techniques and styles to accomplish the same surgical task/goal. These different surgical techniques have diverse vision and kinematics

features and further increases the difficulty of training an accurate surgical state estimator with limited data availability. A representation of the current surgical scene that is invariant to such nuisance factors and surgical techniques is therefore highly desirable. Chapter 6 therefore proposed StiseNet - a surgical state estimation model with an invariance induction framework that minimizes the effect of nuisance factors and different surgical techniques. StiseNet achieves such invariance induction through adversarial training that separates other factors from information pertinent to surgical state estimation. Through the training and evaluation with real-world RAS procedures performed by surgeons, StiseNet improved the surgical state estimation accuracy of state-of-the-art methods and our previous work by up to 7%. The contributions of using an invariant representation of the RAS time series data features for state estimation was proven through the comparison between StiseNet and its variation that was not trained in an adversarial manner. The effectiveness of invariance induction to nuisance factors and the invariance induction to surgical technique variability was separately shown through an ablation study. Chapter 6 also demonstrated another endoscopic video feature extraction technique I developed as described in Chapter 2. To eliminate the negative effect of the highly complex and dynamic surgical background on endoscopic vision feature extraction, a semantic segmentation model was implemented to generate a semantic mask of the current endoscopic video frame. The semantic mask assigns each pixel of the video frame to one of three scene classes, and was concatenated to the video frame for a more effective feature extraction.

In Chapter 7, two hierarchical surgical state estimation models are proposed to concurrently estimate the current surgical stage/task (superstate) and fine-grained surgical state, which is the first attempt at concurrent hierarchical surgical state estimation in the field of surgical robotics research. The models, denoted as HESS-DNN and CHASSEN, were demonstrated through the simultaneous estimation of surgical states at two levels of temporal granularity using the HERNIA-40 dataset. Chapter 7 also explored the usage of correlations between surgical states at different levels of temporal granularity for a more effective hierarchical surgical state estimation method. The current surgical task (superstate) and fine-grained surgical state are usually highly correlated. Some fine-grained states could only occur during certain surgical superstates but not others. For example, the fine-grained surgical state of pushing a needle through tissue can occur during the surgical superstate of suturing; however, it would not occur during dissection as needles are not used during dissection. The knowledge of the current surgical state at one level of temporal

granularity is therefore valuable for the estimation of surgical state at another level. To utilize this knowledge, an alternating training schematics was deployed to train a hierarchical surgical state estimator with both direct data sources (endoscopic video, robot kinematics, system events) and inferred data sources (surgical state sequence from another level of temporal granularity). I showed that incorporating the correlations between surgical superstates and fine-grained states improved both the state estimation accuracy and efficiency through the comparison of model performances and process times between the two proposed models.

Finally, Chapter 8 summarized the contributions of this thesis work to the field of robotic surgery research. Future work directions and extensions to current surgical state estimation model architectures were also suggested.

As mentioned previously, hierarchical surgical state estimation finds diverse applications both during and after the RAS procedure. To comprehensively evaluate the surgical state estimation performances of models in this thesis work for both real-time and post-operative applications, two experimental settings were used to evaluate the performance of state estimation models in this thesis work: causal and non-causal. In a causal setting, the models only have knowledge of the current and preceding time steps. This is to mimic the real-time state estimation application of our model, in which the robot cannot foresee the future. In a non-causal setting, the models have access to data from the future time steps. The models' performance in a non-causal setting better represents their potentials for post-operative applications such as surgeon skill evaluation.

Chapter 2

DATASET COLLECTION AND VISUAL FEATURE EXTRACTION EFFORTS

The da Vinci® surgical system, designed and manufactured by Intuitive Surgical Inc., is the first robotic surgical system approved by the Food and Drug Administration [21] for human clinical use. Throughout the last two decades, multiple iterations of da Vinci® surgical systems have been designed and manufactured. Specifically, two types of robots were used to collect RAS data utilized in this work: the da Vinci Research Kit (dVRK) and the da Vinci® Xi (dVXi) surgical system. These surgical robotic systems share the same teleoperation concept: the user operates two manipulators in front of a console, and their actions are mirrored to surgical instruments operating on the patient.

Throughout an RAS procedure, these surgical robotic systems are capable of recording various data streams, including the endoscopic video as seen by the operator, the robot kinematics data, and system events such as the pressing of a button or pedal. These diverse types of time series data provide a comprehensive understanding of the RAS procedure and contains rich information for the temporal perception of the current surgical scene. They are therefore extremely valuable for the estimation of the current surgical state at various levels of temporal granularity. Due to the complex and dynamic nature of an RAS, however, these data streams are also high-dimensional and noisy. While this problem could be alleviated by the collection and annotation of a large and diverse RAS dataset, such dataset are costly to acquire. Effective feature extraction methods are therefore necessary for the training of deep learning-based surgical state estimation models using these data sources.

In the following sections, an overview of two types of da Vinci® surgical systems used in this work for data collection, including its hardware and software components, data availability and recording methods are given. Three RAS datasets, including two collected as part of this thesis work, were used for model training and evaluation. These datasets' features and components are also described in this chapter. Additionally, an overview of the machine learning models and architectures used in this thesis work was given.

I also developed various methods for a more effective spatial feature extraction

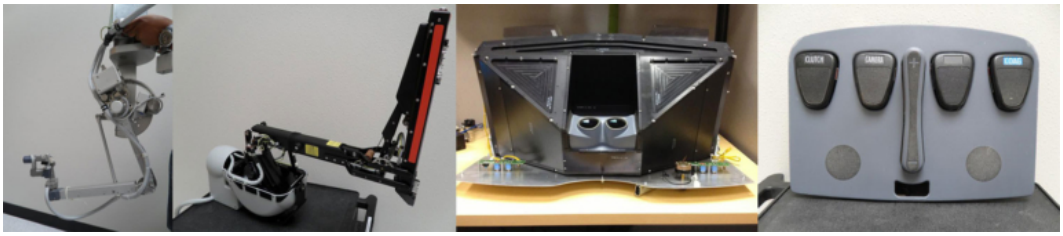

Figure 2.1: Components of the da Vinci Research Kit (dVRK). From left to right: Master Tool Manipulator, Patient-Side Manipulator, High Resolution Stereo Viewer, foot petal tray.

from the endoscopic video data prior to surgical state estimation. As mentioned previously, the combination of limited availability and high diversity of RAS time series data calls for such effort. In Section 2.3, these methods are described in details. The work presented in this chapter was described in publications [95, 96].

2.1 Hardware components and data recording capability

Throughout the last two decades, multiple iterations of da Vinci® surgical systems have been designed and manufactured. Specifically, two types of robots were used to collect RAS data utilized in this work: the da Vinci Research Kit (dVRK) and the da Vinci® Xi (dVXi) surgical system. Whilst dVRK is a popular open-source platform for surgical robotics research ranging from image-guided surgeries to developing novel surgical instrumentation [52], the da Vinci® Xi surgical system - a proprietary product - is the 4^{th} generation robotic surgical system and has served in hospitals worldwide on millions of patients. These two surgical robotic systems both have a master-slave design for the teleoperation of RAS. The operator controls multiple robotic arms from a central console where the endoscopic vision are displayed. Comparing to laparoscopic surgery, these robotic systems improves the user's control and dexterity of both the endoscopic camera and surgical instruments during an operation. Additionally, the teleoperation design removes the hand tremor effect that made laparoscopic surgery a challenging task. The movements of the user at the console are communicated to surgical instruments with flexible wrist design, which enables a precise control of the instrument that mimics direct hand control but with a wider range of motion. The teleoperation design also creates a more comfortable surgical environment, as the surgeon can be seated by the console while operating. Additionally, as the surgeon is able to control both the endoscopic camera and surgical instruments from the console, the surgical workflow is more streamlined as no assistant is needed to move the endoscopic camera during the

Figure 2.2: Comparison between a laparoscopic procedure and a dVXi procedure. Left: sample image during a laparoscopic procedure in which the surgeon needs to stand besides the patient and an assistant is required to operate the endoscopic camera. Middle and right: sample images during a dVXi surgery in which the surgeon is seated in front of the SSC.

operation. Sample surgical scenes during a laparoscopic procedure and a dVXi procedure are shown in Fig. 2.2, which showcases the superiority of an RAS over other MIS surgeries.

The dVRK is a telesurgical system consisting of firmware, electronics, and software components based on the first-generation da Vinci system [52]. It contains two Master Tool Manipulators (MTMs), two Patient-Side Manipulators (PSM), a High Resolution Stereo Viewer (HRSV), and a foot petal tray (Fig. 2.1). It also includes the Surgical Assistant Workstation (SAW) package based on the open-source *cisst* libraries [50], which provides teleoperation, high-level control, joint-level control, low-level I/O, and Robot Operating System (ROS) interfaces. dVRK uses proprietary mechanical hardware on the da Vinci system, which allowed it to mimic an RAS environment in an experimental setting. dVRK records the forward and inverse kinematics of both the MTMs and the PSMs as well as the endoscopic view from the HRSV.

The da Vinci® Xi surgical system (dVXi) consists of three main hardware components (Fig. 1.1): surgeon-side console (SSC), vision-side cart (VSC), and patient-side cart (PSC). It is a proprietary robotic surgical system. It contains two MTMs on the SSC and four Universal Patient-Side Manipulators (USMs) on the PSC. The VSC houses a 3D viewer of the current endoscopic view as well as processors and energy instruments. The details of MTMs and the endoscopic view displayed on both the SSC and VSC are shown in Fig. 2.3. dVXi records the kinematics data of both the MTMs and the USMs and calculates the 3D Cartesian positions and velocities data of their end-effectors along with the manipulators' gripper angles. High resolution endoscopic video was also recorded by the VSC. Additionally, various

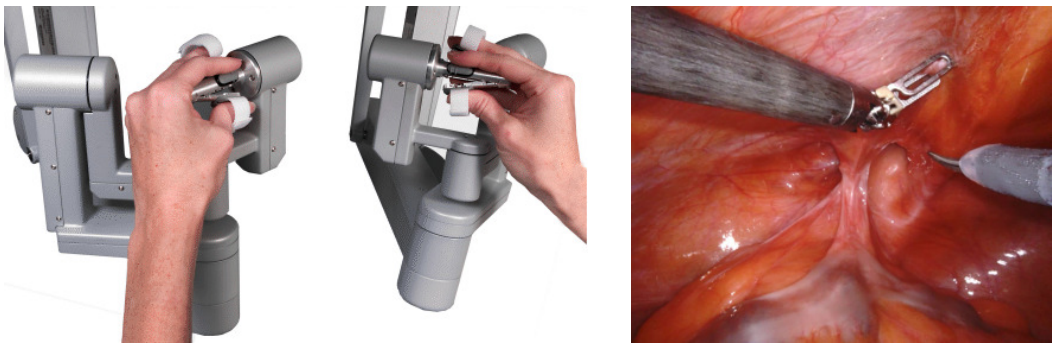

Figure 2.3: Details of the da Vinci® Xi surgical system. Left: MTMs used by an operator to control the surgical instruments. Right: The endoscopic view as seen by the operator through the surgeon-side console.

system events occur during an RAS procedure such as the pressing of a button or foot pedal, clutching of the MTM manipulators, operating energy instruments, etc. These events are also recorded by dVXi.

2.2 Dataset curation

Both dVRK and dVXi have the capability of generating and recording various types of time series data when being operated. dVRK has been one of the main platforms for RAS data curation for surgical robotics research, generating many popular RAS datasets including the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) [32], Colec80 dataset [115], m2cai16-workflow dataset [110], and many others; however, dVRK records both the endoscopic video and robot kinematics data at with limited resolution and frame rates. dVXi, on the other hand, is able to record high-resolution stereo endoscopic video data and high-frequency robot kinematics data. It also provides an additional data source of system events during an RAS. Due to proprietary reason, however, there has not been an RAS dataset collected with it prior to this thesis work. I was able to receive processed endoscopic video, robot kinematics, and system events data from Intuitive Surgical Inc. recorded with dVXi from both a bench-top setting and during real-world RAS procedures, which allowed us to generate two new RAS datasets.

Three RAS datasets were used in this work: JIGSAWS, RIOUS+, and HERNIA-40. Whilst JIGSAWS is surgical activity dataset in a bench-top setting, RIOUS+ and HERNIA-40 contains data recorded in a real-world RAS environment. In the following subsections, each dataset's content and features are described in details.

Figure 2.4: Example endoscopic video frames from the JIGSAWS suturing dataset.

JIGSAWS

The JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) [32] dataset, collected by The Johns Hopkins University and Intuitive Surgical, Inc., has been one of the most commonly used surgical activity datasets in the field of surgical robotics. JIGSAWS includes data on three surgical tasks commonly observed during RAS training and practice, in a bench-top setting (Fig. 2.4):

- Knot-tying: The user picks up one end of a surgical suture and ties a single loop knot;
- Suturing: The user picks up a needle, inserts the needle at a dot marked on one side of a suturing pad and exits on the corresponding dot on the other side. The needle is then extracted from the suturing pad. This process is repeated three more times;
- Needle-passing: The user picks up a needle and passes it through four metal hoops.

Only the suturing dataset of JIGSAWS was used in this thesis. It contains 39 trials performed by eight users with varying levels of robotic surgical experience, ranging from novice to expert. Each user repeated the same suturing task four or five times. JIGSAWS suturing dataset is annotated manually with nine atomic surgical activities referred to as *gestures* [32], which are treated as fine-grained surgical states. Fig. 2.4 shows sample scenes from the JIGSAWS suturing dataset. Table 2.1 lists these nine states and their average durations and Fig. 2.5a illustrates some of the state transitions observed in JIGSAWS suturing dataset.

The JIGSAWS suturing dataset includes the following kinematics and video data. The kinematics data of dVRK's two MTMs and two PSMs was recorded using its SAW at 30 Hz. Each manipulator's tool tip Cartesian positions (3 variables), a rotation matrix (9 variables), linear and rotational velocities (6 variables), and

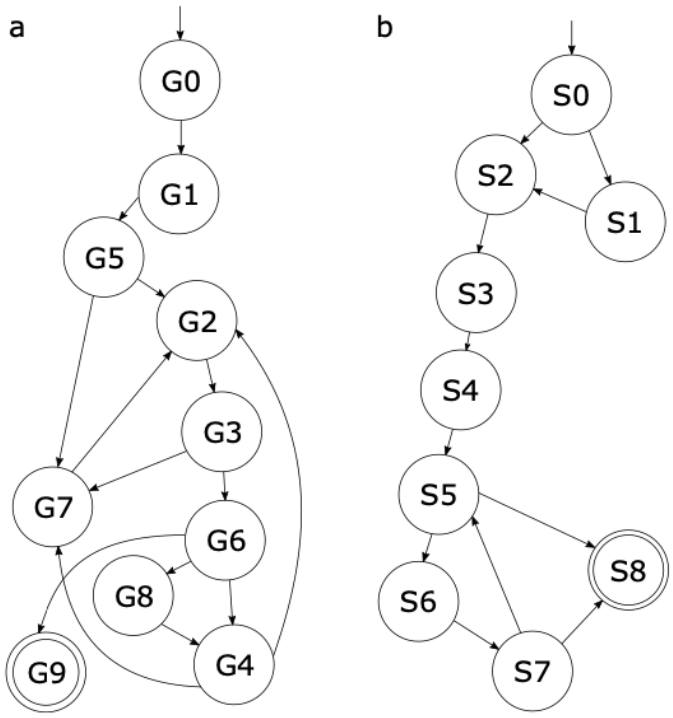

Figure 2.5: Observed state transitions of the JIGSAWS suturing task (a) and the RIOUS+ ultrasound imaging task (b). The 0 states are the starting of tasks. The states with a double circle are the accepting (final) states. The actions in the JIGSAWS suturing task are represented with gestures (G) and the states in the RIOUS imaging task are represented with states (S).

gripper angle (1 variable) were released. The stereo video from dVRK's endoscopic camera was recorded at a 620×480 resolution and 30Hz. The video data was synchronized with the kinematics data.

Table 2.1: JIGSAWS Suturing Dataset State Descriptions and Duration

Gesture ID	Description	Duration (s)
G1	Reaching for the needle with right hand	2.2
G2	Positioning the tip of the needle	3.4
G3	Pushing needle through the tissue	9.0
G4	Transferring needle from left to right	4.5
G5	Moving to center with needle in grip	3.0
G6	Pulling suture with left hand	4.8
G7	Orienting needle	7.7
G8	Using right hand to help tighten suture	3.1
G9	Dropping suture and moving to end points	7.3

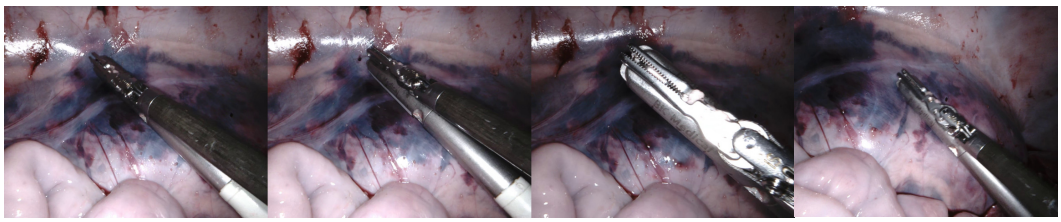

Figure 2.6: Example endoscopic video frames from the RIOUS+ dataset.

RIOUS+

While JIGSAWS remains one of the most popular open-source surgical activity datasets for the past decade, its has obvious limitations. The users were not allowed to move the endoscopic camera while performing the surgical task, which is unrealistic during real-world RAS, during which endoscopic camera movements are frequent and spontaneous. Additionally, JIGSAWS's bench-top experimental setting also hinders its real-world applicability. With the recent development in machine learning-based surgical state estimation research, JIGSAWS' limited dataset size, lack of trial variety, and unrealistic experimental setting significantly lessens its effectiveness.

I therefore curated a Robotic Intra-Operative Ultrasound (RIOUS) dataset to address some of JIGSAWS' limitations. The usage of a drop-in ultrasound probe to scan the organs is a technique widely observed in many RAS procedures. It is commonly practiced by surgeons to localize an organ's underlying anatomical structures such as tumors and vasculature. The state estimation of this surgical task would allow us to develop many surgeon-assisting functionalities during this task ranging from UI integration to its supervised automation. The RIOUS dataset was later expanded to RIOUS+ dataset by adding new trials and users.

The RIOUS+ dataset contains 40 trials of the ultrasound scanning surgical task performed by five users in various experimental settings. 27 trials were performed on a phantom kidney in a bench-top setting, 9 were performed on porcine kidneys in an OR setting, and 4 were performed on a cadaver liver performed in an OR setting. The ultrasound machine used is the bk5000 with a robotic drop-in probe from BK Medical Holding Company, Inc. During each trial, the user was instructed to perform the scanning task with actions from a pre-determined list of states, but had no limitation on transitions between states nor endoscopic camera movement. Users also had no limitation on trial lengths. The eight surgical states and their average durations are shown in Table 2.2 with free transitions between them. Fig.

Table 2.2: RIOUS+ Dataset State Descriptions and Duration

State ID	Description	Duration (s)
S1	Probe released, out of endoscopic view	6.3
S2	Probe released, in endoscopic view	7.6
S3	Reaching for probe	3.1
S4	Grasping probe	1.1
S5	Lifting probe up	2.4
S6	Carrying probe to tissue surface	2.3
S7	Sweeping	5.1
S8	Releasing probe	1.7

2.6 shows sample scenes from the RIOUS+ dataset. Fig. 2.5b presents sample state transitions of the ultrasound imaging task observed commonly in the RIOUS+ dataset.

The RIOUS+ dataset includes three types of synchronized time series data available from a dVXi surgical robot: robot kinematics, endoscopic video, and system events data. The kinematics data of dVXi's two MTMs and four USMs were recorded at 120Hz For each manipulator, the same 19 kinematics variables, consisting of the rotations of each mechanism joint, as the JIGSAWS suturing dataset were included. The stereo video from the endoscopic camera was recorded at a 1280×1024 resolution and 60Hz. In addition to surgical instrument movements and endoscopic camera movements, the following six binary events are recorded during each RAS:

- surgeon head in/out of the SSC: when the surgeon's head is out of the console, instruments are not able to move;
- camera follow: when the camera follow foot pedal on the SSC is pressed, the MTMs control the USM with the endoscope installed and all other USMs are not able to move;
- instrument follow: when the camera follow foot pedal on the SSC is not pressed, the USM holding the endoscope is not able to move;
- master clutch (2 variables): when an MTM is clutched, its movements are not reflected to the PSMs;
- ultrasound probe activation;
- ultrasound probe in contact with tissue.

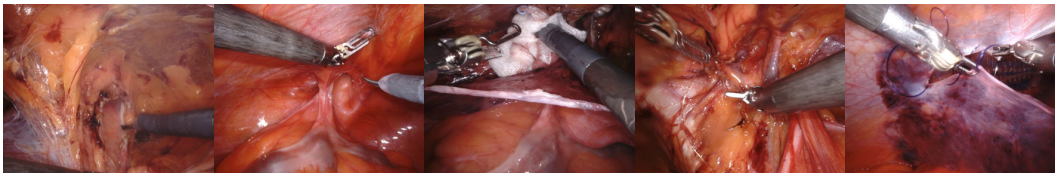

Figure 2.7: Example endoscopic video frames from the HERNIA-40 dataset.

System events are a cardinal component of any surgery, as they contain rich information about the current surgical states.

The robot kinematics, endoscopic video, and system events data were synchronized and downsampled to 30 Hz.

Despite addressing some of JIGSAWS' limitations, such as a fixed endoscope position and limited range of experimental settings, RIOUS+ is a small surgical dataset that incorporates only a single surgical task, with a limited number of trials, and a lower surgical complexity. During data collection, the users were instructed to strictly adhere to a pre-determined list of actions; therefore, the variation among trials is still limited, as the surgical task was performed in a highly structured and uniform manner.

HERNIA-40

The JIGSAWS or RIOUS+ surgical trials only includes one surgical task that consists of fine-grained surgical states. These datasets are therefore not suitable for the development of a comprehensive RAS state estimation solution that recognizes the current states of RAS at multiple levels of temporal granularity. Additionally, their limitations in dataset size and variability, simplistic experimental settings, and rigid pre-determined lists of allowed fine-grained surgical actions are non-negligible limitations. As surgical state estimators rely heavily on datasets for model fitting and training, these limitations are propagated, and possibly even amplified, to affect the performance of a surgical state estimator designed using this data. For example, during a bench-top trial in the JIGSAWS suturing dataset, in which suturing is performed on a marked pad, valuable anatomical background information is missing from the endoscopic video data, which may lead to insufficient training of a vision-based surgical state estimation model. Trials performed using the same technique or following the same predetermined workflow, as they were in JIGSAWS suturing and RIOUS+ datasets, could cause overfitting of the state estimator model.

A real-world RAS dataset consisting of complete RAS procedures performed by a

diverse set of surgeons on real patients is therefore highly desirable. To meet such needs, I collected the HERNIA-40: an RAS dataset that contains 40 real-world robot-assisted transabdominal preperitoneal inguinal hernia repair procedures using the dVXi surgical system. Thirty of these surgeries are performed in a unilateral fashion and 10 are bilateral hernia repair procedure. A bilateral hernia repair procedure targets hernias that occur on both side of the lower abdomen and therefore has the similar operative steps and workflow as a unilateral hernia repair procedures [46]. Since HERNIA-40 contains real-world robot-assisted-surgeries, patient privacy-related information such as the surgeon's identity is not available; therefore, the number of unique surgeons in HERNIA-40 is also unknown.

Since the HERNIA-40 data set contains both unilateral and bilateral hernia repair procedures, each trial lasts from 15 to 90 minutes. Unlike the JIGSAWS or RIOUS+ datasets, in which each trial contains a single surgical task with one level of temporal granularity (fine-grained surgical states that make up the surgical task). All of the HERNIA-40 trials are annotated with two levels of temporal granularity under the professional guidance by practicing surgeons. The standard-of-care operative steps of an inguinal hernia repair surgery are considered the surgical superstates of the surgical HFSM. Each surgical superstate is, in turn, broken down into fine-grained surgical states. A complete list of surgical superstates and fine-grained surgical states within each superstate is shown in Tables 2.3 and 2.4. Fig. 2.7 shows sample scenes from the HERNIA-40 dataset.

As the trials in HERNIA-40 capture complete surgical procedures, multiple robotic surgical instruments are involved across all of the trials, and the tools are sometimes uninstalled/reinstalled onto the USMs during a surgery. Additionally, laparoscopic instruments were used in all trials. In HERNIA-40, three types of da Vinci surgical

Superstate ID	**Inguinal Hernia Repair**	Percentage of instances (%)	Mean duration (s)
SS0	Create peritoneal flap	16.5	74.6
SS1	Dissect mesh pocket	19.7	88.1
SS2	Dissect hernia sac	16.9	136.1
SS3	Deploy mesh	18.0	138.7
SS4	Close peritoneum	16.8	202.3
SS5	Endoloop suture	3.1	131.3
SS6	Anchor mesh	5.3	117.2
SS7	Manipulate gauze	3.7	81.7

Table 2.3: HERNIA-40 superstates descriptions and mean durations.

Fine-grained Surgical States

Create Peritoneal Flap	Mean duration (s)
Cut peritoneum with monopolar scissors	5.7
Stretch peritoneum with left hand	3.1
Adjust endoscope	2.9
Dissect Mesh Pocket	**Mean duration (s)**
Pull tissue with left hand	4.2
Local energized cut	2.8
Push and cut tissue	3.9
Adjust endoscope	3.1
Dissect Hernia Sac	**Mean duration (s)**
Push and cut tissue	4.9
Pull and cut tissue	4.1
Pull tissue with left hand	4.3
Push tissue with left hand	2.6
Adjust endoscope	3.7
Deploy Mesh	**Mean duration (s)**
Unfold mesh	6.6
Push mesh to tissue with left hand	2.9
Push mesh to tissue with right hand	2.6
Pull tissue with left hand	3.8
Pull tissue with right hand	3.4
Manipulate mesh	7.0
Adjust endoscope	2.8
Close Peritoneum	**Mean duration (s)**
Reach for the needle	3.9
Position the tip of the needle	3.3
Push needle through the tissue	4.2
Pull tissue with left hand	3.6
Transfer needle from left to right	3.7
Orient needle	6.6
Pull suture with left hand	5.8
Pull suture with right hand	4.8
Transfer needle from right to left	4.6
Use right hand to tighten suture	4.3
Adjust endoscope	3.8

Table 2.4: HERNIA-40 fine-grained states descriptions and mean durations.

instruments (ProGrasp™ forceps, a large needle driver, and monopolar curved scissors) and two types of laparoscopic instruments (tack fixation device and suction irrigator) are used. The HERNIA-40 dataset includes the robot kinematics, endoscopic video, and system events data from five different dVXi systems. These time series data streams were collected and synchronized in the same way in the RIOUS+ dataset. The system events that were collected from the dVXi are, however, different from those found in RIOUS+. Specific to inguinal hernia repair surgery, with its higher surgical complexity, six binary system events and four categorical system events were included:

- binary events
  - energy pedal: when the energy foot pedal on the SSC is pressed, a da Vinci E-100 generator integrated with dVXi executes controlled energy delivery to the tip of the monopolar curved scissors;
  - surgeon head in/out of the SSC;
  - camera follow;
  - instrument follow;
  - master clutch (2 variables).
- categorical events
  - Type of instruments installed on USMs (4 variables): the da Vinci surgical instrument installed on each of the four USMs of dVXi.

Both the JIGSAWS suturing dataset and the RIOUS+ dataset are used in Chapters 4, 5, and 6 for the evaluation of fine-grained surgical state estimation accuracy of multiple models. The HERNIA-40 dataset is used in Chapters 4, 6, and 7 for the evaluation of both fine-grained state estimation accuracy and hierarchical surgical state estimation accuracy.

2.3 Data processing and feature extraction

A real-world RAS is characterized by a highly complex and dynamic operating environment. As mentioned in previous chapters, numerous nuisance factors and variations in surgeon techniques further complicates real-world RAS datasets. The combination of limited data availability and high complexity calls for an effective and robust method of data processing and feature extraction method, which is

a fundamental prerequisite for robust hierarchical surgical state estimation. The datasets used in this work are also high dimensional, since they contain high-resolution endoscopic video feeds and a large number of kinematics variables as the surgical robot has high degrees of freedom. Accurate surgical state estimation also benefits greatly from an effective feature extraction method for these complex time series data streams. In the following subsections, I first review two basic machine learning models and their mechanisms. Then, various data processing and feature extraction methods I developed for the endoscopic video data, robot kinematics data, and system event data are described. To effectively extract features that are relevant to hierarchical surgical state estimation, some deep learning-based feature extraction models should not be trained and frozen separately. I therefore only describe the feature extraction models' architectures in this chapter. The training details of certain feature extraction models will be further discussed in following chapters.

A review of machine learning models: Convolutional Neural Networks and Recurrent Neural Networks

Machine learning techniques and models have become the cornerstone of AI applications among various fields of research, including RAS. This section provides a brief review of the Convolutional Neural Network (CNN) and the Recurrent Neural Network (RNN) - two representative and most fundamental building blocks of learning-based models and techniques used in this work.

Convolutional Neural Network

A CNN is a deep learning algorithm that takes an image as input, assigns an *importance* to areas/objects/aspects of the image based on the CNN's parameters, and then differentiates one from another. An image is a matrix of pixel intensity and color values. For small or extremely simple images, simple feature extraction methods such as flattening the image or principal component analysis might serve the purpose. However, as images become complicated and have higher dimensions, a CNN reduces the images to an easier-to-process form without losing features that are potentially important, as would be done by some dimension reduction methods.

A typical CNN has three types of layers: a convolutional layer, a pooling layer, and a fully-connected layer [36]. A convolutional layer performs a dot product between two matrices - a *kernel* and a region of the receptive field. A kernel consists of

a set of learnable parameters and is spatially smaller than an image in order to produce representations of that receptive region - the *activation map*. This process is carried out throughout the image with stride S and padding P, which determines the sliding size of the kernel and the padding size of the image, respectively. As Compared to trivial neural network layers, in which every input unit interacts with every output unit, the convolution operation has sparser interactions between the input and output. When a large or complicated image is processed with a kernel, localized and meaningful features and relevant information within a small region of the image can be detected; therefore, both the model's memory requirement and efficiency are improved.

A pooling layer reduces the spatial size of the image representations by calculating the summary statistics of nearby outputs of the network [36]. Common pooling operations include max pooling (returns the maximum value of the output region) and average pooling (returns the mean value of the output region). This operation further decreases the memory and computational requirements of the model. Unlike convolutional layers, a pooling layer does not contain any trainable parameter. A fully-connected (FC) layer, like its name suggests, connects all activation maps in the previous layer and is commonly used at the end of a CNN model to consolidate all features extracted by previous layers and generate the final output (e.g., classification results). Since convolution is a linear operation and an image - especially a high-dimensional one - is far from linear, it is a common practice to introduce non-linearity to the outputs following convolutional layers [36]. Popular non-linear functions that were explored in this work include Sigmoid, Softmax, Tanh, and Rectified Linear Unit (ReLU) functions. A Sigmoid function ϕ is a logistic function that maps its input z - values from the neurons of the previous layer - to $(0, 1)$ and is commonly used to generate an output that is a *probability*:

$$\phi(z) = \frac{1}{1 + e^{-z}}, \tag{2.1}$$

whereas a Softmax function σ is used in multi-class classification and outputs the probability of each class:

$$\sigma(z_i) = \frac{e^{z_i}}{\Sigma_j e^{z_j}}. \tag{2.2}$$

A Tanh function, as its name suggests, maps its input to the range $(-1, 1)$. Since negative inputs will be mapped to a negative value by the Tanh function, it is

more commonly used in binary classifications. The ReLU function, f, has gained popularity in deep learning research for its linear behavior, output sparsity, and simplicity [41]:

$$ReLU(z) = max(0, z), \tag{2.3}$$

however, the half-rectified ReLU function outputs zero for any negative input, which may result in insufficient training due to units not being activated initially [71]. A leaky ReLU function f_{leaky} relaxes such constraint:

$$ReLU_{leaky}(z) = \begin{cases} z & \text{if } z > 0 \\ az & \text{otherwise,} \end{cases} \tag{2.4}$$

where a is a small value (usually 0.01).

With the rapid development of deep learning and computer vision research, numerous CNN architectures have been proposed, examined, and improved for various applications ranging from multi-class classification to pixel-level segmentation [98, 106]. The various state-of-the-art spatial and temporal CNN models that were explored in this work will be discussed in details in the following chapters.

Recurrent Neural Network

While CNNs remain the most popular class of algorithms for image feature extraction, RNNs are the state-of-the-art type of neural networks for modeling and processing time series data [33] because of their internal memory, which allow RNNs to remember important input information as a hidden state (a representation of previous inputs), and use that information for prediction. Fig. 2.8 illustrates the difference in information flow between an RNN and a feed-forward neural network: RNNs apply learnable weights to both the current and previous inputs and also adjust its internal parameters through backpropagation and gradient descent [36].

One of the outstanding challenges for simple RNNs is the vanishing gradient problem [42]. During the training of an RNN, the gradient is the value used to tweak the networks' trainable parameters. As each time step in an RNN is treated as a layer, backpropagating through time could result in an exponential shrink in the gradient values. The network would thus make extremely small adjustments to its weights, and therefore would not be trained effectively. Historically, the lack of long-term memory hindered the development of RNNs until the proposal of the Long-Short

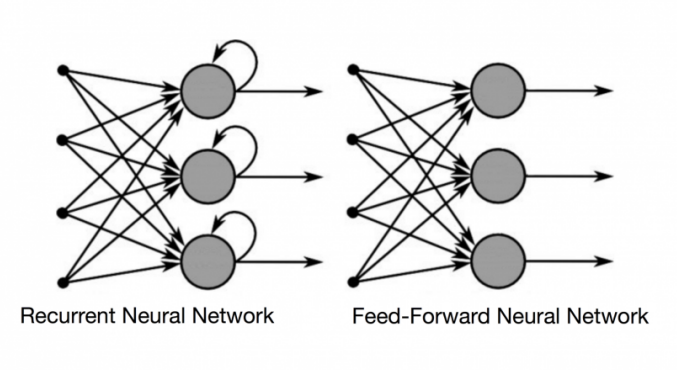

Figure 2.8: Comparison between the information flow of RNN and feed-forward neural network.

Memory (LSTM) [43] model. An LSTM unit has three *gates* to regulate the information in its cell state: a forget gate, an input gate, and an output gate.

The LSTM unit at time step t first needs to decide what information to discard from the previous time step $t-1$'s hidden state $\boldsymbol{h}_{t-1}$ and the input at time step t $\boldsymbol{x}_t$. This is done through a forget gate - a sigmoid function:

$$f_t = \phi(\mathbf{W}_f(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t) + \mathbf{b}_f), \tag{2.5}$$

where $\mathbf{W}_f$ and $\mathbf{b}_f$ are learnable weights and bias parameters of the forget gate. As a new input is received at time step t, an input gate then decides what information to use to update its cell state:

$$i_t = \phi(\mathbf{W}_i(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t) + \mathbf{b}_i), \tag{2.6}$$

where i_t is the extracted input information. The cell state at time step t c_t is then updated from the previous cell state:

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(\boldsymbol{W}_c(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t) + \mathbf{b}_c), \tag{2.7}$$

where $\circ$ denotes a point-wise multiplication. Finally, an output gate decides the hidden state of the LSTM unit at time step t through the learning of what information to output:

$$o_t = \phi(\mathbf{W}_o(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t) + \mathbf{b}_o) \tag{2.8}$$

$$\boldsymbol{h}_t = o_t \circ \tanh(c_t), \tag{2.9}$$

where o_t represents the output information that was used to calculate the new hidden state $\boldsymbol{h}_t$.

As deep learning algorithms for sequential data have gained more popularity, a number of architectural variations have been proposed that significantly improve the capability of RNNs. In this thesis, the attention mechanism - a revolutionary improvement for the encoder-decoder LSTM model - was extensively applied [118]. Originally proposed for neural machine translation [17], an encoder decoder LSTM model - as its name suggests, consists of two components. The encoder LSTM accepts the input sequential data and encode it into a context vector, which is represented by the last hidden state of that LSTM model. This hidden state is then passed to the LSTM decoder. An outstanding problem of only using the last hidden state of the encoder LSTM is that the rapid performance deterioration as for long sequential data streams due to the exploding/vanishing gradients [17]. Since the model only uses the last hidden state, it lacks long-term memory and would perform poorly when the input sequential data becomes longer. Additionally, the context vector represented by the last hidden state does not distinguish the importance differences among different parts of the input sequential data. The attention mechanism tackles this problem by incorporating all hidden states in the encoder LSTM model to the context vector instead of only using the last hidden state [9]. An additive attention mechanism, for example, calculates the context vector at time step t as a weighted sum of all encoder hidden states of length l. At time t, the attention weights $\boldsymbol{\alpha_t} \in \mathbb{R}^l$ are determined from the previous decoder hidden state $\boldsymbol{d}_{t-1}$, encoder hidden states $\boldsymbol{H}$, and cell state as:

$$\boldsymbol{\alpha}_t^j = \text{softmax}(\boldsymbol{u}_\alpha^T \tanh(\boldsymbol{W}_\alpha(\boldsymbol{d}_{t-1}, \boldsymbol{c}_{t-1}) + \boldsymbol{V}_\alpha \boldsymbol{H}_t^j)) \tag{2.10}$$

for $j \in [1, l]$ where $\boldsymbol{u}_\alpha$, $\boldsymbol{W}_\alpha$, and $\boldsymbol{V}_\alpha$ are learnable parameters.

Endoscopic video data

An endoscope becomes the surgeons' eyes during an RAS. The endoscopic video data is therefore highly informative to our hierarchical surgical state estimation effort, and such data is available in all datasets used in this work. The raw endoscopic video data, however, is high dimensional and contains various nuisance factors ranging from brightness and viewing angles to highly variable surgical backgrounds and anatomical structures. As computer vision research advances aggressively, various methods were implemented to effectively extract features (thereby effecting a data reduction step) from endoscopic video data prior to surgical state estimation.

As mentioned in previous chapters, CNNs have been used in prior surgical state estimation research. I used the VGG16 network [106] - a state-of-the-art spatial

CNN architecture originally developed for image recognition. The VGG16 model maps a $224 \times 224 \times 3$ RGB image to a vector $\mathbf{X} \in \mathbb{R}^N$ where N is the number of visual features. At time t, The RGB endoscopic video frame $\mathbf{I} \in \mathbb{R}^{h \times w \times 3}$ where h and w are the height and width of the frame is resized and the *global* visual features from the entire video frame. To effectively extract visual features that are relevant to hierarchical surgical state estimation, parameters in the VGG16 models were initialized with ImageNet pre-trained weights [106] and fine-tuned using transfer learning [121]. The training details will be further discussed in following chapters.

The background environment in real-world RAS contains various nuisance factors, making it highly complex and varies significantly from one trial to another. While the adverse effect of such noise can be reduced by training the hierarchical surgical state estimation model with a large annotated real-world RAS dataset, such dataset is not currently available. Two alternative methods were experimented to address this challenge during the endoscopic video feature extraction phase. As most surgical states are associated with the surgeon's movements and actions on the robotic surgical system, it is reasonable to assume that the surgical instruments' end-effectors and their surrounding areas are especially informative for surgical state estimation. Localized features from these areas could therefore be an effective addition to global vision features extracted from the entire endoscopic view. One prerequisite of this method is the tracking of surgical instruments' end-effector locations in the endoscopic view. Following the work of Allan et al. [5], a silhouette-based instrument tracking model was used for bounding box-based end-effector detection. The detection model was separately trained with extensive and diverse real-world RAS images and frozen. A bounding box is referred to as a Region of Interest (RoI) as it is a localized region in the endoscopic view with potentially more concentrated information about the current surgical states. Two convolutional layers with ReLU activation were implemented as the CNN model to extract features from an RoI instead of a VGG16 network to accommodate the image size difference. Fig. 2.9 illustrates the concatenation of global vision features extracted from the entire endoscopic view and the localized vision features.

In addition to using localized vision features from RoIs, another effective method to combat the noises and nuisance factors in the surgical background during RAS during feature extraction is through *semantic image segmentation* [28]. The semantic segmentation of an image refers to the computer vision technique that classifies each pixel in the image into a category and generates a semantic mask of the image in

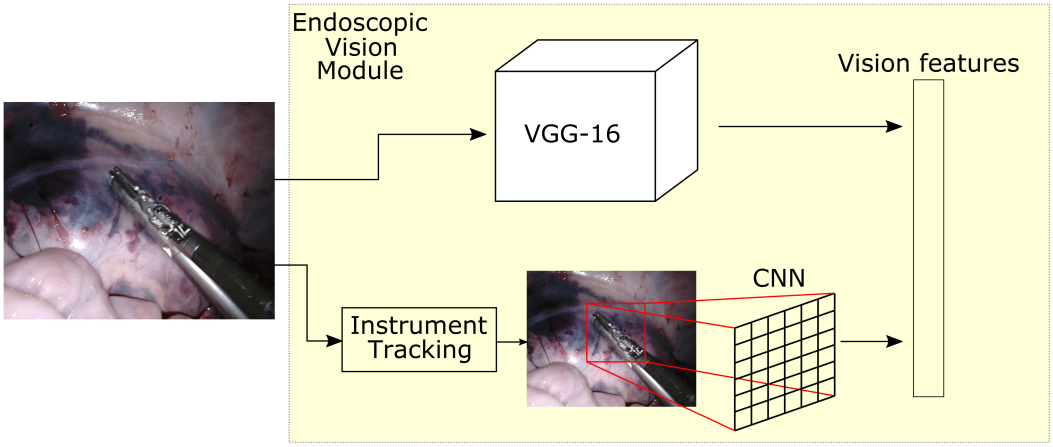

Figure 2.9: An endoscopic vision feature extraction module that incorporates both the global vision features and localized vision features.

which the pixel value is classified into its category label value. It is achieved through CNN-based deep learning methods and has rapidly gained popularity in computer vision research areas including medical imaging [114]. The advantages of generating the semantic mask of an RAS endoscope view is intuitive. The surgical background is complex and dynamic with various objects and noises like smoke, blood, needles, etc. The direct feature extraction of such surgical scene would then include features of these objects which could be irrelevant to surgical state estimation. A semantic mask of the surgical background effectively eliminates such distractions if the only available pixel classes are organs, surgical instruments, and background. Fig. 2.10 illustrates an example RGB image of an endoscopic view and its semantic segmentation map with the above three classes.

Semantic segmentation of endoscopic videos of RAS procedure has been one of the most popular research topics in RAS research [105, 133]. One of the prevailing surgical scene segmentation models is U-Net: a CNN-based architecture as shown in Fig. 2.11 [98]. A U-Net model first contract the input image's features through a series of convolution and pooling operations for downsampling. An expansive path then upsamples the feature map and eventually output a segmentation map. Each pixel location is assigned one of three classes: tissue, surgical instruments, and others. Following the work of Ronneberger et al. [98] and Allan et al. [5], the U-Net model was trained with an extensive annotated surgical image dataset provided by Intuitive Surgical Inc. and frozen.

It is worth noticing that while semantic segmentation eliminates environmental

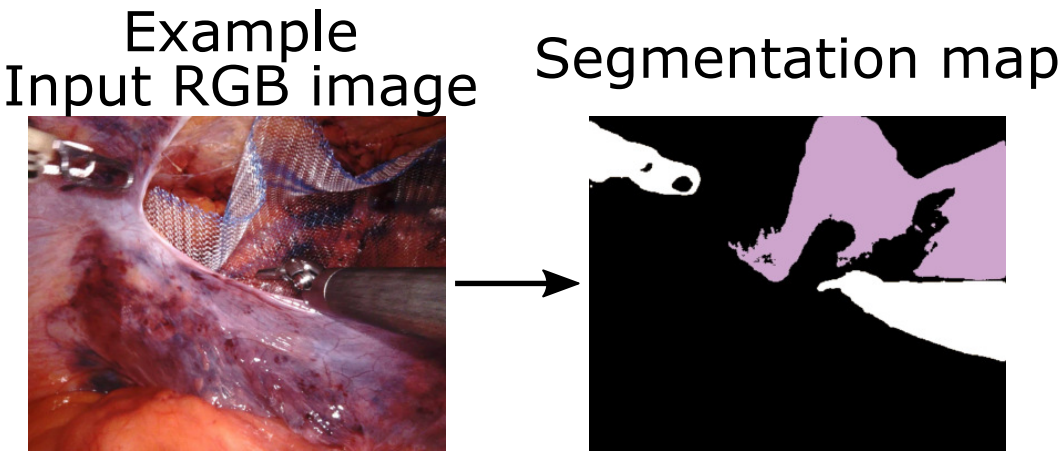

Figure 2.10: An RGB image of the endoscopic view during RAS and its segmentation map with three scene classes: surgical instruments (white), tissue (black), and others (purple).

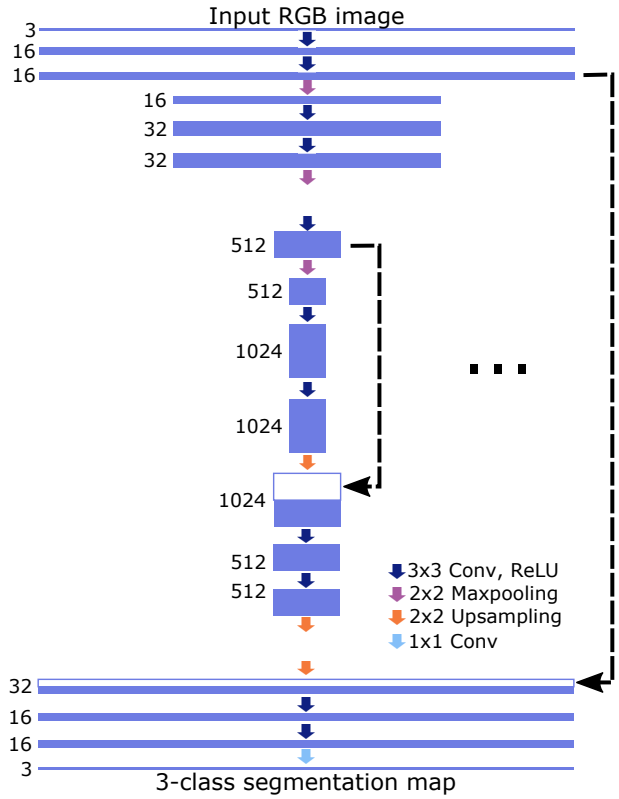

Figure 2.11: U-Net architecture for surgical scene semantic segmentation. Blue boxes represent feature maps with their number of channels. White boxes denote copied feature maps. Different operations are denoted by colored arrows.

noises and distractions, it cannot fully replace features extracted from the original endoscopic view as it is overly simplified at times. To harvest the benefits from both worlds, the semantic mask of the current endoscopic view could be concatenated with the original RGB image, forming a four-channel RGB-Mask image as the input for endoscopic vision feature extraction.

Robot kinematics data

While the endoscopic video data provides a rich representation of the current surgical scene, it is not the only useful data source for surgical state estimation. The current surgical stages and actions are also tightly correlated with the movements of the surgical instruments. The surgical robot's kinematics time series data is therefore another valuable data source for hierarchical surgical state estimation. As described in Sec. 2.2, kinematics data of da Vinci robots' MTMs and USMs was recorded and used in this work. Due to proprietary information protection reasons, original encoder readings from the robotic systems were separately processed by Intuitive Surgical Inc. and the manipulators' end-effector data was made available. As the robot kinematics data is significantly lower in dimensions comparing to the endoscopic video data, non deep learning-based pre-processing techniques were applied.

The kinematics variables available in all datasets used in this work include the 3-dimensional (3D) Cartesian positions (denoted by xyz), a rotational matrix (denoted by R), 3D linear velocities (denoted by $x'y'z'$), 3D angular velocities (denoted by $\alpha'\beta'\gamma'$), and the gripper angle (denoted by a). These variables are available for dVRK's two MTMs and two PSMs in the JIGSAWS suturing dataset. For RIOUS+ dataset and HERNIA-40 dataset, the same variables are available for dVXi's two MTMs and four USMs. To remove encoder recording errors like "sudden jumps" and smooth the time series data, outlier values in the kinematics time series data are removed if they differ considerably from their neighboring values.

System events data

In addition to the endoscopic video and surgical robot kinematics data, the RIOUS+ dataset and HERNIA-40 dataset also contains system events data from dVXi. These events are either binary or categorical, as described in Sec. 2.2. One-hot encoding was implemented for categorical system event data such as the surgical instrument type installed prior to surgical state estimation [130].

Chapter 3

MODELING ROBOT-ASSISTED SURGERY AS A HIERARCHICAL SYSTEM OF DISCRETE SURGICAL STATES

The field of medicine and surgery requires the highest level of safety precautions. Detailed standard-of-care for surgical procedures have therefore been highly standardized and documented for the training of robotic surgeons. From a temporal perspective, an RAS procedure can be viewed as a series of pre-determined and highly uniform surgical tasks and steps that the surgeon performs. For instance, the inguinal hernia repair surgery can be divided into 8 components as described in Table 2.3. Each of these components can, in turn, be divided into smaller temporal segments (as shown in Table 2.4). Just like an RAS procedure being viewed as a collection of surgical tasks and steps, many surgical tasks are also standardized and can be viewed as a collection of finer surgical actions and environmental observations. In Chapter 2, the JIGSAWS suturing dataset was described. The surgical task of suturing, for example, were broken down into actions as listed in Table 2.1. Surgical tasks in the RIOUS+ dataset and HERNIA-40 dataset also share this feature. This temporal hierarchy is observed in many RAS procedures.

As described in previous chapters, one crucial prerequisite for highly sought-after AI applications in RAS is a comprehensive awareness of the current surgical scene from a temporal perspective. For instance, the surgical system should be aware of the current surgical task performed for operating room scheduling, as the the current surgical task indicates the remaining time of the procedure. This level or temporal awareness can therefore aid applications such as operating room scheduling or surgeon evaluation [116]. The awareness of the fine-grained current surgical action such as putting down a needle, on the other hand, aids surgeon-assisting functionalities such as needle counting. While prior work in the field of surgical ontology [34] provided a method of standardizing the descriptions of surgical procedures, it is highly clinically-oriented and is not suitable for AI applications in RAS.

This temporal hierarchy observed during RAS guided us to model an entire RAS procedure as a hierarchical system of discrete surgical states that I refer to as the surgical Hierarchical Finite State Model (HFSM). Unlike the traditional definition of a finite state machine, however, our proposed modeling method is designed for

the propose of surgical state estimation and focuses within the scope of surgical applications. In the following sections, I first review the surgical ontology research efforts and its limitation. The details and definitions of our proposed RAS modeling strategy are then provided. The work presented in this chapter was described in publications [92, 93].

3.1 A review of surgical ontology

An ontology is defined as a formal specification of a shared conceptualization [39]. It is a form to represent knowledge in an interoperable manner [82]. Surgical ontology [34, 82] is a modeling methodology for surgical procedures. In 2016, the OntoSPM Collaborative Action was started with the goal of developing the ontology of the Surgical Process Models (SPM). Since then, surgical ontology development has standardized the description and modeling of various surgical procedures, scenarios, components, and equipment. It promoted a more consistent communications among researchers in the field of surgery, including RAS. Specifically, surgical ontology focuses on the construction of a database that standardizes the expressions and vocabulary of surgery-related information ranging from the surgical devices used during the procedure and the imaging techniques to the patient's medical history and conditions. This database should therefore be as comprehensive and inclusive as possible to widen surgical ontology's application. Fig. 3.1 shows a part of the surgical ontological description of a laparoscopic surgery.

Researchers have devoted enormous efforts to the enrichment of information of this knowledge base from all aspects of a surgery [34, 82, 97, 111]. Such information includes the anatomical condition of the patient, the operating room equipment and personnel, the surgical devices used during the surgery, imaging techniques, and many others. Ontology development efforts have found diverse applications in the field of medicine. The documentation and information processing of a surgery, for example, is ubiquitous throughout all stages of treatment of a patient. Surgical ontology provides a standardized method for this task, which is extremely valuable and beneficial for the patient's care. Surgical ontology also finds applications in the education and training of healthcare professionals, as it standardizes the languages and expressions for training and allows a smoother communication.

3.2 Modeling RAS as a hierarchical system of discrete surgical states

While surgical ontology remains an important method of modeling a surgical procedure, it is heavily clinically oriented for a smoother communication in the field of

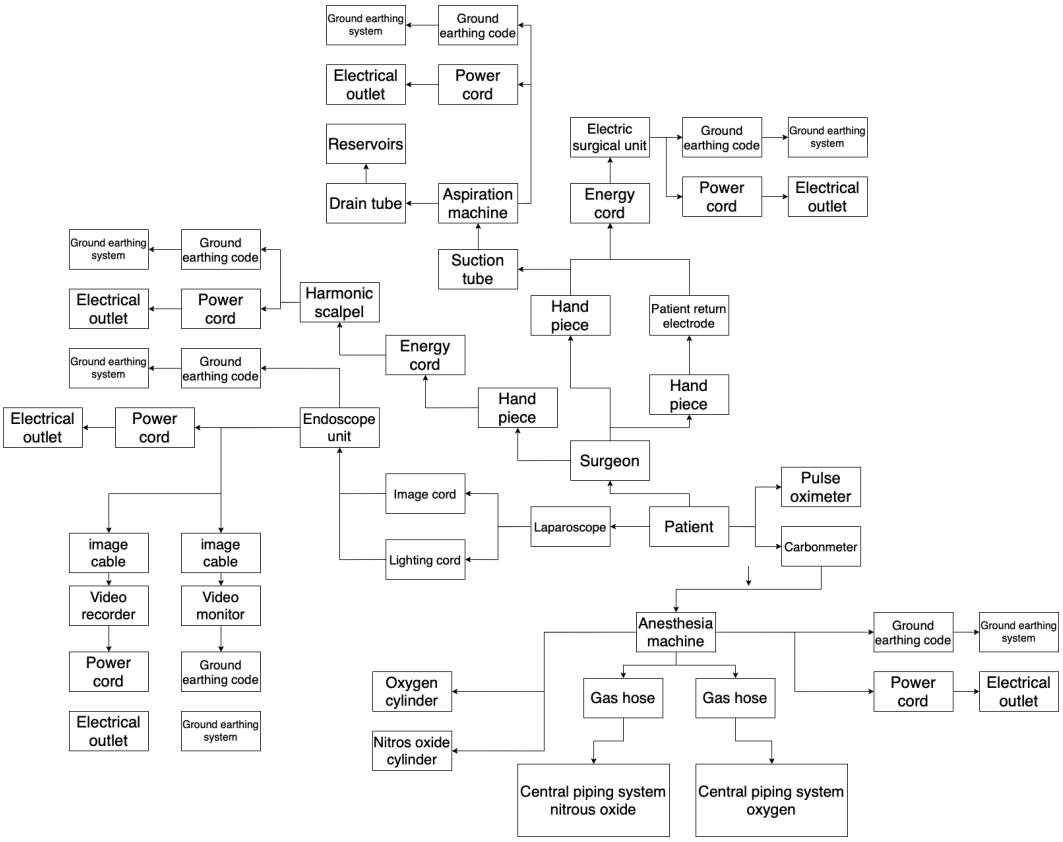

Figure 3.1: An example of a part of surgical ontological descriptions of a laparoscopic procedure.

medicine and is less desirable or suitable for AI applications in RAS. The purpose of surgical ontology is to document a surgery as complete as possible; therefore, high diversity and inclusiveness is highly desirable. Surgical ontology aims to describe a surgical procedure as comprehensively as possible, which includes not only the tasks and steps that occur during a surgical procedure, but all aspects of it. This includes diverse information ranging from the names of surgical devices used during the surgery to the pathologies of the patient.

This diversity in information, while extremely valuable in some fields of surgical research, is unnecessary and cumbersome for the development of AI applications in RAS. As shown in Fig. 3.1, the ontological description of a surgical procedure involves equipment details to the level of a power cord, which is unnecessary for AI applications related to surgical state estimation. Repeated entries are also frequently observed. For instance, Fig. 3.1 contains multiple entries of electrical outlets and ground earthing systems as many machines used during the surgery need them. These repeated entries, while inclusive, are cumbersome for temporal

perception applications. Additionally, as discussed in previous chapters, many AI-based surgeon-assisting functionalities requires the temporal awareness at multiple levels of granularity. While surgical ontology aims to describe the entire procedure comprehensively, it does not extend into the level of details from the temporal perspective required by some AI applications. For example, needle counting is a necessary task at the end of an RAS procedure to ensure that all needles used during the surgery are accounted for. A potential AI-based application is the autonomous bookmarking of the endoscopic video recording whenever a needle is released from the surgical instruments during the procedure such that the operating room staff could review the bookmarks for needle counting at the end of the procedure. This application requires the recognition of a fine-grained surgical state of releasing the needle. Whilst surgical ontology would document the types of needles used in the surgery, it does not record fine-grained surgical actions that last for seconds such as releasing the needle. Ontology is therefore not the most ideal modeling strategy for the AI applications during RAS, especially from a temporal perspective.

I therefore propose a holistic definition specifically designed for an RAS procedure as a hierarchical system of discrete surgical states at multiple levels of temporal granularity. I refer to this system as a surgical Hierarchical Finite State Model (HFSM), which is not to be confused with a traditional Finite State Machine in the field of automata theory [7]. While the HFSM is motivated by the HMM concept, it is distinct and simpler. Here, the definitions of a surgical HFSM and its components within the scope of consideration in this thesis work are provided.

Definition 1. A surgical state is the smallest temporal unit that makes up an RAS procedure. It is a surgical action, gesture, or environmental observation during an RAS.

Definition 2. A surgical superstate models coarser divisions of an RAS procedure into tasks, phases, or operative steps. It consists of surgical states or finer-grained surgical superstates.

Definition 3. A surgical Finite State Model (FSM) $M(S, \Sigma)$ is comprised of:

S: a finite non-empty set of surgical states s;

Σ: the input symbols (or data) to the system.

Definition 4. A surgical Hierarchical Finite State Model (HFSM) is comprised of a finite non-empty set of surgical superstates. These superstates are surgical

HFSMs themselves consisting of finer-grained surgical (super)states at lower levels of temporal granularity.

A surgical superstate could therefore be modeled as a surgical HFSM consisting of finer-grained surgical (super)states. For example, the surgical task of suturing is a surgical superstate during RAS. It consists of multiple actions and gestures such as reaching for the needle and pushing the needle through the tissue. Each of these actions is a surgical state. The suturing task superstate is, in turn, a surgical FSM that consists of finer-grained surgical states such as the ones mentioned above.

These definitions are similar to a typical (hierarchical) finite state machine; however, our current study does not incorporate transition dynamics between states. Unlike a traditional finite state machine, a surgical FSM does not put any constraint on the transition probabilities between surgical states due to the highly complex and dynamic nature of an RAS. Hence, a surgical HFSM is akin to a *hypergraph* [10, 29] that models the relationships between surgical states and hierarchy. Because our study dataset does not have any forbidden transitions between surgical states, in this thesis work I do not incorporate the potential impact of forbidden transitions between surgical states. In Fig. 2.5, the observed state transitions in the JIGSAWS suturing dataset and the RIOUS+ dataset were shown; however, all transitions are permitted despite not being observed in a specific dataset.

Comparing to surgical ontology, there are several advantages of modeling an RAS procedure as an HFSM. Surgical ontology does not include very fine-grained surgical actions and gestures that last for seconds. The knowledge of these fine-grained surgical states either during or after the surgery is crucial for applications such as needle counting and surgeon skill evaluation. Additionally, correlations can be observed between the estimation of surgical superstates and their finer-grained components in many cases, as seen in Tables 2.3 and 2.4. For instance, the surgical superstate of dissecting mesh pocket contains the fine-grained surgical state of a local energized cut, which is not a surgical state in the superstate of deploying mesh. The information of the current surgical state estimation result at one level of temporal granularity should therefore aid the surgical state estimation at other levels of temporal granularity. The HFSM of an RAS procedure would allow further explorations of such correlations and improve surgical (super)state estimation performances. Modeling an RAS procedure as an HFSM also eliminates unrelated information about the surgery included in surgical ontology, which makes it more suitable for AI applications in the field of surgical robotics.

Chapter 4

RECOGNITION OF FINE-GRAINED SURGICAL STATES WITH MULTIPLE DATA SOURCES

As defined in Chapter 3, a fine-grained surgical state is considered the smallest unit that makes up a surgical HFSM. Such surgical states usually last for a few seconds such as cutting or orienting a needle [24, 63], and could take the form of either a robot action or an observed change in surgical environment [96]. Tables 2.1, 2.2, and 2.4 summarized the fine-grained surgical states observed in datasets used in this work. While several efforts have been devoted to fine-grained surgical state estimation using diverse methods from probabilistic models to deep learning-based algorithms, the majority of them only utilized either the robot kinematics data [76, 78, 99, 113] or the endoscopic video data [49, 65] available in RAS datasets alone.

This chapter shows that the incorporation of various data types recorded by the robotic surgical system is beneficial to the fine-grained surgical state estimation. Through the development of a unified approach that incorporates various RAS data sources, I significantly improved both the accuracy and robustness of state-of-the-art fine-grained surgical state estimation models. As different data sources contains diverse information about the RAS procedure, they have their respective strengths and weaknesses in recognizing certain fine-grained surgical states, which supports the superior performance of our unified approach of state estimation. In the following sections, I describe in detail a model architecture for recognizing surgical states, and the components, implementation and training strategies of this model. The model is also evaluated on surgical data. The work presented in this chapter was described in the publication [96].

4.1 Model architecture

As both endoscopic video and robot kinematics time series data has respectively shown its effectiveness in fine-grained surgical state estimation in the past, various single-source state estimation models were implemented before the incorporation of state estimation results using these models. Our proposed approach (Fig. 4.1), denoted as Fusion-KVE, consists of four single-source state estimation models based on endoscopic vision, robot kinematics, and system events, respectively. The outputs are fed to a fusion model that makes a comprehensive inference. This

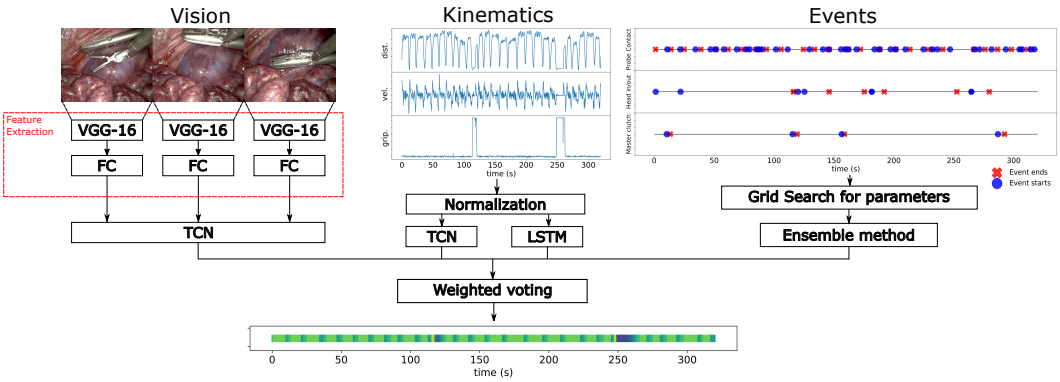

Figure 4.1: Fusion-KVE contains four single-input state estimation models receiving three types of input data. A fusion model that receives individual model outputs is used to make the comprehensive state estimation result.

section discusses each individual model as well as the fusion model which effectively combines the outputs of each model.

Spatial feature extraction of endoscopic video

As mentioned in previous chapters, while the endoscopic video time series data is a powerful and informative source for surgical state estimation that contains rich information about the current surgical scene, the high dimensionality of raw image data and the high level of surgical background nuisances requires an effective spatial feature extraction method prior to surgical state estimation. A VGG16 model was deployed [106] and maps each $224 \times 224 \times 3$ RGB endoscopic video frame $\boldsymbol{I}_t$ at time t to a vector $X^{vis} \in \mathbb{R}^{N_{vis}}$ where N is the number of features. The Feature Extraction component of Fig. 4.1 illustrates this process. The training of the VGG16 feature extractor started with network weights initialized with ImageNet pre-trained weights [59]. Guided by transfer learning, the weights were fine-tuned with one FC layer replacing the original top of the VGG16 model for surgical state estimation.

Temporal Convolutional Network-based state estimation model

A Temporal Convolutional Network (TCN) is a CNN architecture consisting of 1-dimensional (1D) convolutional layers on the temporal axis with the same input and output lengths [65]. In Fusion-KVE, a TCN is used to extract the temporal correlations among adjacent entries of time series data for state estimation. An encoder-decoder TCN framework was implemented, as shown in Fig. 4.2. The TCN model accepts time series features - whether extracted from endoscopic video or robot kinematics - as its input. At time step t, the input vector is denoted by X_t

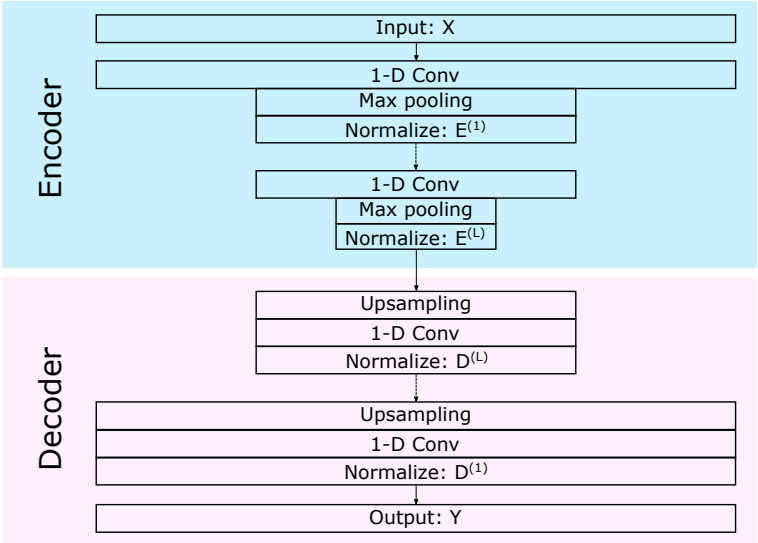

Figure 4.2: The encoder-decoder TCN network architecture with temporal convolutions, pooling, and upsampling operations to produce surgical state estimation Y from input features X.

for $0 < t \leqslant T$ where T is the length of the time series data vector.

The encoder component of the TCN model consists of L sets of 1D convolutional layers with pooling and normalization operations. For the l^{th} 1-D convolutional layer ($l \in \{1, ..., L\}$), F_l filters of kernel size k are applied along the temporal axis in order to capture the temporal progress of X_t. T_l is the number of time steps in the l^{th} layer. In each layer, the filters are parameterized by a weight tensor $W^{(l)} \in \mathbb{R}^{F_l \times k \times F_{l-1}}$ and a bias vector $b^{(l)} \in \mathbb{R}^{F_l}$. The raw output activation vector for the l^{th} layer at time t, $E_t^{(l)}$, is calculated from the normalized activation matrix from the previous layer $\hat{E}^{(l-1)} \in \mathbb{R}^{F_{l-1} \times T_{l-1}}$

$$E_t^{(l)} = ReLU(\boldsymbol{W}^{(l)} \hat{E}_{t:t+k-1}^{(l-1)} + \boldsymbol{b}^{(l)}), \tag{4.1}$$

where $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$ are learnable parameters, and the activation function is a Rectified Linear Unit (ReLU) [84]. A max pooling layer of stride 2 is applied after each convolutional layer in the encoder part such that $T_l = \frac{T_{l-1}}{2}$. The pooling layer is followed by a normalization layer, which normalizes the l^{th} activation vector at time t, $E_t^{(l)}$, using its highest value:

$$\hat{E}_t^{(l)} = \frac{E_t^{(l)}}{max(E_t^{(l)}) + \epsilon} \tag{4.2}$$

where $\epsilon = 10^{-5}$ is a small number to ensure non-zero denominators, and $\hat{E}_t^{(l)}$ is the normalized output activation vector.

The decoder component of the TCN model also contains L layers, but with an upsampling layer that repeats each data point twice, preceeding each temporal convolutional and normalization layer. The output vector is calculated and normalized in the same manner as the encoder part. The state estimation at time step t is done by a time-distributed FC layer with Softmax to normalize the logits.

As mentioned in previous chapters, surgical state estimation is analyzed for for both real-time and post-operative applications in this thesis work. For post-operative applications, the entire time series data is available to the model. During an RAS procedure; however, real-time state estimation model can only use the information from the current and preceding time steps. Data padding was therefore applied for the TCN model in a causal experimental setting. The temporal input with $\frac{k}{2}$ zeros on the left side before the convolutional layer and $\frac{k}{2}$ data points are cropped on the right side afterwards.

LSTM-based state estimation model

In addition to a TCN model, an LSTM model was also implemented to extract temporal features from the robot kinematics data. As introduced in Chapter 3, an LSTM model has no constraints on learning only from the nearby data on the temporal axis. Rather, it maintains a memory cell and learns when to read/write/reset the memory [33]. For real-time applications of surgical state estimation (causal setting), a unidirectional forward LSTM model was implemented, in which the model does not have access to data from future time steps and stores only the information from the preceding time steps. A bi-directional LSTM (biLSTM) model was also tested for post-operative applications of surgical state estimation (non-causal setting). The biLSTM model implemented in this work adds a *backward layer* in which information from time step $t + 1$ is used to calculate the states in the LSTM unit in time step t. The loss function for the LSTM model is the cross entropy between the ground truth and the predicted labels, and the stochastic gradient descent (SGD) is used to minimize loss.

Classification algorithm-based state estimation model

System events time series data from the dVXi system are available in the RIOUS+ dataset and HERNIA-40 dataset. These events take the form of either a binary events or a categorical event. These system events, such as surgeon head in/out and the usage of foot pedals on the surgical robotic system, are intuitively significantly lower in occurrence frequency and have extremely high spontaneity (as shown in

Table 4.1). Additionally, the occurrence frequencies of the same system event could differ significantly from a bench-top setting (in RIOUS+ dataset) to a real-world RAS setting (in HERNIA-40 dataset). F

To provide a comprehensive analysis of the proposed approach, various classification algorithms were implemented at each time step t that accepts system events at t as inputs, including Adaboost classifier, decision trees, Random Forest (RF), Ridge classifier, Support Vector Machine (SVM), and SGD [83]. Grid search over the hyperparameters of each model was performed and evaluated using the Area Under the Receiver Operating Characteristic Curve (ROC AUC) score [13]. The search process was iterated 200 times, with an early stopping criterion of score improvement under 10^{-6}. At each iteration, the best-performing model with replacement was recorded. The top three models that were selected most frequently were used in Fusion-KVE for surgical state estimation, and the final state estimation result is the mean of each model's prediction. The three top-performing models for the RIOUS+ dataset

System Events Occurrence Frequencies

RIOUS+	Mean frequency (event/s)
Surgeon head in/out of the SSC	5.1×10^{-3}
Camera follow	3.2×10^{-2}
Master clutch (left)	8.1×10^{-2}
Master clutch (right)	1.0×10^{-2}
Ultrasound probe activation	3.4×10^{-3}
Ultrasound probe in contact with tissue	4.7×10^{-2}
HERNIA-40	Mean frequency (event/s)
Energy pedal	4.7×10^{-6}
Surgeon head in/out of the SSC	1.8×10^{-4}
Camera follow	1.2×10^{-3}
Master clutch (left)	1.9×10^{-2}
Master clutch (right)	2.6×10^{-2}
Instrument change or install/uninstall (USM1)	8.2×10^{-5}
Instrument change or install/uninstall (USM2)	7.4×10^{-4}
Instrument change or install/uninstall (USM3)	4.6×10^{-4}
Instrument change or install/uninstall (USM4)	9.3×10^{-5}

Table 4.1: System events occurrence frequencies in the RIOUS+ dataset and HERNIA-40 dataset. The type of surgical instruments installed on each of the USMs of a dVXi is a categorical variable and changes its value when there is an instrument change on a USM; therefore, its occurrence frequency is the same as the frequency of instrument change or the installing/uninstalling of the surgical instrument on that USM.

are RF (n_{trees}=500, min_samples_split=2), SVM (penalty=$L2$, kernel=linear, λ=2), and RF (n_{trees}=400, min_samples_split=3). The three top-performing models for the HERNIA-40 dataset are RF (n_{trees}=200, min_samples_split=3), RF (n_{trees}=500, min_samples_split=3), SVM (penalty=$L2$, kernel=linear, λ=3).

The Random Forest (RF) algorithm implemented in Fusion-KVE, consisting of a collection of decision trees, is a popular machine learning algorithm for classification such as surgical state estimation [83]. During the training of an RF model, data are drawn randomly to build n_{trees} decision trees with its corresponding parameters. The classification output is based on majority voting of all decision trees. The RF model is more robust against overfitting and high data dimensionality than decision tree models, as multiple decision trees were created and fitted independently. Additionally, not all variables are available to an individual decision tree within the RF model, which increases randomness to model training. A Support Vector Machine (SVM) model is also widely used for classification. Tts core mechanism finds a hyperplane that best divides data into two classes [83]–the One-Versus-Rest (OVR) strategy was used in the multi-class problem of fine-grained surgical state estimation. An OVR strategy trains the SVM model to distinguish data from one category from the rest through the minimization of:

$$L_{SVM} = \lambda \|\mathbf{W}_{SVM}\|^2 + \frac{1}{n_{events}} \sum_{i=1}^{n_{events}} max(0, 1 - \mathbf{Y}_i(\mathbf{W}_{SVM}^T \mathbf{X}_i - \mathbf{b}_{SVM})), \quad (4.3)$$

where $\mathbf{W}_{SVM}$ and $\mathbf{b}_{SVM}$ are learnable parameters of the hyperplane, λ is the regularization parameter, and n_{events} is the number of system events recorded in a dataset. A linear kernel and L2 loss were found to be most suitable for system event-based fine-grained surgical state estimation.

Fusion model

As described in previous sections of this chapter, a unified approach that incorporates various RAS data sources using various fine-grained surgical state estimation models was proposed. These individual state estimation models have their respective strengths and weaknesses in recognizing certain states, since different states have inherent features that make them easier to be recognized by one type of data than the other(s). For instance, the *transferring needle from left to right* state in the JIGSAWS suturing dataset can be distinctly characterized by the sequential opening and closing of the left and right needle drivers which is captured by the robot kinematics data.

A weighted voting method was therefore used as the fusion model for all fine-grained surgical state estimation methods implemented. The fusion model accepts the state estimation result vector $\mathbf{Y}$ from all signal source estimators in order to generate a comprehensive state estimate. At time step t, let $\mathbf{Y}^{(t)} \in \mathbb{R}^{n_{models} \times n_{states}}$, where n_{models} is the number of models and n_{states} is the total number of possible states in a dataset. Row vector $\mathbf{Y}_{i,\cdot}^{(t)}$ is the output vector of the i^{th} model at time t and $\sum_{j=1}^{n_{states}} \mathbf{Y}_{i,j}^{t} = 1$. The overall probability for the system to be in the j^{th} state at time t - according to the models - is then

$$P_j^{(t)} = \sum_{i=1}^{n_{models}} \alpha_{i,j} \mathbf{Y}_{i,j}^{(t)} \tag{4.4}$$

where $\alpha_{i,j}$ is the weighting factor for the i^{th} model predicting the j^{th} state. The values of α are calculated from the diagnostic odds ratio (OR) derived from the model's accuracy in recognizing each state in the training data:

$$\alpha_{i,j} = \frac{TP_{i,j} \cdot TN_{i,j}}{FP_{i,j} \cdot FN_{i,j} + \epsilon} \tag{4.5}$$

where the (i, j)'s components of TP, TN, FP, FN are the number of true positives, true negatives, false positives, and false negatives of the i^{th} model on recognizing the j^{th} state, respectively. The parameter $\epsilon = 10^{-5}$ prevents the denominator from taking a zero value. α is normalized proportionally such that $\sum_{i=1}^{n_{models}} \alpha_{i,j} = 1$. The comprehensive estimate of state at time t $\mathbf{Y}_{fusion}^{(t)}$ is then made by

$$\mathbf{Y}_{fusion}^{(t)} = argmax_j P_j^{(t)}. \tag{4.6}$$

4.2 Implementation and training strategies

As shown in Fig. 4.1, the proposed model for fine-grained surgical state estimation contains $n_{models} = 4$ single-source estimators based on features extracted from the endoscopic video $\mathbf{X}^{vis} \in \mathbb{R}^{N_{vis}}$, robot kinematics $\mathbf{X}^{kin} \in \mathbb{R}^{N_{kin}}$, and system events $\mathbf{X}^{evt} \in \mathbb{R}^{N_{evt}}$. At time step t, each estimator's output is then $\mathbf{Y}^{(t)} \in \mathbb{R}^{n_{models} \times n_{states}}$, which is in turn the input to the fusion model which produced the comprehensive fine-grained surgical state estimation result $\mathbf{Y}_{fusion}^{(t)}$.

The vision-based state estimation model is a CNN-TCN model with VGG16 being the spatial CNN model. Spatial feature extraction was applied to the raw endoscopic video data in which a video frame $\mathbf{I}_t$ was used to extract $\mathbf{X}_t^{vis}$ at time step t. A grid search over parameters was performed to select hyperparameters in the CNN-TCN model. $N_{vis} = 1024$ was used for all three datasets during spatial

feature extraction. For the JIGSAWS suturing dataset and the RIOUS+ dataset, $L = 3$ with $F_l = \{32, 64, 96\}$ was used in the vision-based TCN model, whereas $L = 4$ with $F_l = \{16, 32, 64, 96\}$ were used for the HERNIA-40 dataset. As the mean duration of fine-grained surgical states in the three datasets varies, different kernel sizes were used in the vision-based TCN model for each dataset: $k = 6$ for the JIGSAWS suturing dataset and $k = 3$ for the RIOUS+ dataset and HERNIA-40 dataset, respectively. For training, I used the cross entropy loss with Adam optimization algorithm [55].

Two kinematics-based fine-grained surgical state estimation models were used: a TCN model and an LSTM model. The incorporation of both models better captures fine-grained states with various lengths of duration. An LSTM has no constraints on its ability to learn only from the nearby data on the temporal axis. Rather, it maintains a memory cell and learns when to read/write/reset the memory [33]. It has been shown that LSTM-based approaches exceed the state-of-the-art performance in longer-duration action recognition [24]. I therefore incorporated both TCN, which applies temporal convolution to learn local temporal dependencies, and LSTM, which is able to capture longer-term data structures. In the non-causal setting, a biLSTM model was used, whereas a forward LSTM model was used when the model was evaluated in the causal setting. In the causal setting, data padding of the TCN model was also applied.

The loss function for the LSTM model is the cross entropy between the ground truth and the predicted labels, and the stochastic gradient descent (SGD) is used to minimize this loss. The feature vector for the kinematics data $\mathbf{X}^{kin} \in \mathbb{R}^{N_{kin}}$ contains the Cartesian positions, rotation matrix, linear and angular velocities, and gripper angle for all PSMs in dVRK and all USMs in dVXi; therefore, $N_{kin} = 38$ for the JIGSAWS suturing dataset and $N_{kin} = 76$ for the RIOUS+ dataset and the HERNIA-40 dataset. A grid search over parameters was performed and $L = 2$ with $F_l = \{64, 96\}$ was used for the kinematics-based TCN model for all three datasets. For the LSTM model, I performed a grid search over the initial learning rate (0.1 to 1.0), the number of hidden layers (1 or 2), the number of hidden units per layer (256, 512, 1024, or 2048), and the dropout probability (0 or 0.5). The optimized set of parameters is 1 hidden layer with 1024 hidden units and 0.5 dropout probability for JIGSAWS, 512 hidden units for the RIOUS+ dataset, and 1024 hidden units for the HERNIA-40 dataset. The optimized initial learning rate is 0.1.

4.3 Model evaluation and result discussion

This technique was evaluated on both the JIGSAWS suturing dataset and the RIOUS+ dataset using *Leave One User Out* [32] in which the model was trained on all but one user and tested on the remaining users in each split. This method ensures that the estimation models are not overfitted to specific users. As HERNIA-40 dataset contains real-world RAS data, the surgeon identity information is not available for patient privacy protection reasons; therefore, a 5-fold cross validation evaluation method was used. As mentioned in previous chapters, surgical state estimation finds applications both during an RAS procedure and in post-operative analysis. I therefore adapted two experimental settings: causal and non-causal. In a causal setting, the models only have access to data from the current and preceding time steps. This approach mimics the real-time state estimation application of our model. In a non-causal setting, the models have access to data from the future time steps as well.

Given our goal of performing both real-time and post-operative fine-grained state estimation of the surgical task, two evaluation metrics were used: the *frame-wise classification accuracy* and the *edit distance*. The frame-wise classification accuracy is the percentage of correctly-recognized frames, which is measured without taking temporal consistency into account. This is because the model has only the knowledge of the current and preceding data entries in the real-time state estimation setting. The frame-wise classification accuracy was therefore used in both casual and non-causal settings. The edit distance, or *Levenshtein distance* [67], measures the number of operations needed to transform the inferred sequence of states in the segment level to the ground truth. The operations include insertion, deletion, and substitution. For instance, if the ground truth sequence is $G = [AAABBCC]$, then the ground truth sequence in the segment level is $\hat{G} = [ABC]$. An inferred sequence of states $P_1 = [AABBBCC]$ would then have an edit distance $L(G, P_1) = 0$ while an inferred sequence of states $P_2 = [AAACCCC]$ would have $L(G, P_2) = 1$. The edit distance was normalized, denoted as $\hat{L}(G, P)$, following [24, 63] using the maximum number of segment-level models. I computed the edit distance score using $(1 - \hat{L}(G, P)) \cdot 100$. As the calculation of edit distance requires information from the future time steps, it was only used for model evaluation in the non-causal experimental setting.

Fusion-KVE was evaluated and compared against the reported performances of several state-of-the-art fine-grained surgical state estimation methods in both causal and

non-causal experimental settings. When the reported performance is not available for a particular setting or a dataset was not available, the source code provided by the authors were used when available. An ablated version of our model (denoted as Fusion-KV) containing only the vision-based and kinematics-based models were used for evaluation for the JIGSAWS suturing dataset, as no system event is available in this dataset.

Table 4.2 compares the performances of the state-of-the-art surgical state estimation models with our models in the non-causal experimental setting. Table 4.3 assumes a causal experimental setting and the corresponding padding techniques and LSTM models were used. Fig. 4.3 shows an example of fine-grained surgical state estimation results of a trial from the RIOUS+ dataset in the causal setting. The state estimation results of the fusion model as well as their components are included and compared to the manually annotated ground truth (GT). Fig. 4.4 shows examples of the weight matrix α distributions used in the fusion models for both JIGSAWS suturing and RIOUS+ datasets. A large $\alpha_{i,j}$ indicates that the i^{th} model performs well in estimating the j^{th} state during training.

In Table 4.2, Fusion-KV achieves a frame-wise estimation accuracy of 86.3% and an edit distance score of 87.2 for the JIGSAWS suturing dataset, both improving the reported performances of state-of-the-art surgical state estimation models. The same improvements were observed for both RIOUS+ dataset and the HERNIA-40 dataset. In a causal setting in which the models only had information from the current and preceding time steps (Table 4.3), Fusion-KVE achieved a frame-wise accuracy of 89.4%, with an improvement of 11% comparing to the best-performing single-input model. Fusion-KV also achieves a higher accuracy comparing to single-input models.

A closer observation of the inferred state sequences by various models and their weighting factors as shown in Fig. 4.3 and Fig. 4.4 reveals the key contributions to the improvements offered by our unified approach and the use of multiple input data sources. Although kinematics-based state estimation models generally have a higher frame-wise accuracy comparing to vision-based models (Tables 4.2 and 4.3), which are very sensitive to camera movements, each model has its respective strengths and weaknesses in recognizing certain surgical states.

For instance, at around 200s of the illustrated sequence from the RIOUS+ dataset in Fig. 4.3, both kinematics-based models show a consecutive block of errors where the models fail to recognize the *probe released and in endoscopic view*

JIGSAWS Suturing dataset

Method	Input data type	Accuracy (%)	Edit Distance
ST-CNN[64]	Vision	71.3	59.9
TCN[65]	Vision	79.6	85.8
Forward LSTM[24]	Kinematics	80.5	75.3
TCN[65]	Kinematics	81.4	83.1
TDNN[78]	Kinematics	81.7	-
TricorNet[22]	Kinematics	82.9	86.8
biLSTM[24]	Kinematics	83.3	81.1
LC-SC-CRF[63]	Vision+Kinematics	83.5	76.8
Fusion-KV	**Vision+Kinematics**	**86.3**	**87.2**

RIOUS+ dataset

Method	Input data type	Accuracy (%)	Edit Distance
ST-CNN[64]	Vision	52.7	46.2
TCN[65]	Vision	55.9	41.7
Forward LSTM[24]	Kinematics	72.2	68.4
LC-SC-CRF[63]	Kinematics	72.8	61.9
TDNN[78]	Kinematics	79.1	-
TCN[65]	Kinematics	79.2	71.8
biLSTM[24]	Kinematics	82.4	71.0
Fusion-KV	Vision+Kinematics	85.5	73.8
Fusion-KVE	**Vision+Kinematics+Events**	**90.2**	**77.2**

HERNIA-40 dataset

Method	Input data type	Accuracy (%)	Edit Distance
Trivial	-	4.3	-
TCN[65]	Vision	41.2	37.4
TCN[65]	Kinematics	43.5	68.4
Forward LSTM[24]	Kinematics	48.8	55.0
biLSTM[24]	Kinematics	53.0	55.1
Fusion-KV	Vision+Kinematics	54.7	61.2
Fusion-KVE	**Vision+Kinematic+Events**	**59.9**	**65.5**

Table 4.2: Model performance comparison in a non-causal experimental setting.

JIGSAWS suturing dataset

Method	Input data type	Accuracy (%)
ST-CNN[64]	Vision	67.7
LC-SC-CRF[63]	Kinematics	68.1
TCN[65]	Vision	69.0
TDNN[78]	Kinematics	71.1
Forward LSTM[24]	Kinematics	74.7
Fusion-KV	**Vision+Kinematics**	**77.6**

RIOUS+ dataset

Method	Input data type	Accuracy (%)
ST-CNN[64]	Vision	46.3
TCN[65]	Vision	54.8
LC-SC-CRF[63]	Kinematics	71.5
Forward LSTM[24]	Kinematics	72.2
TDNN[78]	Kinematics	78.1
TCN[65]	Kinematics	78.4
Fusion-KV	Vision+Kinematics	82.7
Fusion-KVE	**Vision+Kinematics+Events**	**89.4**

HERNIA-40 dataset

Method	Input data type	Accuracy (%)
Trivial	-	4.3
TCN[65]	Vision	40.4
TCN[65]	Kinematics	41.1
Forward LSTM[24]	Kinematics	47.5
Fusion-KV	Vision+Kinematics	50.2
Fusion-KVE	**Vision+Kinematic+Events**	**57.6**

Table 4.3: Model performance comparison in a causal experimental setting.

state. Considering the relatively random robotic motions in this state, this is to be expected. The low weighting factors for both kinematics-based model in estimating this state, as shown in Fig. 4.4, also support this observation. On the other hand, the vision-based model correctly estimates this state, since the state is more visually distinguishable. When incorporating both vision- and kinematics-based methods, our fusion models perform weighted voting based on the training accuracy of each model. In this example, the weighting factor for the vision-based model is higher than the kinematics-based models; therefore, our fusion models are able to correctly estimate the current state of the surgical task. In other states where the robotic motions are more consistent but the vision data is less distinguishable (such as the *transferring needle from left to right* state), the kinematics-based models have higher weighting factors.

The incorporation of system events into the estimation process further improves the performance of our proposed approach. Comparing Fusion-KV's and Fusion-KVE's performance in Fig. 4.3, fewer errors are observed - many errors are corrected where α for the event-based model is high, such as states with shorter duration or frequent camera movements. At around 250s to 300s of the presented sequence of ultrasound imaging, frequent state transitions can be observed. Fusion-KVE more accurately estimate the states and shows fewer fluctuations, as compared to other models. The event-based model is less sensitive to environmental noises, as the events are collected directly from the surgical system. Additionally, when state transitions occur frequently, models that solely focus on the temporal dependencies of input data, such as TCN and LSTM, are less accurate. As the event-based model does not take temporal correlations into consideration, incorporating such data sources reduces the fluctuation in state estimation results, especially when the state transition is frequent or the duration of each state is short.

The average duration of fine-grained surgical states varies significantly, as shown in Section 2.2. To better capture states with different lengths of duration, two kinematics-based state estimation models were implemented: TCN and LSTM. Fig. 4.4 supports our decision. When the average duration of a state is long, the LSTM-based model has a higher weighting factor. The *sweeping* state (S7) in the RIOUS+ dataset, for example, has the average duration of 5.1s. The LSTM model has a higher α in the estimation of this state. S4 (Grasping probe), S5 (Lifting probe up), and S6 (Carrying probe to tissue surface) are more transient surgical states with shorter durations. The TCN-based model therefore has a higher weighting factor

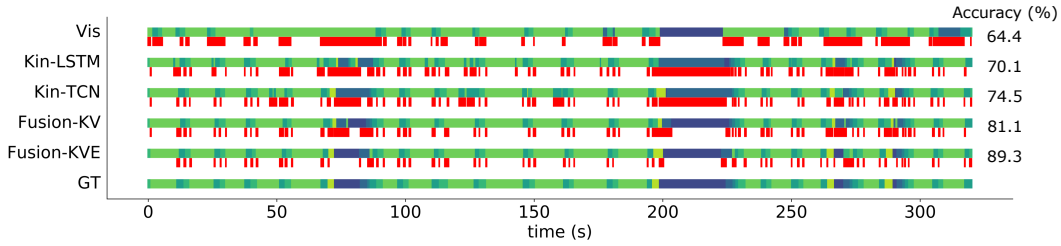

Figure 4.3: Example state estimation results of the vision-based model (Vis) and the kinematics-based models (Kin-LSTM and Kin-TCN) used in the fusion models, along with the ablated version of our model (Fusion-KV) and the full model (Fusion-KVE), comparing to the ground truth (GT). The top row of each block bar shows the state estimation results, and the frames marked in red in the bottom row are the discrepancies between the state estimation results and the ground truth.

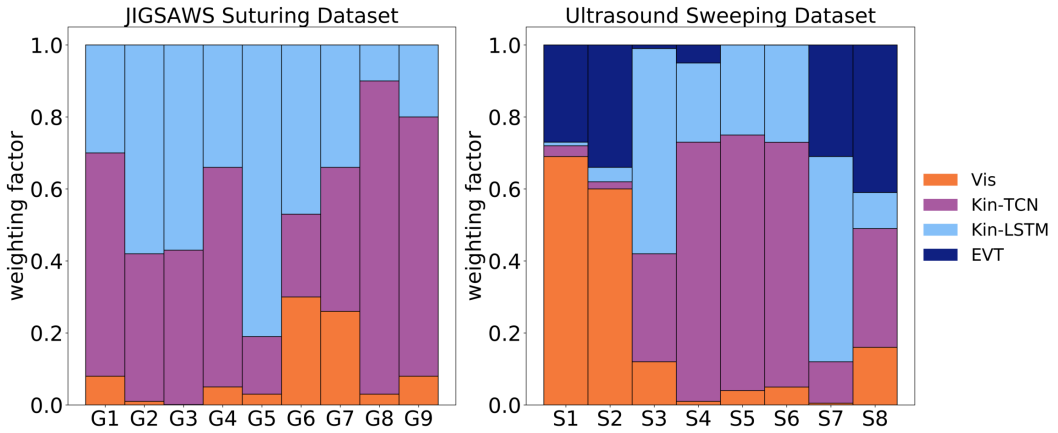

Figure 4.4: Example distributions of the normalized weighting factor matrix α for the JIGSAWS suturing task and the RIOUS+ imaging task in a causal setting. A larger weighting factor indicates that the model performs better at estimating the corresponding state.

for the estimation of them.

4.4 Conclusions

A unified approach of fine-grained state estimation for various surgical tasks during an RAS using multiple data sources improves the state-of-the-art estimation performance significantly. The model performance benefited significantly from the use of various data sources and diverse model architectures and algorithms. Evaluated on multiple RAS activity datasets including the JIGSAWS suturing dataset, the RIOUS+ dataset, and the HERNIA-40 dataset, our proposed model proved its robustness against complex and realistic surgical tasks by achieving a superior

frame-wise accuracy and edit distance score. The improvement is especially significant in a causal setting, where the model has knowledge of data obtained only in the current and preceding time steps. This is valuable for many real-time AI applications during RAS. Using the real-world RAS data in the HERNIA-40 dataset, I further showcased the robustness of Fusion-KVE by achieving a 10.1% frame-wise estimation accuracy improvement comparing to single-source models.

This Chapter showed how different types of input data (endoscopic video, robot kinematics, and system events) have their respective strengths and weaknesses in the recognition of fine-grained states. The fine-grained state estimation of surgical tasks is challenging due to the duration variability of different states and the frequent state transitions. By incorporating multiple types of input data, richer information associated with the current fine-grained surgical state can be extracted during the training process, which ultimately results more accurate estimation. To further improve the fine-grained state estimation accuracy, the weighting factor matrix can be used for boosting methods to more efficiently train the unified model.

Chapter 5

JOINT PREDICTION OF SURGICAL INSTRUMENT MOTIONS AND SURGICAL STATES

The implementation of autonomy in the field of surgical robotics, from passive functionalities such as virtual fixtures [103] to autonomous surgical tasks [88, 104], has attracted the attention of many RAS researchers. Such autonomy can enrich the manual teleoperation experience in robot-assisted surgeries (RAS) and assist the surgeons in many ways. Enhancements include automated changes in the user interface during surgery, additional surgeon-assisting system functionalities, and shared control or even autonomous tasks [15, 21, 54]. In 2016, Yang et al. proposed a definition of the levels of autonomy in medical robotics, ranging from mechanical robot guidance to fully autonomous surgical procedures [125], as shown in Fig. 5.1. The sensing of the user's desires plays an integral role, especially in Conditional autonomy.

One prerequisite for the autonomy applications mentioned above is the ability to anticipate the surgeon's intention and the robot's motions. Prediction of the robotic surgical instruments' trajectories, for instance, contributes to collision prediction and avoidance, including collisions between surgical instruments or with other obstacles in the proximity, such as delicate organ tissues. The advancement in autonomy would support applications to safe multi-agent surgical systems in which various surgical tasks are performed concurrently. Additionally, instrument trajectory prediction aids the process of human-computer interaction during an operation. Weede et al. presented an instrument trajectory prediction method for the optimal endoscope positioning through autonomous endoscopic guidance [120]. This prediction effort, however, took the form of a classifier with several pre-determined endoscope positions, which limited its application. The prediction of the future surgical states, either fine-grained states (picking up a needle) [2] or surgical phases (bladder dissection) [135], is useful in many surgeon-assisting features. Examples include the predictive triggering of cloud-based features or heavy-processing services which are inherently time consuming. This functionality provides a more seamless operational workflow. The prediction of the next surgical step or task also allows for more synchronized collaborations between the surgeons and operating room staff through workflow recognition [86, 109].

Deep learning-based methods for path and action prediction have been used in the field of computer vision, including path predictions using personal visual features and Long-Short Term Memory (LSTM) [57, 124] and early recognition of actions [70, 101]; however, prediction has received little attention in the field of surgical robotics. Outside of surgical applications, deep learning methods have predicted human paths and actions seconds in advance. Liang et al. recently proposed a multi-task model for predicting a person's future path and activities in videos using various features, including the person's position, appearance, and interactions [68]. Compared to human activity datasets (such as ActEV[8]) which are used for human path and activity predictions, RAS datasets enjoy the privilege of having synchronized robot kinematics, endoscopic vision, and system events as data sources. This is especially useful in the prediction of surgical states, since different sources of input data have their respective strengths and weaknesses in representing states with various kinematics and visual features. However, RAS can only observe a surgeon's intent via the movements of the input mechanisms and a few console events.

In Chapter 4, I proposed a unified model for surgical state estimation that incorporated multiple types of input data and exceeded the state-of-the-art state estimation performance [96]. Building on this work, the present chapter studies the task of concurrent instrument path and surgical state prediction via the use of multiple data streams and the incorporation of historic state transition sequences. This chapter proposes a model to jointly predict surgical instrument trajectories and fine-grained surgical states in RAS tasks. The method performs feature extraction and makes multi-step predictions of both the end-effectors' trajectories in the endoscopic reference frame and the future surgical states. I aimed for real-time predictions of up to 2 seconds in advance. Our model, denoted as daVinciNet, achieves accurate and robust real-time prediction performances. The following sections describe the model architecture and components of daVinciNet, its implementation and training strategies, and an evaluation of its performance. The work presented in this chapter was described in publications [94, 96].

5.1 Model architecture

An end-to-end joint prediction model (denoted as daVinciNet) that concurrently predicts the end-effector trajectories and the surgical states during an RAS procedure is proposed in Fig. 5.2. At time step t, daVinciNet extracts temporal features from each data source for an observation window with duration T_{obs}, and uses these features to make concurrent multi-step predictions of the instruments' end-effector

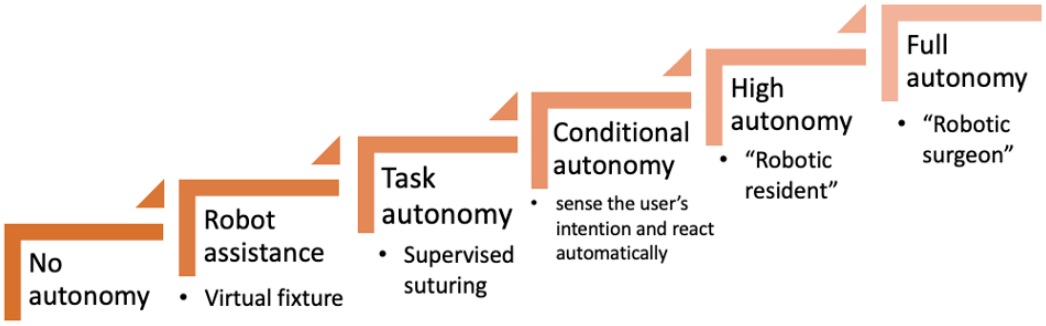

Figure 5.1: Six levels of autonomy in medical robotics. Example applications or analogies are listed for each level [125].

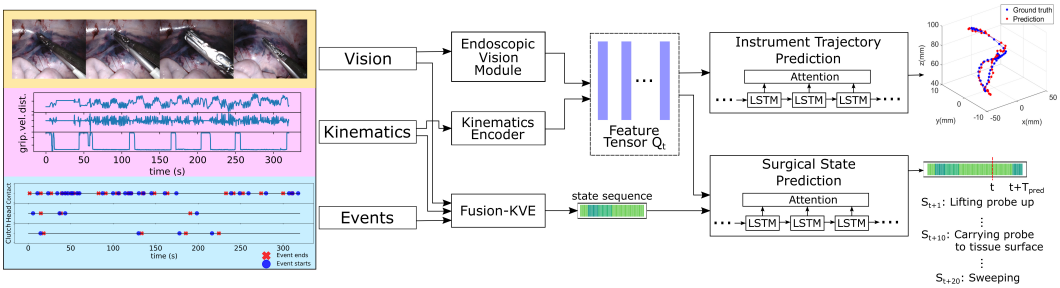

Figure 5.2: The daVinciNet model architecture. Given synchronized endoscopic video, robot kinematics and system events data streams, the model uses multiple encoders and Fusion-KVE to extract visual, kinematics, and states features. The concatenated feature tensor $\boldsymbol{Q}$ is used for both instrument trajectories and surgical state predictions. The state sequence, in addition to $\boldsymbol{Q}$, is a part of the input of the surgical state prediction model. Both prediction tasks rely upon an attention-based LSTM decoder . The example is shown with data sampled at 10 Hz.

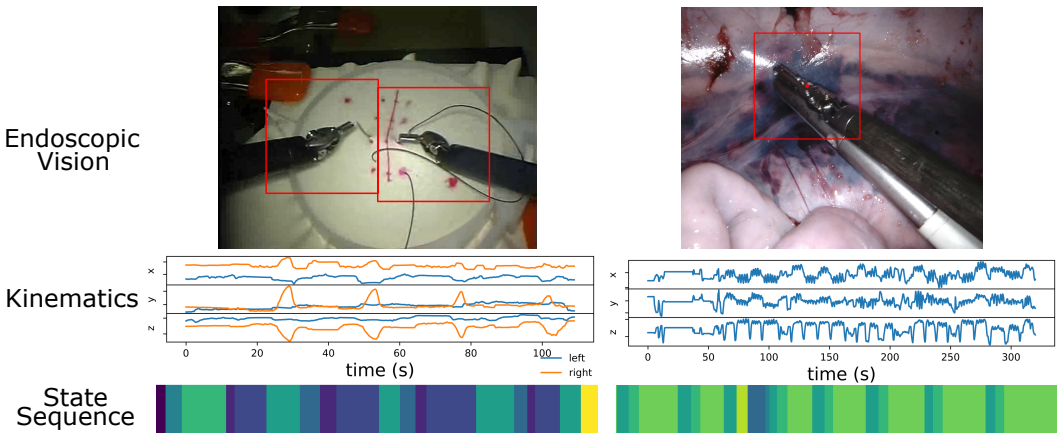

Figure 5.3: Samples of input data sources to daVinciNet.

3D Cartesian trajectories and a prediction of the fine-grained surgical states from $t + 1$ to T_{pred}, where T_{pred} is the number of prediction steps. More specifically, endoscopic video and robot kinematics features were used to construct a feature tensor $\boldsymbol{Q}_t$ for the end-effector trajectory prediction. Fusion-KVE - the fine-grained state estimation model described in Chapter 4 - was introduced in addition to $\boldsymbol{Q}_t$ for the fine-grained surgical state prediction. Samples of input data sources are shown in Figure 5.3. The prediction module resembles the structure of an encoder-decoder LSTM model with an attention mechanism. The following subsections first describe the feature extraction and prediction modules, and then the training details and strategies of the model.

Global and local vision feature extractions

The surgical background of endoscopic video data during RAS is complex and dynamic. While it contains rich information about the current surgical scene, a high level of noise and nuisance is inevitable; however, as seen in Tables 2.1, 2.2, and 2.4, most fine-grained surgical states are associated with surgeon actions which occur around the end-effectors of the surgical instruments in the endoscopic view. Additionally, the trajectory of the instruments, as seen by the user, should also be more correlated to the surrounding area of the end-effectors than other areas in the endoscopic view. It is therefore beneficial to extract both global and local visual features of the endoscopic video. The local visual features should focus on the visual areas surrounding the surgical instruments' end-effectors.

I therefore developed a novel endoscopic video feature extraction module that extracts visual features at both global and local levels. The global endoscopic vision feature at time step t was extracted following the same procedure in Section 4.1, in which a VGG16 model maps $\boldsymbol{I}_t \in \mathbb{R}^{224\times224\times3}$ to a vector $\boldsymbol{X}_t^{global} \in \mathbb{R}^{N_{global}}$. As daVinciNet aims to predict the fine-grained surgical state up to T_{pred} time steps into the future, the VGG16 model initialized with ImageNet pre-trained weights was fine-tuned with one FC layer for surgical state prediction at time step $t + T_{pred} - 1$.

Local endoscopic video features are extracted from the area in the endoscopic view around the surgical instruments' end-effectors. These areas are the Regions of Interest (RoIs) of an endoscopic view, and were determined with a silhouette-based surgical instrument tracking model as described in Section 2.3. The instrument tracking model was individually trained with an extensive RAS image dataset and frozen prior to the training of daVincinet. It provided the coordinates of bounding

boxes surrounding the end-effectors of all surgical instruments in the current endoscopic view. These bounding boxes were then the input to the CNN model as described in Section 2.3, which generates spatial RoI feature vector $\boldsymbol{X}_t^{RoI} \in \mathbb{R}^{N_{RoI}}$.

Temporal feature extraction via encoding

Instead of directly using the temporal concatenation of endoscopic vision CNN features or raw kinematics data as described in [64, 65, 126], I drew inspiration from vision-based human path prediction [3, 68] and implemented two LSTM-based encoders to capture the endoscopic video and robot kinematics time series data's temporal dependencies.

Two endoscopic video feature encoders extracted the global and local spatial vision features as described in the previous subsection, respectively. Given CNN features $\boldsymbol{X}_t^{global} = (\boldsymbol{x}_{t-T_{obs}+1}^{global}, ..., \boldsymbol{x}_t^{global})$ with $\boldsymbol{x}_t^{global} \in \mathbb{R}^{N_{global}}$, a forward LSTM encoder maps the hidden state $\boldsymbol{h}_t^{global}$ from $\boldsymbol{x}_t^{global}$ with:

$$\boldsymbol{h}_t^{global} = LSTM(\boldsymbol{h}_{t-1}^{global}, \boldsymbol{x}_t^{global}), \tag{5.1}$$

where $\boldsymbol{h}_t^{global}$ denotes the hidden state, and contains global temporal features from the endoscopic video at time step t. The concatenated encoder hidden states $\boldsymbol{H}_t^{global} = (\boldsymbol{h}_{t-T_{obs}+1}^{global}, ..., \boldsymbol{h}_t^{global}) \in \mathbb{R}^{T_{obs} \times n_{global}}$ forms a part of the feature tensor $\boldsymbol{Q}_t$. The dimension of the hidden states is denoted as n_{global}. Following the same method, another LSTM encoder generated $\boldsymbol{H}_t^{RoI} \in \mathbb{R}^{T_{obs} \times n_{RoI}}$ which contains RoI temporal features and is also a component of $\boldsymbol{Q}_t$. The dimension of the RoI features' hidden states is n_{RoI}.

To extract kinematics features from various types of surgical instruments' kinematics inputs (end-effectors' translational and rotational positions, etc.), and capture the long-term data structure, I followed the work of [91] and implemented an LSTM encoder with input attention to identify the importance of different driving time series data streams. At time t, the kinematics input to the kinematics LSTM encoder is $\boldsymbol{X}_t^{kin} = (\boldsymbol{x}_{t-T_{obs}+1}^{kin}, ..., \boldsymbol{x}_t^{kin})$, where $\boldsymbol{x}_t^{kin} \in \mathbb{R}^l$ and l is the number of kinematics variables. Instead of deriving $\boldsymbol{h}_t^{kin}$ directly from Eq. (5.1) as was done for the endoscopic video data, I constructed the input attention mechanism by learning a multiplier vector that represents the weights of each input series at time t from the previous hidden state $\boldsymbol{h}_{t-1}^{kin}$ and the LSTM unit's cell state $\boldsymbol{c}_{t-1}^{kin}$:

$$\boldsymbol{\beta}_t^i = softmax(u_e^T tanh(\boldsymbol{W}_e(\boldsymbol{h}_{t-1}^{kin}, \boldsymbol{c}_{t-1}^{kin}) + \boldsymbol{V}_e \boldsymbol{x}^{kin,i})), \tag{5.2}$$

where $\boldsymbol{x}^{kin,i} \in \mathbb{R}^{T_{obs}}$ is the i-th kinematics input series ($1 \leq i \leq l$). u_e, $\boldsymbol{W}_e$ and $\boldsymbol{V}_e$ are learnable parameters. The weighted kinematics input at time t is then:

$$\tilde{\boldsymbol{x}}_t^{kin} = \sum_{i=1}^{l} \beta_t^i \boldsymbol{x}_t^{kin,i}, \tag{5.3}$$

which substitutes $\boldsymbol{x}_t$ in the kinematics feature encoder. The encoded hidden states $\boldsymbol{H}_t^{kin} = (\boldsymbol{h}_{t-T_{obs}+1}^{kin}, ..., \boldsymbol{h}_t^{kin}) \in\in \mathbb{R}^{T_{obs} \times n_{kin}}$ is the final component of feature tensor $\boldsymbol{Q}_t$.

Instrument trajectory and surgical state predictions via decoding

After encoding, a feature tensor $\boldsymbol{Q} = (\boldsymbol{q}_{t-T_{obs}+1}, ..., \boldsymbol{q}_t) \in \mathbb{R}^{T_{obs} \times (n_{global}+n_{RoI}+n_{kin})}$ is available. LSTM decoders were implemented to predict the 3D Cartesian instrument paths in the endoscopic reference frame after up to T_{pred} steps ahead ($\boldsymbol{y}_t \in \mathbb{R}^{T_{pred} \times 3}$) and future states ($s_t \in \mathbb{R}^{T_{pred}}$), respectively.

As described in Section 2.3, despite its improvement from simple RNN models, an LSTM model still suffers from performance deterioration as the input sequences' length increases. Temporal attention [9] was introduced to alleviate such deterioration. The additive attention mechanism described in Section 2.3 was implemented in daVinciNet's LSTM decoders. The attention mechanism allows the decoders to use relevant hidden states among all time-steps from $\boldsymbol{Q}$ in an adaptive manner. At time step t, the temporal attention weights $\boldsymbol{\alpha} \in \mathbb{R}^{T_{obs}}$ was extracted with the method described in Section 2.3 and Eq. 2.3. The weighted feature $\tilde{\boldsymbol{q}}_t = \sum_{j=1}^{T_{obs}} \alpha_t^j \boldsymbol{q}_t^j$ and the historic target sequences (3-D end-effector path $\boldsymbol{y}$ or estimated surgical state s) from $t - T_{obs} + 1$ to t were used to extract the target embedding following [68]:

$$\tilde{\boldsymbol{y}}_{t-1} = \boldsymbol{W}_d(\tilde{\boldsymbol{q}}_{t-1}, \boldsymbol{y}_{t-1}) + \boldsymbol{V}_d, \tag{5.4}$$

where variables $\boldsymbol{W}_d$ and $\boldsymbol{V}_d$ are learned. The update of the decoder hidden state $\boldsymbol{d}_t$ is:

$$\boldsymbol{d}_t = LSTM(\boldsymbol{d}_{t-1}, [\tilde{\boldsymbol{y}}_{t-1}, \tilde{\boldsymbol{q}}_t]), \tag{5.5}$$

after which the end-effector trajectory predictions $\hat{\boldsymbol{y}}_t$ are computed by a fully connected (FC) layer.

The probability vector $\boldsymbol{s}$ for surgical state prediction can be similarly derived, and the state prediction $\hat{s}_t \in \mathbb{R}^{T_{pred}}$ is the future state sequence, with each state having the maximum likelihood among all states at each time-step. The historic sequence of surgical state ($s_{t-T_{obs}+1}$) was used in addition to $\boldsymbol{Q}_t$ for the fine-grained surgical state prediction. Fusion-KVE from Chapter 4 - a unified surgical state estimation model

- replaced the ground truth (GT) state sequence. This approach enables real-time applications for daVinciNet.

During an RAS procedure, the surgical state prediction model does not have access to the manually-labeled historic surgical state sequence; therefore, a state estimation model must provide the historic state sequence. The accuracy of the state estimation model therefore affects the fine-grained surgical state prediction performance. By incorporating Fusion-KVE, which outperformed state-of-the-art fine-grained surgical state estimation methods, the negative effect on prediction performances caused by state estimation errors is reduced.

5.2 Implementation and training strategies

The training of the global endoscopic video feature extractor followed the same strategies as described in Section 4.2, with $N_{global} = 1024$ CNN features extracted. $\boldsymbol{I}_t$ was the input to the instrument tracking model to determine the RoIs (area surrounding the end-effectors of all surgical instruments) of the current endoscopic view. Specifically, the CNN architecture for RoI feature extraction consisted of two convolutional layers with ReLU [84] activation. $N_{RoI} = 100$ CNN features were extracted. Both the global vision feature encoder and the local vision encoder had $n_{global} = n_{RoI} = 32$ hidden states. The kinematics encoder had $n_{kin} = 32$ hidden states.

Both the trajectory prediction decoder and the surgical state prediction decoder were implemented with 96 hidden states after a grid search for parameters. $r = 6$ variables (3D end-effector paths for two PSMs) were predicted for the JIGSAWS data set, while $r = 12$ variables were predicted for the RIOUS+ dataset (3D end-effector paths for four USMs). Multi-step predictions were implemented, with $T_{obs} = 20$ and $max(T_{pred}) = 20$ for data streaming at 10Hz.

The trajectory loss function is the cumulative L_2 loss between the predicted end-effector trajectory and the ground truth trajectory, summed up from $T_{obs} + 1$ to T_{pred}:

$$L_{traj} = \sum_{t=T_{obs}+1}^{T_{pred}} (\hat{\boldsymbol{y}}_t - \boldsymbol{y}_t)^2, \tag{5.6}$$

which includes the prediction accuracy of multiple time steps into the future. The state estimation loss function is the cumulative categorical cross-entropy loss that accounts for the discrepancies between the predicted surgical states and the ground

truth:

$$L_{state} = \sum_{t=T_{obs}+1}^{T_{pred}} -log(\frac{e^{\hat{s}_t}}{\sum_{i=1}^{n_{state}} e^{s_{t,i}}}), \tag{5.7}$$

where n_{state} is the total number of fine-grained surgical states. daVinciNet was trained end-to-end with the goal of minimizing a loss function that accounts for both the trajectory prediction accuracy and state prediction accuracy:

$$L = \rho L_{traj} + (1 - \rho) L_{state}, \tag{5.8}$$

where ρ weights the trajectory loss and surgical state loss functions. $\rho = 0.5$ was used during the training of the results described below.

5.3 Model performance and results discussions

daVinciNet's performance was evaluated with the JIGSAWS suturing dataset [32] and the RIOUS+ dataset [96] with the *Leave One User Out* method for validation [32]. Multi-step end-effector trajectory and fine-grained surgical state predictions were performed for various time spans in order to realize a comprehensive model performance evaluation. Specifically, the model performance was evaluated with T_{pred} ranges from 0.1s to 2s with a 0.1s increment. Ablation studies [79] were also performed to identify the contributions of various types of features used in both prediction tasks. To the best of our knowledge, there is no current benchmark for surgical instrument trajectory or surgical state predictions. daVinciNet was therefore compared with its own ablated versions to showcase its robustness and the necessity of its architectural and design components. In the following subsections, the evaluation metrics and model performances are described and discussed.

Evaluation metrics and model performances

The end-effector trajectory predictor and surgical state predictor were evaluated separately. Three types of metrics were used to evaluate the accuracy of daVinciNet's end-effector trajectory prediction: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) [90, 91]:

$$\begin{aligned} RMSE &= \sqrt{\frac{\sum_{i=1}^{N}(y^i - \hat{y}^i)^2}{N}} \\ MAE &= \frac{\sum_{i=1}^{N}|y^i - \hat{y}^i|}{N} \\ MAPE &= \sum_{i=1}^{N}\left|\frac{y^i - \hat{y}^i}{y^i}\right| \times 100\%. \end{aligned} \tag{5.9}$$

Since RMSE and MAE are independent of the variables' absolute values, they provide an intuitive comparison among variables in the same dataset. MAPE calculates the percentage error; therefore, it provides a direct comparison between prediction accuracies across different datasets. To evaluate the surgical state prediction accuracy, I calculated the percentage of accurately predicted frames in the testing sequences following [65, 96]. Both trajectory and state predictions are performed in a multi-step manner for up to 2-second into the future ($max(T_{pred}) = 20$ for 10Hz data streams). For each prediction time-step, the evaluation metrics are based only on the prediction at that time-step, without accounting for the errors in previous prediction steps.

As an example, daVinciNet's end-effector trajectory prediction and surgical state prediction performance when the prediction time-step is 1 second ($T_{pred} = 10$) is shown in Tables 5.1 and 5.2, respectively. Fig.5.4 and Fig.5.5 illustrate how the model performance changes at various prediction time-steps from 0.1s to 2s. Both figures and tables also include the performance of ablated versions of daVinciNet as compared to the performance of the unablated model. Fig. 5.6 shows a sample surgical state sequence of the ultrasound imaging task in the RIOUS+ dataset. Surgical state prediction results when $T_{pred} = 10$ using daVinciNet as well as ablated versions of it were compared to the manually annotated ground truth.

Discussion

Table 5.1 compares the differences in end-effector path prediction accuracy when daVinciNet uses only kinematics features $\boldsymbol{H}^{kin}$, global endoscopic vision and kinematics features $\{\boldsymbol{H}^{global}, \boldsymbol{H}^{kin}\}$, and a full feature tensor $\boldsymbol{Q} = \{\boldsymbol{H}^{global}, \boldsymbol{H}^{RoI}, \boldsymbol{H}^{kin}\}$. The accuracy of end-effector trajectory prediction in the endoscopic reference frame was evaluated, along with the end-effector distance $d = \sqrt{x^2 + y^2 + z^2}$ from the origin (camera tip). daVinciNet predictions based on all data streams consistently achieve up to 20% better performance. Clearly, endoscopic video features contribute to better prediction of the end-effector trajectory in the endoscopic reference frame. Many instrument movements have advanced visual cues, e.g., suture pulling usually occurs after the needle tip has appeared on the suturing pad or tissue. Visual features, such as the distance between the end-effector and nearby tissue, also help in the prediction of trajectory changes. Therefore, including visual features, especially RoI information about the surrounding area of end-effectors, is helpful to the trajectory prediction task.

JIGSAWS Suturing dataset

		x_1	y_1	z_1	d_1	x_2	y_2	z_2	d_2
$\boldsymbol{H}^{kin}$	RMSE	2.81	2.42	3.28	4.16	3.8	4.26	4.75	5.92
	MAE	2.19	1.95	2.86	3.7	3.42	3.91	4.31	5.34
	MAPE	6.8	6.09	7.39	8.93	7.77	8.03	8.2	10.14
$\{\boldsymbol{H}^{global}, \boldsymbol{H}^{kin}\}$	RMSE	2.7	2.29	3.25	4.01	3.65	4.01	4.63	5.2
	MAE	2.17	1.88	2.79	3.5	3.15	3.7	4.16	4.76
	MAPE	6.73	5.88	7.18	8.44	7.05	7.5	7.91	9.27
$\boldsymbol{Q}$	RMSE	2.53	1.89	2.96	3.35	3.15	3.5	3.91	4.51
	MAE	2.07	1.51	2.46	3.09	2.78	3.06	3.5	4.17
	MAPE	6.43	4.72	6.35	7.46	6.13	6.11	6.67	7.95

RIOUS+ dataset

		x	y	z	d
$\boldsymbol{H}^{kin}$	RMSE	1.67	1.8	1.22	2.3
	MAE	1.45	1.62	1.24	2.06
	MAPE	1.89	2.62	1.76	2.17
$\{\boldsymbol{H}^{global}, \boldsymbol{H}^{kin}\}$	RMSE	1.67	1.7	1.18	2.1
	MAE	1.33	1.52	1.12	1.91
	MAPE	1.7	2.43	1.57	2.01
$\boldsymbol{Q}$	RMSE	1.23	1.41	1.08	1.98
	MAE	1.09	1.34	0.97	1.64
	MAPE	1.31	2.16	1.1	1.72

Table 5.1: End-effector trajectory prediction performance measures when predicting one second ahead ($T_{pred} = 10$). The prediction performances for the Cartesian end-effector path in the endoscopic reference frame (x, y, z) and $d = \sqrt{x^2 + y^2 + z^2}$ are compared when the trajectory prediction decoder uses only kinematics features ($\boldsymbol{H}^{kin}$), uses global endoscopic video and kinematics features ($\{\boldsymbol{H}^{global}, \boldsymbol{H}^{kin}\}$), and uses global endoscopic video, RoI, and kinematics features ($\boldsymbol{Q}$).

Input data	JIGSAWS Suturing dataset (%)	RIOUS+ dataset (%)
Q only	64.11	65.44
Fusion-KVE only	75.08	76.5
Q+Fusion-KVE	**84.3**	**91.02**

Table 5.2: Surgical State Prediction performance when predicting one second ahead ($T_{pred} = 10$). Prediction performance is compared when the state prediction decoder uses only the feature tensor ($\boldsymbol{Q}$ only), only the historic state sequence (Fusion-KVE only), and both ($\boldsymbol{Q}$+Fusion-KVE) as its input data source.

Table 5.2 investigates the surgical state prediction accuracy with only $\boldsymbol{Q}$, only the historic state estimation results, or both as input features. The significant improvement in state prediction accuracy that arises from the incorporation of both data sources supports our model design and data selection. As mentioned in the previous section, daVinciNet does not have access to the manually annotated ground truth state sequence in real-time prediction. The high prediction accuracy using an estimated state sequence also shows the robustness of our model in real-time state prediction. It also serves as evidence of the accurate state estimation performance of Fusion-KVE.

Fig.5.4 shows how the surgical instruments' end-effector trajectory prediction accuracy changes with increasing prediction time-steps. I compared the trajectory prediction MAE when the feature tensor used for trajectory prediction includes only $\boldsymbol{H}^{kin}$, $\{\boldsymbol{H}^{global}, \boldsymbol{H}^{kin}\}$, or all three types of features, $\boldsymbol{Q}$. For the JIGSAWS suturing dataset, the MAE of the left (d_1) and right (d_2) instruments were averaged and shown. The use of RoI endoscopic video features consistently decreases the trajectory prediction MAE and improves the model performance. This trend is especially significant at larger prediction steps. This observation reaffirms the earlier discussion that the visual features and advanced visual cues concentrated in RoIs were detected by the RoI feature encoder and contributed to the end-effector trajectory prediction.

Fig.5.5 shows the progress of surgical state prediction accuracy as the prediction time-step increases. For fine-grained surgical state prediction, I compared daVincinet's performance when the state prediction decoder is based only on $\boldsymbol{Q}$, only on the historic state sequence estimated by Fusion-KVE, and both. The ablated prediction models were constructed with the $\tilde{\boldsymbol{q}}$ or $\boldsymbol{y}$ term omitted from Eq. 5.5, respectively. When the feature tensor $\boldsymbol{Q}$ was the only available data source to the surgical state prediction decoder, the state prediction accuracy is constantly the lowest and exponentially decreases as the prediction time-step increases. Due to the absence of the historic target series (the historic surgical state sequence), the state prediction model could only rely on predicted state sequences from previous time-steps to make the next prediction. This caused compounding errors, as previous prediction errors affected the next state prediction with little correction.

When the state prediction decoder incorporates the historic state estimation sequence generated by Fusion-KVE, the surgical state prediction accuracy was improved, especially at large prediction steps, since the target series provided corrections to

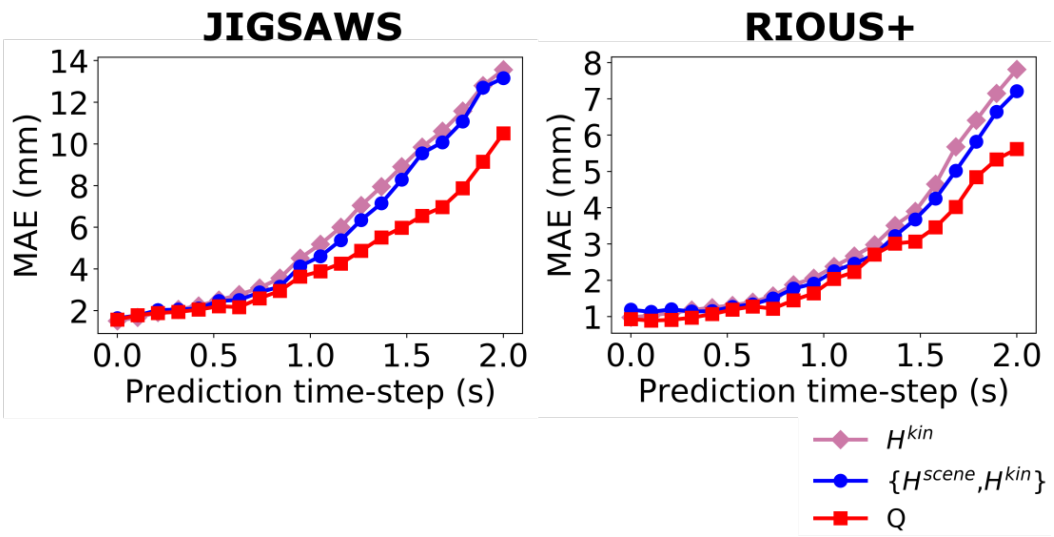

Figure 5.4: Comparisons of model performance as different features are included for the end-effector trajectory prediction at various prediction time steps. daVinciNet was constructed with only kinematics features ($\boldsymbol{H}^{kin}$), global endoscopic video features and kinematics features ($\{\boldsymbol{H}^{global}, \boldsymbol{H}^{kin}\}$), and global endoscopic video, RoI, and kinematics features ($\boldsymbol{Q}$). $mean(MAE_{d_1} + MAE_{d_2})$ and MAE_d were plotted for the JIGSAWS suturing dataset and the RIOUS+ dataset, respectively.

previous prediction errors. As shown by the black curve in Fig. 5.5, the surgical state predictions were performed without advanced visual or kinematic cues that indicate state changes from the feature tensor $\boldsymbol{Q}$. The surgical state predictor could therefore only depend on the target series with no additional inputs. The improvement in prediction accuracy was therefore limited. By incorporating both the feature tensor $\boldsymbol{Q}$ and the historic surgical state sequence from Fusion-KVE, advanced cues from visual and kinematics features were used to forecast state changes, and the historic state sequence provided corrections to prediction errors. The full daVinciNet model therefore achieves the highest fine-grained surgical state prediction accuracy that is significantly maintained as prediction time-step increases.

The sample sequence of ultrasound imaging fine-grained surgical state prediction results seen in Fig. 5.6 further supports the inclusion of both the feature tensor $\boldsymbol{Q}$ and Fusion-KVE output for the surgical state prediction. Prediction errors occur in blocks when only $\boldsymbol{Q}$ is used for prediction due to uncorrected errors from using predicted state sequences from previous time-steps. A model using only the historic state sequence showed fewer errors in consecutive time blocks; however, the missing input of endoscopic visual or kinematics cues in the feature tensor $\boldsymbol{Q}$ led to delayed responses to state changes, or even the missing of states with relatively shorter dura-

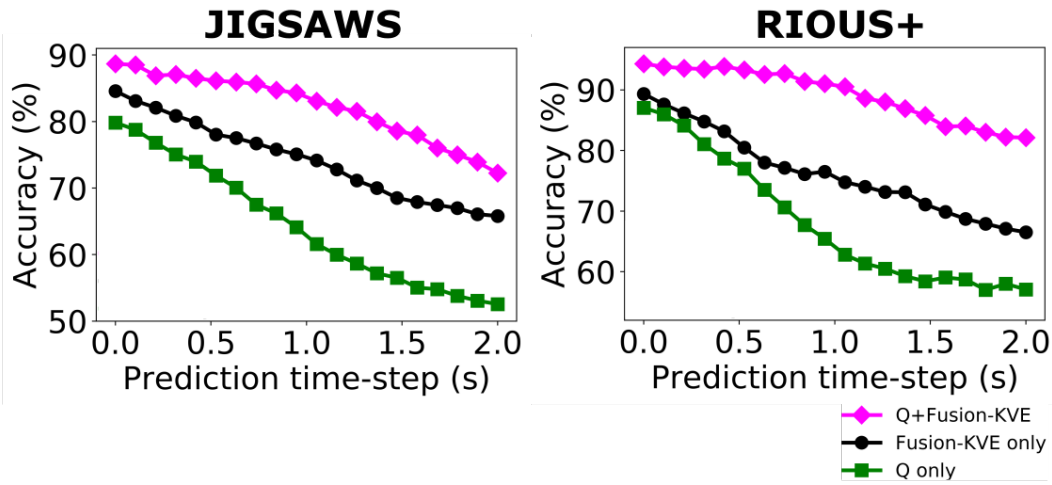

Figure 5.5: Model performance comparison when different features are included for the surgical state prediction at various prediction time steps. The state prediction decoder was constructed with only the feature tensor ($\boldsymbol{Q}$ only), only the historic fine-grained surgical state sequence (Fusion-KVE only), and both ($\boldsymbol{Q}$+Fusion-KVE) were plotted for the JIGSAWS suturing dataset and the RIOUS+ dataset, respectively.

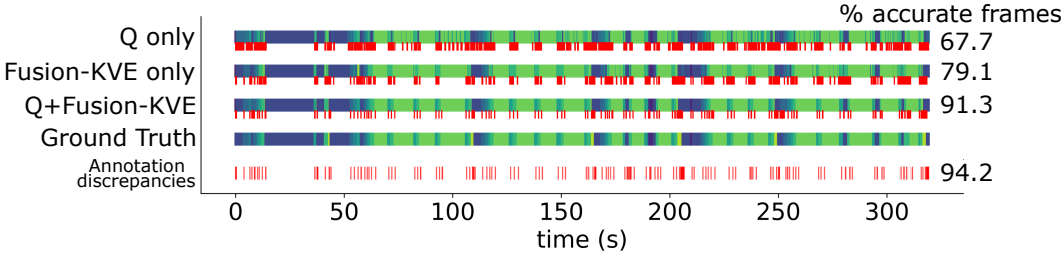

Figure 5.6: A sample surgical state sequence from the RIOUS+ dataset and the 1-second state prediction results using only the feature tensor (**Q** only), only the historic state sequence (Fusion-KVE only), both (**Q**+Fusion-KVE), and the manually annotated ground truth. Each block bar contains the state prediction results when $T_{pred} = 10$ (top). The discrepancies between the prediction results and the ground truth are shown in red. The annotation discrepancies row (bottom) shows the locations of frames where multiple annotators used different state labels, with the mean matching rate of 94.2% among annotators.

tion. The model that incorporated both $\boldsymbol{Q}$ and and the Fusion-KVE estimation result sequence significantly improved the state prediction accuracy, with the remaining errors located mostly around state transitions.

Additionally, since the temporal annotations of surgical states were done manually by humans in both the JIGSAWS suturing dataset and the RIOUS+ dataset, I investigated the annotation variance in the ground truth state sequence introduced by human annotators. Five users were asked to annotate the sample sequence in Fig. 5.6 frame-by-frame with fine-grained surgical states in Table 2.2. The discrepancies among annotations are shown in the bottom row of Fig.5.6, with an average matching rate of 94.2% among annotators. Even human annotators cannot agree perfectly on a state sequence: their discrepancies occur mostly at state transitions. These discrepancies are expected, as the transition from one state to another in a surgical subtask is not abrupt, but gradual. Hence, annotators may identify different video frames as a state transition. The fine-grained state prediction errors of daVinciNet can therefore be partially attributed to human annotation errors in identifying the exact state transition times. Thus, daVinciNet's robustness is further established by its high state prediction accuracy even in the presence of noise in the ground truth data.

5.4 Conclusions

This Chapter focused on the real-time prediction of variables that are crucial to AI applications in RAS, including the surgical instruments' end-effector trajectories in the endoscopic viewing frame and the future fine-grained surgical states. I proposed daVinciNet: a unified end-to-end joint prediction model that uses synchronized sequences of robot kinematics, endoscopic vision, and system events data as input to concurrently predict the surgical instrument trajectories and fine-grained surgical states. Our model achieves accurate predictions of the end-effector path, with distance error as low as 1.64mm and MAPE of 1.72% when predicting the end-effector location 1 second in the future. The surgical state estimation accuracy achieved by daVinciNet is up to 91.02%, and compares well with human annotator accuracy of 94.2%. By accurately predicting the end-effector trajectory and surgical states in datasets with various experimental settings and complex surgical backgrounds, daVinciNet proved its robustness in realistic RAS tasks.

The necessity and advantages of including multiple data sources for the joint prediction task was illustrated by comparing the performance of the full daVinciNet

model against versions of it that received only certain types of input data. Improved performance arises, for instance, because many instrument movements have visual features and advanced cues that can be captured by daVinciNet's endoscopic vision feature extraction module. Including a full feature tensor with the historic fine-grained surgical state sequence also significantly improves the accuracy of surgical state prediction,comparing to only using one type of input. Richer information regarding surgical states can be extracted from multiple encoders, which can lead to more accurate predictions. I also showed the sizeable contribution of RoI endoscopic visual features to both the end-effector trajectory prediction and the surgical state prediction performances. daVinciNet incorporated a silhouette-based instrument tracking algorithm to identify the RoIs in the current endoscopic view (the surrounding areas of the surgical instruments' end-effectors). Additionally, the applications of existing surgical scene-understanding models (Fusion-KVE) allowed us to achieve better performances in surgical state prediction.

Chapter 6

LEARNING INVARIANT REPRESENTATION OF TASKS FOR ROBUST SURGICAL STATE ESTIMATION

Fine-grained surgical states, with their inherently frequent and spontaneous state transitions during real-world RAS procedures, are challenging to recognize, especially in real-time. In previous chapters, it has been shown that the incorporation of multiple types of input data, including the robot kinematics, endoscopic vision, and system events, can improve the fine-grained surgical state estimation accuracy. With the emergence of deep learning-based surgical state estimation models, however, additional challenges arose as they rely heavily on the datasets for model fitting/training. Limitations in the dataset can be propagated (and perhaps amplified) to the estimator, possibly resulting in a lack of robustness and cross-domain generalizability, which is crucial for the safety-critical field of medicine.

Prior to this thesis work, most currently available RAS datasets were extremely limited in dataset size and were derived from highly uniform tasks performed using the same technique in only one setting. The JIGSAWS dataset, for example, contains the task of suturing obtained only in a bench-top setting with suturing performed on marked pads. Valuable anatomical background visual information during a surgery is not present in the training data, which may lead to errors when a fine-grained surgical state estimator trained on it is applied in real-world surgeries. Another important factor for a more realistic RAS dataset is the endoscope motion, which is frequent and spontaneous during real-world RAS. Surgical state estimators that were trained on datasets devoid of any endoscope motion do not generalize well to new endoscopic views. Additionally, users in existing surgical activity datasets typically perform the task with the same technique, or were instructed to follow a predetermined workflow, which limits the technique variability among trials. During an RAS, however, different surgeons might have diverse preferred styles and techniques to perform the same surgical task. These limitations can cause surgical state estimators to overfit to the techniques presented during training. During the deployment of the trained system, inaccurate associations between surgical states and specific instrument placements and visual layout can occur, resulting in significant errors.

In real-world RAS, endoscope lighting and angles, surgical backgrounds, and patient health conditions vary considerably even for the same type of procedures, as do surgical state transition probabilities. These variations are considered as examples of potential *nuisance factors* that increase the training difficulty of a robust surgical state estimation model. Additionally, surgeons may employ diverse techniques to perform the same surgical task depending on the patient condition and their own preferences. The limited size of real-world RAS datasets, coupled with their high variability, presents significant challenges to the development of data-driven surgical state estimation methods. While the effects of nuisances and technique variations on surgical state estimation accuracy can be reduced by training the model with a large and diverse real-world dataset, such datasets are costly to acquire. The combination of limited data and high diversity calls for more robust state estimation training methods, as state-of-the-art methods are not accurate enough for adoption in the safety-critical field of RAS.

This chapter attacks this problem through Invariant Representation Learning (IRL) [123], which has been an active research topic in computer vision, where robustness is achieved through *invariance induction* [1, 47, 48, 123, 128]. Specifically, I developed a robust and accurate fine-grained surgical state estimation framework (denoted as StiseNet) that is largely invariant to nuisances and variations in surgical techniques. Through an adversarial model design that pits two composite deep learning models against each other during training, StiseNet yields an invariant latent representation of the endoscopic video, robot kinematics, and system event data during RAS for fine-grained surgical state estimation. Through the competitive training of state estimation and input data reconstruction, and the disentanglement between essential information for state estimation and nuisance factors, StiseNet learns a split representation of the input data. The influence of surgeon technique is excluded by adversarial training between state estimation and the obfuscation of a latent variable representing the technique type employed. StiseNet's training does not require any additional annotation apart from the target variable (surgical states). In the following sections, the mechanism of invariance induction through adversarial training, our model design that achieves invariance induction during RAS, and the model's performances are discussed. The work presented in this chapter was described in publications [94, 95].

6.1 Invariance induction through adversarial training

One of the most popular and commonly seen forms of supervised machine learning application is the association between data (features) and labels (target variables). This process involves decomposing the features into factors of variation [47] and learning its correlations with the target variables. The *nuisance factors* in the data, however, may be incorrectly associated with the target variables and cause the trained model to overfit and/or have poor generalizability. For example, when training a hand-written digit recognition model with pictures of hand-written numbers, the variation in lighting conditions of the training images is a nuisance factor.

Various techniques have been proposed to combat the problem of nuisance factors, as such noises are commonly present in the data. Feature selection - either manually or through model design such as pooling strategies - is an effective method to eliminate nuisance factors from data [16]. Data augmentation such as re-scaling/rotating images or adding various types of noises is another popular method widely adapted in the computer vision and deep learning research community [56]. These relatively naive methods, however, require a significant amount of manual efforts or a large and diverse dataset; therefore, they are less scalable in the current era of deep learning research. Additionally, these methods only eliminate or reduce the effect or certain and targeted types of nuisance factors.

Invariance induction was another widely adapted method to reduce the negative effect of nuisance factors [1, 4, 48]. The invariance induction to nuisance factors is achieved through the learning of a latent representation of the data that is invariant to nuisance and does not contain information about these factors. Zemel et al. proposed a supervised adversarial model to achieve the fair classification under the two competing goals of encoding the input data correctly and obfuscating the group to which the data belongs [128]. A regularized loss function using information bottleneck also induces invariance to nuisance factors [1]. Jaiswal et al. described an adversarial invariance framework in which nuisance factors are distinguished through disentanglement [48], and bias in the data that are associated with the target variable is distinguished through the competition between goal prediction and bias obfuscation [47]. Previous work on IRL via adversarial invariance in time series data have focused mostly on speech recognition [44, 89].

RAS data, arising from multiple sources, provides a new domain for IRL of high-dimensional noisy time series data. Fine-grained surgical state estimation can be made invariant to irrelevant nuisances and surgeon techniques if the latent represen-

Notation	Description
$\boldsymbol{H}$	Concatenated endoscopic video, robot kinematics, and event features $\boldsymbol{H} = \{\boldsymbol{h}^{vis}, \boldsymbol{h}^{kin}, \boldsymbol{h}^{evt}\}$
s	Surgical state
T_{obs}	Observational window size
$\boldsymbol{e}_1$	All factors pertinent to the estimation of s
$\boldsymbol{e}_2$	All other factors (nuisance factors), which are of no interest to goal variable estimation [12]
l	Latent variable (type of surgical technique)
$\overline{d}$	Mean silhouette coefficient quantifying clustering quality
E	Encoder encodes $\boldsymbol{H}$ into $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$
M	Estimator infers s from $\boldsymbol{e}_1$
ψ	Dropout
R	Reconstructer attempts to reconstruct $\boldsymbol{H}$ from $[\psi(\boldsymbol{e}_1), \boldsymbol{e}_2]$
f_1	Disentangler infers $\boldsymbol{e}_2$ from $\boldsymbol{e}_1$
f_2	Disentangler infers $\boldsymbol{e}_1$ from $\boldsymbol{e}_2$
D	Discriminator estimates l from $\boldsymbol{e}_1$

Table 6.1: Key variables, concepts, and notations used in StiseNet.

tation of the input data contain minimal information about those factors [123]. To effectively learn such invariant latent representation, I adopt an adversarial model design loosely following Jaiswal et al. [47], but with model architectures more suitable for time series data. Jaiswal et al.'s adversarial invariance framework for image classification separates useful information and nuisance factors, such as lighting conditions, before performing classification. StiseNet extended this idea by separating learned features from RAS time series data into desired information for surgical state estimation ($\boldsymbol{e}_1$) and other information ($\boldsymbol{e}_2$). Estimation was then performed using only $\boldsymbol{e}_1$ to eliminate the negative effects of nuisances and variations in surgical techniques. LSTM computational blocks were used for feature extraction and surgical state estimation. An LSTM model learns memory cell parameters that govern when to forget/read/write the cell state and memory [24]. They therefore better capture temporal correlations in time series data. In the following subsections, I describe StiseNet's model architecture, our invariance induction mechanisms, and our training strategies. Table 6.1 lists the key concepts and notations that are important to the understanding of the model's framework.

Spatial and temporal feature extractions

Fig. 6.1 depicts the extraction and encoding of features from the endoscopic video, robot kinematics, and system events data in an RAS dataset used in StiseNet. As

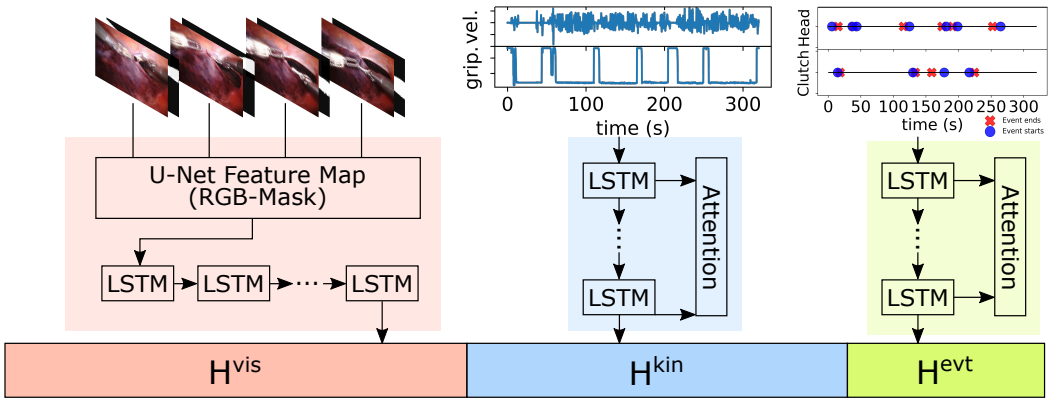

Figure 6.1: Features $\boldsymbol{h}^{vis}$, $\boldsymbol{h}^{kin}$, and $\boldsymbol{h}^{evt}$ are respectively extracted from endoscopic vision, robot kinematics, and system events. A semantic mask is appended to the endoscopic vision data to form an RGB-Mask vision input.

described in Chapter 2, at the stage of endoscopic video spatial feature extraction, various methods could be applied to reduce the negative effect of environmental distractions and complexity in the surgical background. In StiseNet, a separately trained and frozen surgical scene segmentation model based on U-Net [98] was implemented to extract a pixel-level semantic mask for each endoscopic video frame. The derivation of the U-Net model was described in Section 2.3. Three scene classes were used: tissue, surgical instruments, and others. Although a semantic mask of the current endoscopic video frame eliminates the environmental variability in the surgical background, it also omits potentially important features that are important to fine-grained surgical state estimation as each pixel of the video frame was assigned to one in only three categories. To balance the complexity of a raw endoscopic image and the simplicity of its semantic mask, I concatenated the semantic mask to the unmodified endoscope image as a fourth image channel. This RGB-Mask image $\boldsymbol{I}_t \in \mathbb{R}^{h \times w \times 4}$ was then the input to extract spatial endoscopic video features $\boldsymbol{x}_t^{vis}$, since a condensed surgical scene representation can be taken advantage of by adapting U-Net weights of the semantic segmentation model trained on a large endoscopic image dataset. An LSTM encoder was implemented to capture the temporal correlations in visual CNN features. This helps the visual processing system to extract visual features that evolve in time. At time t, a visual latent state, $\boldsymbol{h}_t^{vis} \in \mathbb{R}^{n_{vis}}$, is extracted with the LSTM model.

At time step t, the robot kinematics temporal features are extracted using an LSTM encoder with both input attention mechanism [91] and temporal additive attention mechanism [118] to identify the important kinematics data types [94]. Following

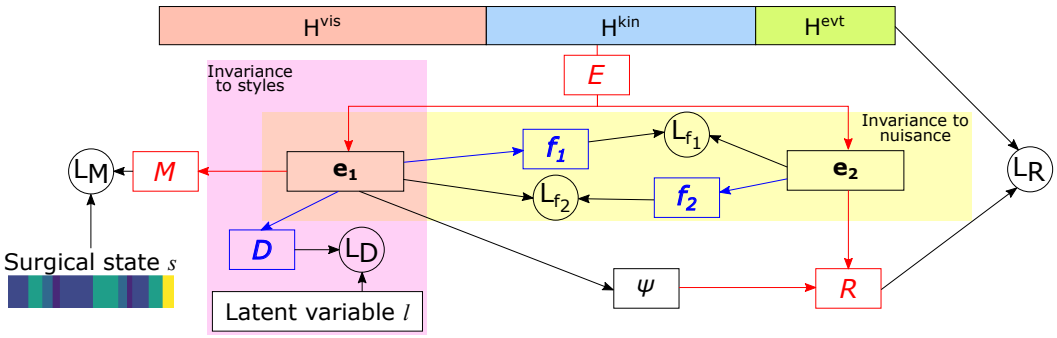

Figure 6.2: StiseNet's training architecture. Symbols for the estimator components $P1 = \{E, M, R\}$ are red, the adversarial component $P2 = \{f_1, f_2, D\}$ is blue, and training loss calculations are black. $P2$ implements invariance to nuisance (yellow shading) and surgical techniques (pink shading). RAS data features $\boldsymbol{H}$ are divided into information essential for state estimation, $\boldsymbol{e}_1$, and other information $\boldsymbol{e}_2$. $\boldsymbol{H}$ is reconstructed from $\psi(\boldsymbol{e}_1)$ and $\boldsymbol{e}_2$, where ψ is dropout.

the same method as Section 5.1, a multiplier $\boldsymbol{\beta}_t$, whose elements weight each type of kinematics data, was learned as follows:

$$\boldsymbol{\beta}_t = softmax\left\{\boldsymbol{u}^T tanh\left(\boldsymbol{W}(\boldsymbol{h}_{t-1}^{kin}, \boldsymbol{c}_{t-1}^{kin}) + \boldsymbol{V}\boldsymbol{X}_t^{kin}\right)\right\}, \tag{6.1}$$

where $\boldsymbol{h}_{t-1}^{kin}$ is the latent state from the previous frame, $\boldsymbol{c}_{t-1}^{kin}$ is the LSTM cell state, and $\boldsymbol{X}_t^{kin} = (\boldsymbol{x}_{t-T_{obs}+1}^{kin}, ..., \boldsymbol{x}_t^{kin})$ denote the kinematic data inputs. $\boldsymbol{u}$, $\boldsymbol{W}$, and $\boldsymbol{V}$ are learnable parameters. The weighted kinematics data feature vector $\boldsymbol{h}_t^{kin} \in \mathbb{R}^{n_{kin}}$ is calculated as:

$$\boldsymbol{h}_t^{kin} = LSTM(\boldsymbol{h}_{t-1}^{kin}, \boldsymbol{\beta}_t \cdot \boldsymbol{x}_t^{kin}). \tag{6.2}$$

The system event features $\boldsymbol{h}_t^{evt}$ are extracted via the same method as the robot kinematics data.

Feature encoder and surgical state estimator

As shown in Fig. 6.2, an encoder E extracts useful information for fine-grained surgical state estimation from the encoded feature data $\boldsymbol{H}$. If we assume that $\boldsymbol{H}$ is composed of a set of factors of variation, then $\boldsymbol{H}$ is composed of mutually exclusive subsets:

- $\boldsymbol{e}_1$: all the factors pertinent to the estimation of the goal variable (in this case, the current surgical state s);
- $\boldsymbol{e}_2$: all other factors (nuisance factors), which are of no interest to goal variable estimation [12].

Encoder E is a function trained to partition $\boldsymbol{H}$: $[\boldsymbol{e}_1, \boldsymbol{e}_2] = E(\boldsymbol{H})$. Specifically, an FC layer was implemented to map $\boldsymbol{H}$ to $\boldsymbol{e}_1$, and another FC layer to map $\boldsymbol{H}$ to $\boldsymbol{e}_2$. Once distinguished, the surgical state s at time t is estimated from the concatenation of time series data of the useful signal $\{\boldsymbol{e}_{1,t-T_{obs}+1}, \ldots, \boldsymbol{e}_{1,t}\}$, where T_{obs} is the size of the observation window. The fine-grained surgical state estimation was achieved with an LSTM decoder with temporal attention mechanism following the same method as described in Chapter 5. The LSTM decoder is denoted $\boldsymbol{M}$. By learning the parameters in M using $\{\boldsymbol{e}_{1,t-T_{obs}+1}, \ldots, \boldsymbol{e}_{1,t}\}$ instead of $\{\boldsymbol{H}_{1,t-T_{obs}+1}, \ldots, \boldsymbol{H}_{1,t}\}$, we avoided learning inaccurate associations between nuisance factors and the goal variable (fine-grained surgical state s).

Learning an invariant representation

The invariance induction to nuisance and technique factors is learned via *competition* and *adverseness* between model components [37] (yellow and pink shaded components in Fig. 6.2). While M encourages the pooling of factors relevant to surgical state estimation in signal $\boldsymbol{e}_1$, a reconstructor R (a function implemented as an FC layer) attempts to reconstruct $\boldsymbol{H}$ from the separated signals $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. Dropout ψ is added to $\boldsymbol{e}_1$ to make it an unreliable source to reconstruct $\boldsymbol{H}$ [48]. This configuration of signals prevents a convergence to the trivial solution where $\boldsymbol{e}_1$ monopolizes all information, while $\boldsymbol{e}_2$ contains none. The mutual exclusivity between $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ is achieved through adversarial training. Two FC layers f_1 and f_2 are implemented as *disentanglers*. f_1 attempts to infer $\boldsymbol{e}_2$ from $\boldsymbol{e}_1$, while f_2 infers $\boldsymbol{e}_1$ from $\boldsymbol{e}_2$. To achieve mutual exclusivity, the information in $\boldsymbol{e}_1$ should not be able to infer $\boldsymbol{e}_2$ or vise versa. Hence, the losses of f_1 and f_2 must be maximized. This leads to an adversarial training objective [75]. The loss function with invariance to nuisance factors is:

$$\begin{aligned} L_{nuisance} = \alpha L_M\left(s, M(\boldsymbol{e}_1)\right) + \beta L_R\left(\boldsymbol{H}, R(\boldsymbol{e}_2, \psi(\boldsymbol{e}_1))\right) \\ + \gamma\left(L_f\left(\boldsymbol{e}_1, f_1(\boldsymbol{e}_2)\right) + L_f\left(\boldsymbol{e}_2, f_2(\boldsymbol{e}_1)\right)\right) \end{aligned} \tag{6.3}$$

where α, β, and γ respectively weight the adversarial loss terms [75] associated with architectural components M, R, and disentanglers f_1 and f_2. The training objective with invariance to nuisance factors is a minimax game [37, 74]:

$$\min_{P1} \max_{P2_{nuisance}} L_{nuisance} \tag{6.4}$$

where the loss of component $P1 = \{E, M, R\}$ is minimized while the loss of $P2_{nuisance} = \{f_1, f_2\}$ maximized.

Besides the presence of nuisance factors, variability in $\boldsymbol{H}$ could also arise from variability in surgical techniques. Variations in technique may not be entirely separable by an invariance to nuisance factors, as they may be correlated to the surgical state. StiseNet therefore adopts an *adversarial debiasing* design [129] that deploys a *discriminator* $D : \boldsymbol{e}_1 \rightarrow l$ for surgical technique invariance. l represents the type of technique employed by the surgeon to perform a surgical task. l is a trial-level categorical attribute that is inferred by k-means clustering of kinematics time series training data based on a dynamic time warping distance metric[102]. The clusters represent different surgical techniques used in the training trials. The optimal number of clusters k is dataset-specific. To determine it, I implemented the *elbow method* using inertia [30] and the *silhouette method* [100].

The inertia is defined as the sum of distances between each cluster member and its cluster center[30] for all clusters. The inertia of a cluster C Q_C is:

$$Q_C = \sum_{i=1}^{N} \phi(i, c_C), \tag{6.5}$$

where N is the number of samples in cluster C, ϕ is the dynamic time warping distance function, and c_C is the center of cluster C. The inertia decreases as k increases. The elbow point (the point after which the inertia starts to decrease in a linear fashion) of the inertia-k curve is a relatively optimal k value [30].

To calculate the silhouette coefficient d_i for time series $i \in C_I$ (i in the cluster C_I), I first calculate the mean distance between i and all other time series in cluster C_I:

$$a_i = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} \phi(i, j), \tag{6.6}$$

where $|C_I|$ is the number of time series in cluster C_I. It is worth noticing that a_i is a measure of how well i is assigned to its cluster [100]. The dissimilarity of time series i to a cluster $C_J (J \neq I)$ is defined as the mean of the distance from i to all time series in C_J. We define:

$$b_i = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} \phi(i, j) \tag{6.7}$$

as the smallest mean distance of time series i to all points in the closest cluster. The operation $\min_{J \neq I}$ represents the closest cluster to i, of which i is not a member. The

silhouette coefficient d_i is then defined as:

$$d_i = \begin{cases} \frac{b_i - a_i}{max(a_i, b_i)} & \text{if } |C_I| > 1 \\ 0 & \text{if } |C_I| = 1 \end{cases}. \tag{6.8}$$

I used the mean silhouette coefficient among all time series $\overline{d}$ to select k. $\overline{d}$ is a measure of how close each data point in one cluster is to data points in the nearest neighboring cluster. The k with the highest $\overline{d}$ is the optimal number of clusters [100]. The loss function with invariance to both nuisance and surgical techniques is then:

$$L = L_{nuisance} + \delta L_D\left(l, D(\boldsymbol{e}_1)\right) \tag{6.9}$$

where δ is the weight associated with the discriminator loss. The term $P2$ contains an additional term: $\tilde{P2} = \{f_1, f_2, D\}$:

$$\min_{P1} \max_{\tilde{P2}} L. \tag{6.10}$$

6.2 Implementation and training strategies

The spatial feature extraction component of StiseNet was trained following a similar strategy as described in Section 4.2. Specifically, the first three channels of the top layer in U-Net visual feature map were initialized with the weights from the surgical scene segmentation model that was previously trained and frozen. The visual input was resized to $h = 256$ and $w = 256$. The extracted features have dimensions $n_{vis} = 40$, $n_{kin} = 40$, and $n_{evt} = 4$, which were determined after a grid search of parameters. All data sources are synchronized at 10Hz with $T_{obs} = 20\ samples = 2sec$. The optimal cluster number, k, for the JIGSAWS suturing dataset, the RIOUS+ dataset, and the HERNIA-40 dataset were 9, 7, and 4, respectively. The temporal clustering process was repeated to ensure reproducibility due to the randomness in initialization. Details of the selection of the optimal k is described in the following subsection.

StiseNet is trained end-to-end with the minimax objectives (Eq.s 6.4 or 6.10). The categorical cross-entropy loss was used for L_M and L_D. L_f and L_R are mean squared error loss. ψ is a dropout [108] with the rate of 0.4, 0.1, and 0.4 for JIGSAWS, RIOUS+, and HERNIA-40, respectively. To effectively train the adversarial model, I applied a scheduled adversarial optimizer [37], in which a training batch is passed to either $P1$ or $P2$ while the other component's weights are frozen. The alternating schedule was found by grid search to be 1:5.

6.3 Performance evaluation and discussion

StiseNet's performance was evaluated on the JIGSAWS suturing dataset, the RIOUS+ dataset, and the HERNIA-40 dataset. In addition to the comparison between StiseNet's performance against state-of-the-art fine-grained surgical state estimation algorithms, ablation studies were performed to understand the necessity and contributions of StiseNet's components. In the following subsections, the model evaluation details, performance, and the effectiveness of the adversarial model design are discussed.

Model evaluation metrics and ablation study

I used the percentage of accurately identified frames in a test set to evaluate each model's surgical state estimation accuracy. Model performance was evaluated in both non-causal and causal settings. In a non-causal setting, the model can use information from future time frames, which is suitable for post-operative analyses. In a causal setting, the model only has access to the current and preceding time frames. Surgical state estimation is harder in the causal setting; however, it is a more useful evaluation metric for real-time applications. I used the source code provided by the authors of the comparison methods when the model performance of a particular setting or dataset was not available [24, 65] and performed training and evaluation ourselves. The JIGSAWS suturing and RIOUS+ datasets were evaluated using *Leave One User Out* (LOUO) [32], while HERNIA-40 was evaluated using 5-fold cross validation, since each trial's surgeon ID is not available due to privacy protection reasons.

The quality of the learned invariant representations of surgical states $\boldsymbol{e}_1$ and other information $\boldsymbol{e}_2$ was visually examined. Arrays of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ in each state instance (a consecutive block of time frames of the same fine-grained surgical state) were embedded in 2D space using the Uniform Manifold Approximation and Projection (UMAP) algorithm [77] - a widely-adopted dimension reduction and visualization method that preserves more of the global structure of the data.

StiseNet was also compared against its ablated versions: StiseNet-Non Adversarial (StiseNet-NA), StiseNet-Nuisance Only (StiseNet-NO), and StiseNet-Technique Only (StiseNet-TO). StiseNet-NA omitted the adversarial component P2 entirely and uses $\boldsymbol{H}$ directly for surgical state estimation. StiseNet-NO separated useful information and nuisance factors, but excluded the invariance to surgical techniques (pink-shaded area in Fig. 6.2). StiseNet-TO included the invariance to techniques,

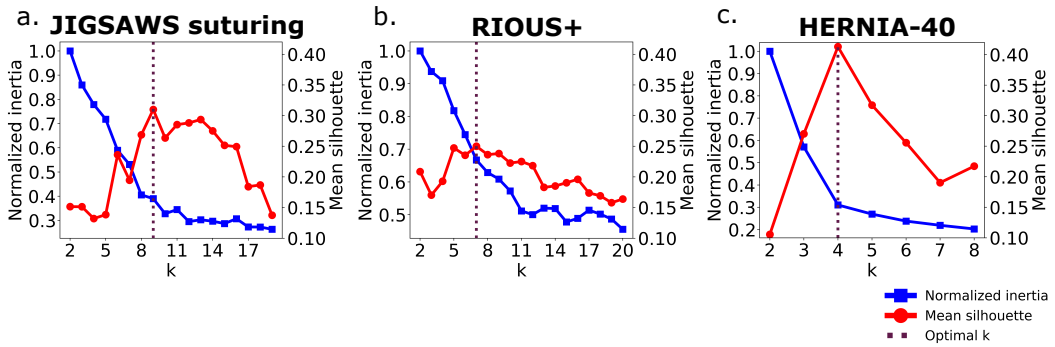

Figure 6.3: Normalized inertia (with respect to the maximum value) and mean silhouette coefficient as functions of the number of clusters k for each dataset. The vertical dotted line indicates the optimal k (the maximum mean silhouette coefficient).

State ID	Description
S1	Reach for the needle
S2	Position the tip of the needle
S3	Pushing needle through the tissue
S4	Pulling tissue with left hand
S5	Transferring needle from left to right
S6	Orienting needle
S7	Pulling suture with left hand
S8	Pulling suture with right hand
S9	Transferring needle from right to left
S10	Using right hand to tighten suture
S11	Adjusting endoscope

Table 6.2: State IDs and fine-grained surgical state descriptions for the "Close Peritoneum" superstate in HERNIA-40 dataset.

but omitted the separation between $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. The ablation study demonstrates the necessity of the adversarial model design and individual contributions of each model component towards a more accurate fine-grained surgical state estimation.

Model performance and discussion

The determination of the latent variable l used in StiseNet's training, which represents the surgical techniques employed by the user in a trial, was through k-mean clustering. As mentioned in the previous section, two parameters - the *total inertia* and the *mean silhouette coefficient* $\overline{d}$ - were used to determine the most optimal number of clusters k for each dataset. Fig. 6.3 plots for each dataset the total inertia and $\overline{d}$ as functions of the number of clusters k. The optimal number of clusters k

Non-causal

	Input data	JIGSAWS	RIOUS+	HERNIA-40
TCN[65]	kin	79.6	82.0	72.1
TCN[65]	vis	81.4	62.7	61.5
Bidir. LSTM[24]	kin	83.3	80.3	73.8
LC-SC-CRF[63]	vis+kin	83.5	-	-
3D-CNN[31]	vis	84.3	-	-
Fusion-KVE[96]	vis+kin+evt	86.3	**93.8**	78.0
StiseNet-NA	vis+kin+evt	86.5	93.1	80.0
StiseNet-TO	vis+kin+evt	88.1	88.9	81.8
StiseNet-NO	vis+kin+evt	87.9	90.3	83.2
StiseNet	vis+kin+evt	**90.2**	92.5	**84.1**

Table 6.3: Fine-grained surgical state estimation performance comparison in a non-causal experimental setting. The JIGSAWS suturing dataset results did not include system events.

Causal

	Input data	JIGSAWS	RIOUS+	HERNIA-40
TCN[65]	vis	76.8	54.8	58.3
TCN[65]	kin	72.4	78.4	68.1
Forward LSTM[24]	kin	80.5	72.2	69.8
3D-CNN[31]	vis	81.8	-	-
Fusion-KVE[96]	vis+kin+evt	82.7	89.4	75.7
StiseNet-NA	vis+kin+evt	83.4	88.9	77.3
StiseNet-TO	vis+kin+evt	84.2	87.1	81.4
StiseNet-NO	vis+kin+evt	84.1	88.9	81.0
StiseNet	vis+kin+evt	**85.6**	**89.5**	**82.7**

Table 6.4: Fine-grained surgical state estimation performance comparison in a causal setting. The JIGSAWS suturing dataset results did not include system events.

Mean Silhouette Coefficient

	e_1	e_2
JIGSAWS suturing	0.43	-0.21
RIOUS+	0.14	-0.13
HERNIA-40	0.33	-0.41

Table 6.5: The mean silhouette coefficients $\overline{d}$ of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ of each graph. A larger mean silhouette coefficient indicates better clustering quality.

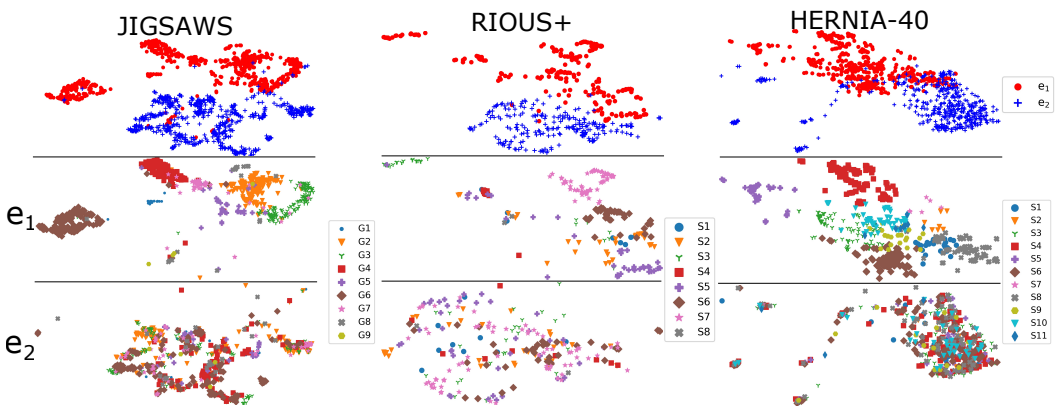

Figure 6.4: 2D UMAP plots of information enclosed in $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ at each state instance. **Top row**: $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ segregates into distinguishable clusters, which indicates little overlap in information between them. **Middle row**: Information in $\boldsymbol{e}_1$ color-coded by surgical states clusters relatively neatly. **Bottom row**: Information in $\boldsymbol{e}_2$ is more intertwined and non-distinguishable by surgical states.

can be estimated from the elbow point of the inertia-k curve, or the k associated with the maximum mean silhouette coefficient $\overline{d}$. I implemented both methods and illustrate our choices of k in Fig. 6.3. The optimal k is easily identifiable for both the JIGSAWS suturing dataset and the HERNIA-40 dataset (Fig. 6.3a and 6.3c), with the largest $\overline{d}$ occurs near the "elbow" of the inertia-k curve. A peak in the RIOUS+ dataset mean silhouette coefficient curve is less evident (Fig. 6.3b). The optimal number of clusters need not match the number of operators, as the inter-personal characteristics are not the only accountable factor for the variations among trials. Intra-personal variations can affect clustering. For instance, JIGSAWS contains metadata corresponding to expert ratings of each trial [32]: the ratings fluctuate among trials performed by the same surgeon. The optimal k determined by kinematics data is somewhat robust against patient anatomy; however, a highly unique patient anatomy can lead surgeons to modify their maneuvers significantly. Such a trial could fall into a different technique cluster.

Fig. 6.4 shows the UMAP visualizations of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ for all fine-grained surgical states in each dataset. Datapoints in Fig. 6.4 are the 2D projections of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. The first row shows that $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ separate neatly into two clusters for all datasets, validating the effectiveness of disentanglers f_1 and f_2 since $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ contain little overlapping information. Since $\boldsymbol{e}_1$ contains useful information for fine-grained surgical state estimation, while $\boldsymbol{e}_2$ does not, $\boldsymbol{e}_1$ should be better segregated into clusters associated to each state. The second and third rows of Fig. 6.4 (color-coded

by surgical state) show cleanly segregated clusters for $\boldsymbol{e}_1$, while the $\boldsymbol{e}_2$ projections are not distinguishable by state. The *mean silhouette coefficient* for each graph as shown in Table 6.5 also supports this observation. These observations strongly suggest that each surgical state has a unique representation in $\boldsymbol{e}_1$, while $\boldsymbol{e}_2$ contains little information useful for state estimation.

StiseNet's surgical state estimation performance in the percentage of accurately estimated frames was compared against various state-of-the-art fine-grained surgical state estimation methods and its ablated versions. The model performance are shown in both non-causal (Table 6.3) and causal (Table 6.4) settings, respectively. StiseNet yields an improvement in frame-wise surgical state estimation accuracy for the JIGSAWS suturing dataset (up to 3.9%) and the HERNIA-40 dataset (up to 7%) under both settings, which shows the necessity and effectiveness of the adversarial model design.

The non-causal performance of StiseNet on the RIOUS+ dataset, however, is slightly worse compared to our Fusion-KVE method as described in Chapter 4, which does not dissociate nuisance or style variables. This result can be explained by StiseNet's model design and training scheme. The added robustness of StiseNet against variations in background, surgical techniques, etc. comes at the cost of the increased training complexity that is associated with the adversarial loss functions and the minimax training. Surgeon techniques and styles vary in the JIGSAWS suturing dataset, and more significantly in the HERNIA-40 dataset. Nuisance factors (tissue deformations, endoscopic lighting conditions and viewing angles, etc.) also vary considerably among trials and users in the HERNIA-40 dataset. However, since RIOUS+ users were instructed to strictly follow a predetermined workflow, there are few nuisance and technique factors. The disentanglement between essential information $\boldsymbol{e}_1$ and other information $\boldsymbol{e}_2$ was therefore less effective.

This hypothesis is supported by the observation that the dropout rate required for StiseNet training convergence is 0.1 for the RIOUS+ dataset, whereas the training processes with JIGSAWS suturing dataset and the HERNIA-40 dataset converged with a dropout rate of 0.4, respectively. A lower dropout rate indicates that $\boldsymbol{e}_2$ contains little information despite the dropout's effort to avoid the trivial solution. Additionally, the uniformity across RIOUS+ participants results in a nearly constant mean silhouette coefficient (Fig. 6.3b). StiseNet's invariance properties therefore could not be fully harnessed, explaining its less competitive performance in RIOUS+ as compared to the real-world data of the HERNIA-40 dataset. As mentioned

before, the superiority of StiseNet originates from its robustness against nuisance and different styles, which is widely observed and significant in real-world RAS. Moreover, when a training dataset does not mimic the complicated real-world RAS scenarios with evident behavioral variations, it cannot take advantage of the full potential of StiseNet. Rather, the state estimation performance suffers due to a more complicated training scheme with a limited dataset. These notions explains the deterioration of performance in the RIOUS+ dataset comparing to Fusion-KVE - a simpler model.

Fig. 6.5 shows the high variability in HERNIA-40 data through sample sequences from three technique clusters, each performed in a distinctively different style with environmental variances. Invariance of StiseNet to nuisances and surgical techniques is shown by its accurate surgical state estimations in the presence of visibly diverse input data. In real-world RAS, surgeons may use different techniques to accomplish the same task. Fig. 6.5 shows three sample HERNIA-40 trials with distinctive suturing geometries: suturing from left to right, from right to left, and back and forth along a vertical seam. These trials fall into three clusters during the k-mean clustering process. Images from instances of states S3, S4, S5, S7, and S8 in each trial (the state IDs and descriptions are shown in Table 6.2) are shown. These images of different instances of the same state vary greatly not just in technique and instrument layout, but also in nuisance factors such as brightness and endoscope angles. Yet, StiseNet accurately estimates the surgical states due to its invariant latent representation of the input data.

Fig. 6.6 shows a sample state sequence from the HERNIA-40 dataset and the causal state estimation results using multiple methods, including a forward LSTM model [24], Fusion-KVE, and the ablated and full versions of StiseNet. Fig. 6.6 demonstrates StiseNet's robustness during rapid and unpredictable state transitions in a real-world RAS suturing task. I compare the causal estimation performance of Forward-LSTM, Fusion-KVE, the ablated, and full versions of StiseNet against ground truth. Forward-LSTM, which only uses the robot kinematics data, has a block of errors from 20s to 30s since it cannot recognize the "adjusting endoscope" state due to a lack of visual and event inputs. When those inputs are added, Fusion-KVE and StiseNet recognize this state. Fusion-KVE still shows a greater error rate due to limited training data with high environmental diversity, which reflects Fusion-KVE's vulnerability to nuisance and various surgical techniques. StiseNet-NO shows fewer error blocks, yet it is still affected by different technique types.

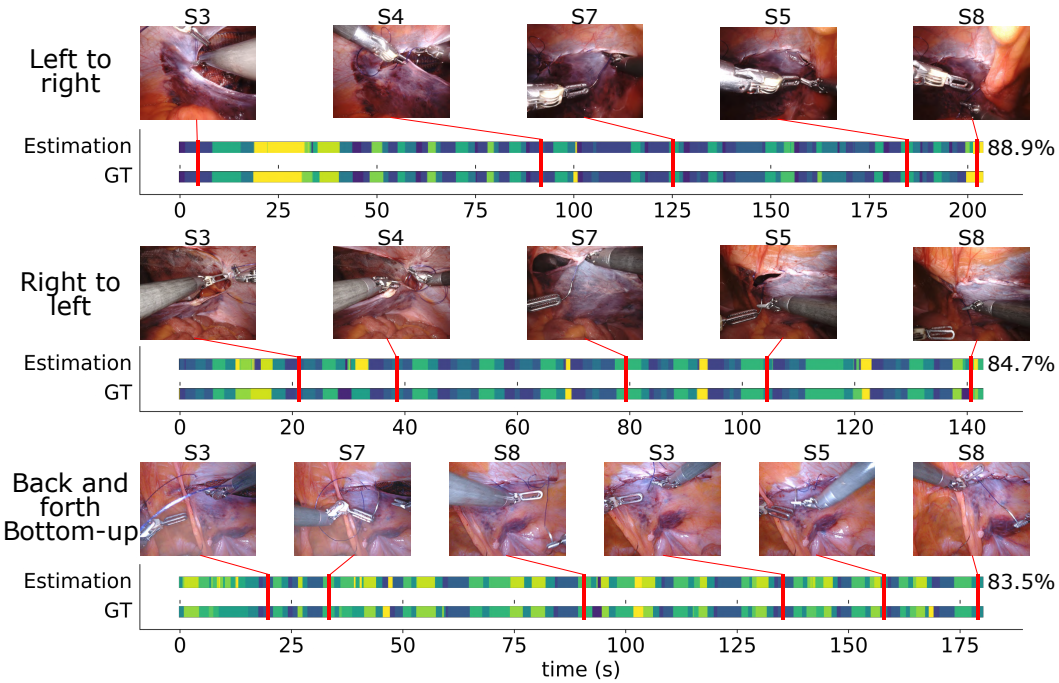

Figure 6.5: Three HERNIA-40 trials from three technique clusters, and StiseNet's performances compared to ground truth (GT). Instances of the same state in different trials are substantially and visibly different; however, StiseNet correctly estimates them. Variations across trials arise from both nuisances and surgical techniques. Potential sources of nuisances include but are not limited to lighting conditions, presence of fat or blood, peritoneum color, endoscope movements, etc.

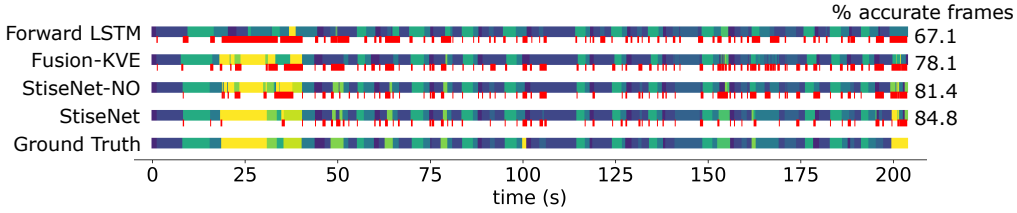

Figure 6.6: Example HERNIA-40 surgical state estimation results by forward LSTM [24], Fusion-KVE [96], StiseNet-NO, and StiseNet, compared to ground truth. State estimation results (top) and discrepancies with ground truth in red (bottom) are shown in each block bar.

The higher estimation accuracy of StiseNet shows its technique-agnostic robustness in real-world RAS, even with a small training dataset that contains behavioral and environmental diversity.

6.4 Conclusions

This chapter focused on improving the accuracy of fine-grained surgical state estimation in complex real-world RAS procedures learned from limited amounts of data with high behavioral and environmental diversity. This is especially crucial in the safety-critical field of RAS research. To do so, I employed IRL and adversarial training strategies to learn a latent representation of fine-grained surgical states that are largely invariant to nuisance factors and environmental noises during RAS as well as various surgical techniques employed by surgeons.

I designed StiseNet: an adversarial learning model with an invariant latent representation of RAS data. Through the evaluation with real-world RAS dataset that includes different surgical techniques carried out in highly diverse environments, StiseNet showed its robustness and improved the state-of-the-art performance by up to 7%. The improvement is especially significant for the real-world data, which benefit greatly from invariance to surgical techniques, environments, and patient anatomy. Ablation studies showed the effectiveness of the adversarial model design and the necessity of invariance inductions to both nuisance and technique factors. StiseNet training does not require additional annotations apart from the surgical states. StiseNet outperforms state-of-the-art surgical state estimation methods and improves frame-wise state estimation accuracy to 84%. This level of error reduction is crucial for surgical state estimation to gain adoption in AI applications during RAS. StiseNet also accurately recognizes actions in a real-world RAS task even when a specific technique was not present in the training data.

Chapter 7

CONCURRENT HIERARCHICAL SURGICAL STATE ESTIMATION THROUGH DEEP NEURAL NETWORKS

While the estimation of the current fine-grained surgical state has various AI applications in RAS, the recognition of surgical phases and tasks has found diverse applications ranging from operating room workflow coordination, surgeon skill evaluation, to workflow analysis [86, 134] as well. As surgical robots can provide synchronized endoscopic vision, robot kinematics, and surgical system events data, these data sources provide a rich representation of a surgery [94–96], which can be taken advantage of for an accurate and comprehensive awareness of the current stage of the surgery from multiple levels of temporal granularity. In addition to the fine-grained surgical actions and environmental observations as described in previous chapters, the recognition of coarser surgical states such as the current surgical task/phase that consists of fine-grained surgical states is also crucial for a more comprehensive understanding of the RAS procedure.

As described in Chapter 3, I modeled an RAS procedure as a surgical Hierarchical Finite State Model (HFSM) (Fig. 7.1) consisting of discrete superstates, where a superstate is a surgical task or an operative step. Each superstate in turn may be an FSM consisting of finer-grained states (surgeon actions and observations). An HFSM model represents at each time step the current surgery state at multi-

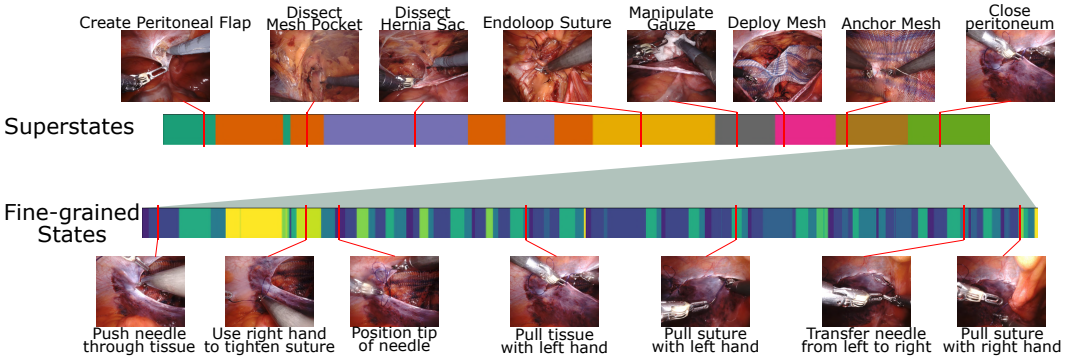

Figure 7.1: An example robotic inguinal hernia repair surgery consists of multiple surgical tasks, which are superstates in an HFSM (top row). A superstate is an FSM consisting of fine-grained surgical states. An example FSM for the superstate *close peritoneum* is shown in the bottom row.

ple levels of temporal granularity, which better captures the temporal progressions of surgeries. The simultaneous surgical state estimation at multiple level of temporal granularity therefore benefits from our modeling strategies and provides a comprehensive understanding of the surgical progress.

As an example, this chapter develops a two-level surgical HFSM and concurrent surgical state estimation methods that achieve accurate surgical (super)state estimations at both levels of temporal granularity. Specifically, two hierarchical surgical state estimation models are examined: Hierarchical Estimation of Surgical States through Deep Neural Networks (HESS-DNN) and Concurrent Hierarchical Autonomous Surgical State Estimation Network (CHASSEN). The following sections describe the details of both models' architectures and training strategies that allowed them to achieve accurate hierarchical surgical state estimation, model performances, and the effects of surgical state estimation at one level on the temporal granularity of another level. The work presented in this chapter was described in publications [92, 93, 95].

7.1 Hierarchical state estimation frameworks and training

HESS-DNN and CHASSEN were implemented to perform surgical (super)state estimation of a surgical HFSM with two levels of temporal granularity: superstates that last for minutes and fine-grained states that last for seconds. While many surgeries will have a deeper hierarchical structure, I focused on a simple hierarchy as a first step. Two hierarchical surgical state estimation models - HESS-DNN and CHASSEN - were proposed. HESS-DNN performed the surgical superstate estimation and the fine-grained state estimation through two separate pathways with little shared knowledge of each other's results. CHASSEN's network architecture, on the other hand, enabled the communication of the current surgical (super)state information between two estimators. In the following subsections, both HESS-DNN's and CHASSEN's components and model architectures are described and compared. The training strategies and implementation details of both models are also discussed.

Feature extraction and preparation for concurrent surgical state estimation

For a more direct comparison of model performances, HESS-DNN and CHASSEN shared the same feature extraction method. The RAS input data (endoscopic vision, robot kinematics, and system events) is processed following Fig. 7.2. Following the method described in Section 2.3, a semantic mask is extracted from the endoscopic

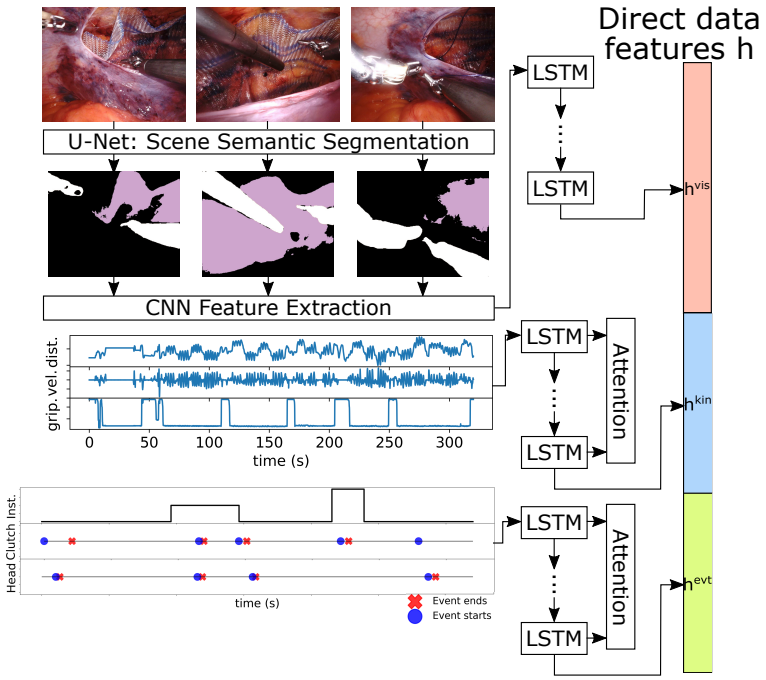

Figure 7.2: Schematic of both HESS-DNN's and CHASSEN's feature extraction components. $\boldsymbol{h}^{\text{vis}}$, $\boldsymbol{h}^{\text{kin}}$, and $\boldsymbol{h}^{\text{evt}}$ are extracted from endoscopic vision, robot kinematics, and system events, respectively.

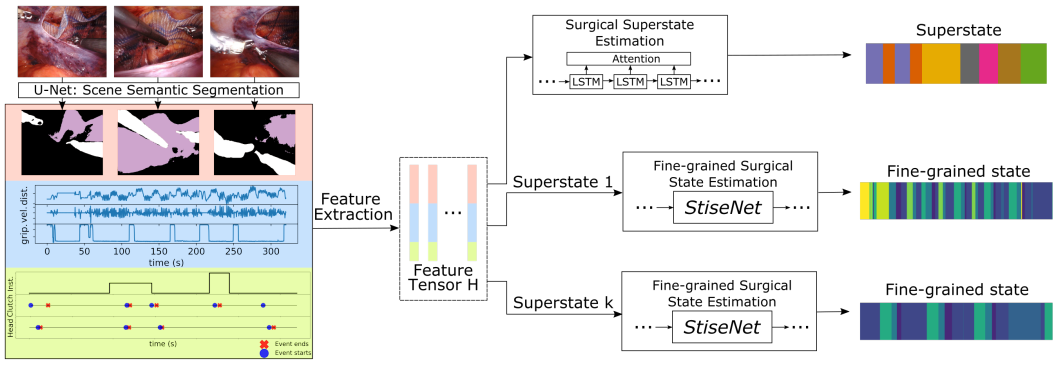

Figure 7.3: HESS-DNN's model architecture. The inputs to HESS-DNN include the endoscopic vision, robot kinematics, and system events. A feature extraction component embeds information in input data for hierarchical surgical (super)state estimation. Our previous work *StiseNet* described in Chapter 6 is implemented for the fine-grained surgical state estimation.

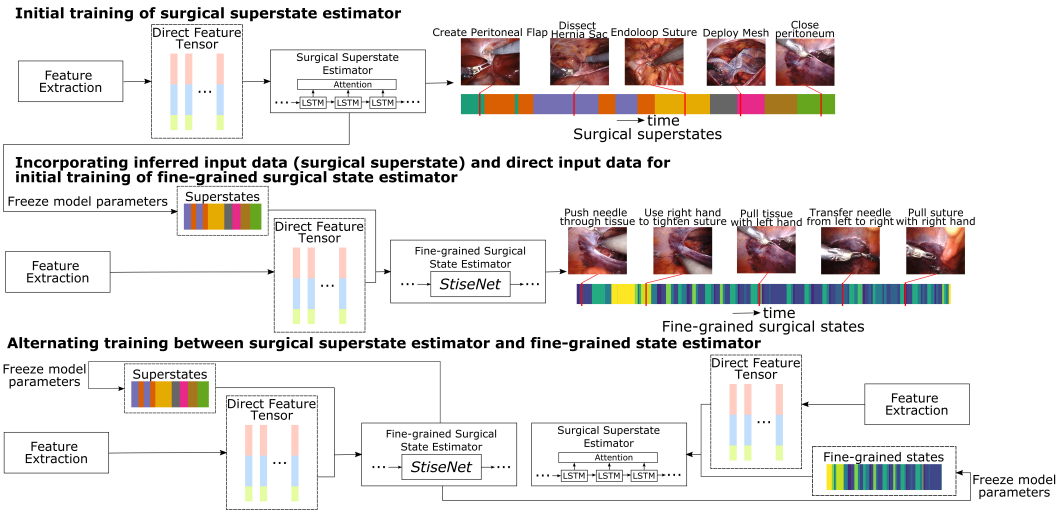

Figure 7.4: CHASSEN's model architecture and its alternating training schematics. Our previous work *StiseNet* described in Chapter 6 is implemented for the fine-grained surgical state estimation.

view to leverage a more efficient visual feature representation. The semantic mask is generated using a trained and frozen surgical scene segmentation model based on U-Net[98] and eliminates the background variability in the endoscopic view. The semantic segmentation model uses three scene classes: surgical instruments, tissue, and others. At time t, a CNN extracts information $\boldsymbol{x}_t^{\text{vis}}$ from the semantic segmentation map of the endoscopic vision. An LSTM encoder was implemented to embed the temporal correlations among $\boldsymbol{x}^{\text{vis}}$ from adjacent frames into a latent representation $\boldsymbol{h}_t^{\text{vis}} = \text{LSTM}(\boldsymbol{h}_{t-1}^{\text{vis}}, \boldsymbol{x}_t^{\text{vis}})$ [33]. The latent representation at time t therefore includes information from prior time steps. The endoscopic video features are denoted as $\boldsymbol{h}_t^{\text{vis}} \in \mathbb{R}^{n_{\text{vis}}}$.

The robot kinematic features are extracted following the same method as described in the previous chapters. Since the input kinematics data include multiple robotic arms' translations, rotations, and joint angles, I implemented an LSTM encoder with an input attention mechanism [94] to identify the driving data types. The embedded latent representation of robot kinematics takes the form $\boldsymbol{h}_t^{\text{kin}} = \text{LSTM}(\boldsymbol{h}_{t-1}^{\text{kin}}, \boldsymbol{\beta}_t \cdot \boldsymbol{x}_t^{\text{kin}})$, where multiplier $\boldsymbol{\beta}_t$ is a vector whose elements represent the weights of all input kinematics data types. An additive temporal attention mechanism was implemented to capture the temporal correlations among features at adjacent time steps more effectively as described in Section 2.3. System events data logged from dVXi was processed and the embedded latent representation of events $\boldsymbol{h}_t^{\text{evt}}$ was extracted following the same method as the robot kinematics data.

HESS-DNN model architecture

The latent representations, or "features", of the endoscopic vision, robot kinematics, and system events data were concatenated to form a feature vector $\boldsymbol{H}_t \in \mathbb{R}^{n_{\text{vis}}+n_{\text{kin}}+n_{\text{evt}}}$. Data are processed over an observation window, $[\boldsymbol{H}_{t-T_{obs}}, \boldsymbol{H}_{t-T_{obs}+\delta}, ..., \boldsymbol{H}_t]$, where T_{obs} is the window duration and δ denotes the sampling interval. Since the durations of fine-grained states and superstates are significantly different, the values of T_{obs} and δ are different for surgical superstate estimation and fine-grained state estimation. For simplicity, T_{obs} and δ are chosen such that the length of the input tensor $l = T_{obs}/\delta + 1$ is an integer. The parameter selection will be discussed in further details in the next subsection. HESS-DNN's architecture is shown in Fig. 7.3.

HESS-DNN estimates the surgical superstates via an LSTM-based decoder network that operates on the input data tensor [94]. Cho et al. showed that the performance of an encoder-decoder network could rapidly deteriorate as the length of the input data increases [18]. I therefore implemented an additional attention mechanism [91] to select important latent representations across all time steps in an adaptive manner. At time t, the attention weights $\boldsymbol{\beta}_t$ are determined from the previous decoder hidden state $\boldsymbol{d}_{t-1}$ and cell state as described in Section 2.3. The LSTM decoder is updated using the weighted feature $\tilde{\boldsymbol{H}}_t = \sum_{j=1}^{l} \beta_t^j \boldsymbol{H}_t^j$.

The estimation of fine-grained surgical states followed the determination of the current surgical superstate. HESS-DNN deploys StiseNet as described in Chapter 6 to estimate the fine-grained surgical states within a superstate. *StiseNet* employs an adversarial model design, which pits two model components against each other during training to produce a latent data representation from $\boldsymbol{H}_t$ that is invariant to nuisances (e.g., anatomical background, brightness, etc.) and variations in surgical style. StiseNet's adversarial model architecture was not applied to superstate estimation: the computational complexity of learning temporal correlations across the long superstate durations (up to 30 minutes) is prohibitive.

Correlations between surgical superstate and fine-grained state estimations

At each time step t, the current surgical superstate and fine-grained state are often highly correlated. E.g., during tissue dissection, tissue cutting is a commonly observed fine-grained state; however, it is not observed during suturing. The fine-grained state of needle manipulation, on the other hand, is frequently observed during suturing, but not during tissue dissection. Such hierarchical correlations are commonly present in a surgical HFSM, and can improve the accuracy of surgical

(super)state estimation. So far, only *direct data sources* such as the endoscopic video, robot kinematics, and system events time series data have been used for surgical (super)state estimations. The surgical (super)state time series are *inferred data sources* that have yet to be incorporated into the estimation process at other levels of temporal granularity to improve estimation accuracy. Since hierarchical correlations between surgical states are common and informative, the knowledge of the current surgical superstate can aid the fine-grained surgical state estimation, and vise versa. In Chapter 5, the historic fine-grained surgical state sequence was used for state prediction and was shown to improve the surgical state prediction accuracy. This result further supports the usage of hierarchical correlations among states for the hierarchical surgical (super)state estimations.

As seen in Fig. 7.3, HESS-DNN conducts the surgical superstate estimation and fine-grained state estimation through two uncoupled network architectures with little shared knowledge, CHASSEN serves as an initial attempt to utilize the surgical state estimation results at one level of temporal granularity to assist the state estimation of another level of temporal granularity. CHASSEN uses both direct (robot kinematics, endoscopic vision, system events) and inferred data (current surgical (super)state) as input sources for concurrent hierarchical surgical (super)state estimations. It learns the hierarchical correlations between surgical states at two levels of temporal granularity through an alternating training schematics. HESS-DNN's architecture can be further optimized through the incorporation of correlation across states in the hierarchy. The two different approaches serve to provide a comparison about how considering the correlation across states can affect hierarchical surgical state estimation.

CHASSEN includes feature extraction and surgical (super)state estimation modules and accepts both direct RAS data and inferred data sources as inputs. The surgical state-related features are extracted from direct RAS data sources via the same method as HESS-DNN as described in Section 7.1. Hierarchical correlations between surgical superstates and fine-grained states are captured through CHASSEN's training procedure (Fig. 7.4).

Knowledge of the current surgical superstate substantially improves the estimation of the current fine-grained surgical state, as the likelihood of occurrence and probability distributions of the fine-grained surgical states are correlated with the current surgical superstate. Similarly, knowledge of the current fine-grained state improves the superstate estimation process. I therefore adapted an alternating training schedule

that learns such hierarchical correlations (Fig. 7.4). CHASSEN initially trains the superstate estimator only with direct input data. The embedded latent representation of the direct input data forms a feature tensor over an observation window of the size T_{obs}, and an LSTM decoder with an addition attention mechanism [91] was used for surgical superstate estimation. The trained superstate estimator then generates the inferred data source - the surgical superstate time series. An LSTM encoder extracts the temporal correlations between superstates and their corresponding fine-grained surgical states, which is incorporated for the initial training of the fine-grained state estimator. This inferred feature tensor was concatenated to the direct feature tensor, and an LSTM decoder with an addition attention mechanism [91] was used for surgical superstate estimation [93]. Like HESS-DNN, CHASSEN implements StiseNet as described in Chapter 6 for fine-grained surgical state estimation. I then adopt an alternating training schedule between the fine-grained surgical state estimator and the surgical superstate estimator: in each training iteration, a previously trained and frozen estimator is used to generate either the surgical superstate or fine-grained state estimation result, which is concatenated with direct features to fine-tune the parameters of the other state estimator. The iteration repeats until convergence of the surgical (super)state estimation performance.

Implementation details and training strategies

During the feature extraction of direct data sources, the U-Net model for endoscopic video data semantic segmentation was separately trained on a large surgical image dataset, following [5]. The trained and frozen model was received from Intuitive Surgical Inc. The endoscopic vision input was resized to a 640×512 RGB image. I determined the dimensions of extracted features using grid search: $n_{\text{vis}} = 40$, $n_{\text{kin}} = 40$, and $n_{\text{evt}} = 4$. All direct data inputs were synchronized at 10Hz. For the surgical superstate estimation, $T_{obs} = 60$sec and $\delta = 5$. The fine-grained state estimation parameters, $T_{obs} = 2$sec and $\delta = 1$, were also determined via a grid search of parameters. HESS-DNN's training is guided by the sum of a superstate estimation loss and StiseNet's fine-grained state estimation loss:

$$L = L_{\text{super}} + L_{\text{StiseNet}} \tag{7.1}$$

where L_{super} is the categorical cross-entropy loss and L_{StiseNet} is Eq. 6.1. As discussed in Chapter 6, StiseNet's training is a minimax game [37]. HESS-DNN therefore inherits a scheduled adversarial optimizer [37], in which the generative or the discriminative component trains on a data batch while the other component's weights are frozen.

For the training process of CHASSEN, an alternating training schedule between the surgical superstate estimator and the fine-grained surgical state estimator was implemented as shown in Fig. 7.4. During the initial training of the surgical superstate estimator, the same training strategy as HESS-DNN was used with the same hyperparameters and observation window sizes with the loss function being the categorical cross-entropy loss. The inferred input data (surgical superstate estimation results) was then concatenated to the direct feature tensor for the training of StiseNet as described in Section 6.2. After the initial training of both the surgical superstate estimator and the fine-grained surgical state estimator, the alternating training schedule begins. In each training iteration, a previously trained and frozen estimator is used to generate either the surgical superstate or fine-grained state estimation result, which is concatenated with direct features to fine-tune the parameters of the other state estimator. The iteration repeats until the convergence of the surgical (super)state estimation performance, which was determined by an estimation performance improvement of less than 0.5%.

7.2 Performance evaluation and discussion

The performances of both HESS-DNN and CHASSEN were demonstrated at two levels of temporal granularity: surgical superstates and fine-grained states. The HERNIA-40 dataset was used to evaluate both hierarchical surgical state estimators. The list of surgical superstates and each superstate's fine-grained surgical states were listed in table 2.4. The quality of our hierarchical surgical state estimation models are quantified by the percentage of time steps with accurate state estimates in the test set, as judged by comparison with the ground truth annotation reviewed by experts. An ablation study investigated how the endoscopic vision, robot kinematics, and system events inputs contributes to HESS-DNN's estimation performance, respectively. Since some autonomy applications require the real-time knowledge of the surgical (super)states, I evaluated HESS-DNN and CHASSEN performances in both non-causal and causal settings. In a non-causal setting, the estimator has access to information from both preceding and future time steps. Accordingly, I implemented bi-directional LSTM units [33] in the non-causal setting. In the causal setting, the estimator only has access to data from the current and preceding time steps; therefore, forward LSTM units were implemented. The dataset was divided into training and test sets following an 80:20 split (as the surgeon identity is not available) and a five-fold cross validation was performed.

I also evaluated the contributions of various input data to surgical (super)state es-

Surgical Superstate Estimation Accuracy (%)

Model/Input data	Non-causal	Causal
Trivial	12.5	12.5
System events	40.6 ± 17.87	39.4 ± 19.04
Raw endoscopic vision	47.1 ± 10.10	41.4 ± 8.91
Robot kinematics	65.9 ± 8.41	52.3 ± 8.75
Endoscopic vision semantic mask	70.1 ± 5.17	64.5 ± 9.67
HESS-DNN-NU	76.6 ± 4.04	70.6 ± 4.77
HESS-DNN	87.6 ± 3.91	84.1 ± 4.11
CHASSEN	**89.8± 5.09**	**88.4 ± 4.89**

Table 7.1: Surgical superstate estimation performance of HESS-DNN, its ablated versions, and CHASSEN when evaluated on the HERNIA-40 dataset.

timation through an ablation study by removing subsets of input data types and comparing the ablated models' performances to the complete estimator. Specifically, HESS-DNN's performance with only one type of input data was compared against its performance with the full input data. Additionally, I examined the effectiveness of semantic segmentation on visual feature extraction by comparing HESS-DNN against an ablated version, HESS-DNN-NU, which omits the semantic segmentation of endoscopic vision and instead extractd features directly from raw endoscopic videos. The ablation study demonstrated the necessity of each input data type and validated HESS-DNN's design. As CHASSEN aims to improve hierarchical surgical state estimation accuracy through an alternating training schedule and the correlations between surgical state estimations at different levels of temporal granularity, no ablation study was applied to it.

Tables 7.1 and 7.2 quantified HESS-DNN's and CHASSEN's estimation performances in both non-causal and causal settings. State estimation accuracy is shown for both surgical superstates (tasks) and all fine-grained states in Table 2.4. Table 7.2 also compares the models' overall fine-grained state estimation performances against state-of-the-art fine-grained state estimation methods. Fig. 7.5 presents a state sequence from one HERNIA-40 surgery in order to visualize the surgical (super)state estimation quality of HESS-DNN and its ablated versions.

Table 7.1, which compares the surgical superstate estimation performance of HESS-DNN, its ablated versions, and CHASSEN, indicates that each type of input data contributes to hierarchical surgical state estimations to different degrees. When the endoscopic video, robot kinematics, and system events input data are all included, the

Overall Fine-grained State Estimation Accuracy (%)

Model/Input data	Non-causal	Causal
Trivial	4.3	4.3
TCN (vision) [65]	45.7 ± 10.04	41.9 ± 12.66
TCN (kinematics) [65]	48.9 ± 16.40	43.4 ± 17.40
Forward LSTM [24]	50.1 ± 8.11	49.7 ± 11.54
Bidir. LSTM [24]	54.7 ± 7.97	-
Fusion-KVE[96]	62.0 ± 6.05	59.9 ± 7.17
StiseNet [95]	64.1 ± 8.66	61.4 ± 8.08
HESS-DNN-NU	70.1 ± 6.71	66.9 ± 6.94
HESS-DNN	80.4 ± 5.60	75.7 ± 5.31
CHASSEN	**82.4.3 ± 4.37**	**77.3 ± 5.90**

Table 7.2: Overall fine-grained surgical state estimation performance comparison across all superstates in the HERNIA-40 dataset.

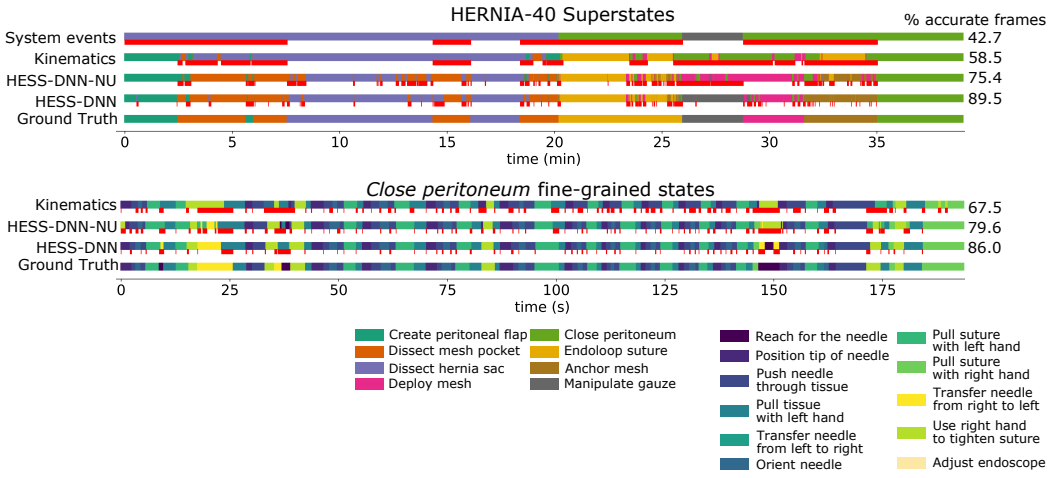

Figure 7.5: An example HERNIA-40 superstate sequence (top) and the fine-grained state sequence of the *close peritoneum* superstate (bottom). The causal estimation results are compared against the manually annotated ground truth. The discrepancies between (super)state estimation results and the ground truth are marked in red.

surgical superstate estimator achieves a superior frame-wise superstate estimation accuracy in both non-causal and causal settings comparing to single-source models. The significant improvement in estimation accuracy comparing to HESS-DNN with a single type of input (first four rows) further confirms the advantage of including multiple sources of input data. As suggested in Chapter 4, different fine-grained surgical states contain different representative features, which may be recognizable through certain types of input data but not others.

The performance improvement shown in Table 7.1 indicatea the same for surgical superstate estimation. For instance, the SS4 (*Close peritoneum*) and SS5 (*Endoloop suture*) superstates both involve suturing; therefore, the installed surgical instrument for both superstates is a large needle driver. The system events data therefore cannot distinguish between these two superstates; however, the visual and kinematics features of SS4 and SS5 are significantly different and distinguishable. On the contrary, the dissection surgical superstates (SS0-SS2) require an energy instrument for cautery. The system event associated with pressing the energy pedals is therefore useful in distinguishing SS0-SS2 from other superstates. In comparison to fine-grained surgical state estimation, there has been less prior work on surgical superstate estimation, with little open-source code to facilitate comparisons; however, both of our models' accurately performed surgical superstate estimation in a complex real-world RAS environment with limited data.

The semantic segmentation of endoscopic video data also contributed to better surgical (super)state estimations, yielding a more effective visual feature extraction. I analyzed how the extraction of semantic segmentation maps from endoscopic vision data affects surgical (super)state estimation. HESS-DNN yields up to a 13.5% improvement in superstate estimation accuracy as compared to HESS-DNN-NU, which omits the U-Net model and extracts visual features directly from raw endoscopic videos. Two factors contributed to this improvement. The extraction of a surgical scene semantic segmentation map significantly reduces environmental distractions and variability. For instance, the surgical mesh implants and gauze deployed during inguinal hernia repair have various colors and forms, depending upon their manufacturers. The variation in mesh appearances is irrelevant to surgical superstate estimation. Through U-Net semantic segmentation, such distractions are eliminated. Additionally, the U-Net model was separately trained and frozen on an extensive surgical scene dataset containing 25,000 semantically annotated images from real-world RAS. The training of a surgical (super)state estimator can therefore

take advantage of a condensed representation of the endoscopic view provided by extensive training data for semantic segmentation purposes. This is especially valuable in real-world RAS datasets, like HERNIA-40, where data is limited and costly to obtain.

From Tables 7.1 and 7.2, it is clear that the determination of surgical state at one level of temporal granularity improves the state estimation performance at other levels of temporal granularity. In Table 7.2, I compared the overall fine-grained surgical state estimation of our models against state-of-the-art fine-grained state estimation methods. The fine-grained state estimation methods were trained to estimate all fine-grained states listed in Table 2.4, with the repeated states treated as the same class. The high complexity of RAS and limited training data hindered flat (non-hierarchical) fine-grained state estimation performances, as there were 23 states in total in a highly dynamic and complex RAS setting. Both HESS-DNN and CHASSEN improved the fine-grained state estimation accuracy significantly comparing to flat estimation techniques, as the superstate estimation is carried out prior to fine-grained state estimation and therefore narrowed down the pool of candidate fine-grained surgical states.

A closer inspection of the surgical (super)state estimation results in Fig. 7.5 further supports the observations discussed above. In an example HERNIA-40 superstate sequence, our model can recognize SS0-SS2 from other superstates when it only has access to system events; however, it is unable to further distinguish among superstates, resulting in poor estimation results. When the only data source is the robot kinematics data, HESS-DNN's ablated version is unable to distinguish among superstates *close peritoneum*, *endoloop suture*, and *anchor mesh* (25min - 34min). Since these three superstates all involve suturing maneuvers, distinguishable kinematics features were difficult to extract. The incorporation of endoscopic vision data overcame this challenge. Many intermittent errors are observed in HESS-DNN-NU for both surgical superstate and fine-grained state estimations, which suggests ineffective visual features are extracted from limited amounts of data. The frequent fluctuations in state identification, which causes the intermittent pattern of errors, further indicates the infirmity of HESS-DNN-NU. Clearly, a full hierarchical surgical state estimator with multiple types of input data achieves the most accurate estimation given a limited dataset in a complex real-world RAS setting.

Comparing to HESS-DNN's de-coupled surgical superstate estimation and fine-grained state estimation process, CHASSEN achieved a surgical superstate estima-

tion accuracy of 88.4% in the causal setting, a 4.3% improvement over HESS-DNN. CHASSEN also improved the fine-grained state estimation performance to 77.3% comparing to HESS-DNN's 75.7%. CHASSEN's improvements in both surgical superstate and fine-grained state estimation accuracy highlighted the importance of considering correlations between hierarchical surgical states. Additionally, CHASSEN's lighter network architecture (as compared to HESS-DNN) significantly improved its state estimation processing time. Since HESS-DNN performs hierarchical (super)state estimations in a decoupled manner, it uses multiple deep neural network architectures trained independently. The processing time of HESS-DNN is 7.1 frames per second on a workstation equipped with an NVIDIA GTX 1080 Ti graphics card, an Intel Core i7-6700 CPU, and 16GB of RAM. CHASSEN estimates both the current surgical superstate and fine-grained states at 9.3fps - a 31% gain. CHASSEN performs more accurate hierarchical surgical state estimation with greater efficiency due to its incorporation of hierarchical correlations between surgical (super)states.

7.3 Conclusions

This chapter presented a first attempt at the hierarchical surgical (super)state estimation process at multiple levels of temporal granularity through a two-level surgical HFSM constructed from the HERNIA-40 dataset. Each surgical superstate in a surgical HFSM represents a surgical task, and consists of fine-grained surgical actions and observations. Two hierarchical surgical state estimation models - HESS-DNN and CHASSEN - were implemented to simultaneously estimate the current surgical superstate and fine grained state of the inguinal hernia repair robotic surgery using multiple types of input data. The performances of HESS-DNN and CHASSEN were illustrated through their application to the HERNIA-40 dataset.

HESS-DNN's surgical superstate estimation result narrows down the pool of candidate fine-grained states and improves the fine-grained state estimation performance of state-of-the-art methods by up to 16.3% when evaluated on a large set of diverse surgical states in a real-world RAS setting. Additionally, I showcased the necessity and contributions of each type of input data to surgical super(state) estimations through an ablation study. However, HESS-DNN's architecture and performance have limitations, as the surgical superstate estimation and fine-grained state estimation were conducted through two uncoupled network architectures with little shared knowledge. This model structure limited the hierarchical state estimation performance in both accuracy and efficiency aspects.

CHASSEN attempts to incorporate the correlations across states in the hierarchy through an alternating training schematic. Comparing to HESS-DNN, CHASSEN's learning and utilization of hierarchical correlations between surgical states at multiple levels of temporal granularity allowed it to achieve a higher state estimation accuracy with a more lightweight network architecture and higher estimation speed, which shows a strong promise for more effective and efficient hierarchical surgical state estimation ability.

Chapter 8

CONCLUSIONS

8.1 Summary of thesis contributions

This thesis developed novel methods for the autonomous perception, understanding, and awareness of robot-assisted surgeries from a temporal perspective. The primary contributions of this work include the proposal of a new modeling methodology that describes an RAS, the development of two new datasets containing real-world RAS procedures, and the development of multiple novel deep learning-based models that achieves an accurate and robust temporal understanding of an RAS procedure. Specifically, I modeled an RAS procedure as a hierarchical system of discrete surgical states that I referred to as the surgical Hierarchical Finite State Model (HFSM). An RAS procedure consists of standardized operational steps and surgical tasks. These tasks could take the form of suturing, tissue dissections, etc. Each of these task/step is, in turn, comprised of finer-grained surgical actions, gestures, and environmental observations that is instantaneous or lasts for seconds such as picking up a needle, which are referred to as fine-grained surgical states. The recognition and awareness of the current surgical task being performed as well as the current fine-grained surgical states is one of the crucial prerequisites for many AI applications in RAS.

Additionally, two new RAS datasets - RIOUS+ and HERNIA-40 - were collected and developed as part of this thesis work (Chapter 2). Deep learning research in the field of RAS has suffered from a lack of real-world RAS datasets, which is necessary for the training and evaluation of an accurate and robust hierarchical surgical state estimation model. The two new datasets, unlike previously available RAS activity datasets such as JIGSAWS, contains endoscope movements and diverse experimental settings including dry-lab, cadaveric, and in-vivo trials. Moreover, the HERNIA-40 dataset contains real-world hernia repair RAS procedures performed by surgeons, which is extremely valuable for learning-based models. The development of these RAS datasets is instrumental in the field of RAS research.

Chapter 3 proposed to model an RAS procedure as a hierarchical system of discrete surgical states that was referred to as the surgical Hierarchical Finite State Model (HFSM). This method views an RAS procedure from a temporal perspective by

decomposing it into a collection of surgical tasks/steps. Each of these surgical tasks/steps is further broken down into finer segments of surgical actions and gestures that were referred to as fine-grained surgical states. Modeling an RAS procedure as an HFSM allows it to be described more smoothly from a temporal perspective and prepares us for the development of deep learning-based hierarchical surgical state estimation models in a more systematic manner.

Chapter 4 proposed a novel deep learning-based model for the estimation fine-grained surgical states that accepts multiple types of RAS time series data as inputs. These input data include the endoscopic video, robot kinematics, and system events dataset from either a dVRK or a dVXi surgical robotic system. The proposed model (Fusion-KVE) is the first attempt at multi-input fine-grained surgical state estimation. Through the evaluation using realistic and complex RAS datasets that I collected, Fusion-KVE showed its superiority over state-of-the-art fine-grained surgical estimation models that only uses one type of input data. Additionally, it was shown that richer and more comprehensive information could be extracted by incorporating various types of input data, as they represents an RAS from diverse perspectives. Each type of input data therefore has its respective strengths and weaknesses in the recognition of certain fine-grained surgical states.

Chapter 5 proposed daVinciNet: a deep learning-based model that concurrently predict in real-time RAS instruments' end-effector trajectories and future surgical states for up to 2 seconds into the future. A new method for endoscopic video feature extraction was also proposed and applied. Instead of only extracting vision features from the raw endoscopic video frame, the Region of Interest (RoI) of the current endoscopic view (the surrounding areas of the surgical instruments' end-effectors) were determined. Regional vision features from the RoIs were used and proven to improve both the end-effector trajectory prediction and the surgical state prediction performances. It was shown that richer visual information can be extracted and utilized from the RoI of an endoscopic video frame, which contains concentrated information about the current fine-grained surgical states.

Chapter 6 further improved the fine-grain surgical state estimation efforts through the learning of an invariant latent representation of RAS time series data. Although new and more realistic RAS datasets were collected and used in this thesis work, deep learning-based surgical state estimator performances still suffer from the combination of limited data availability and high complexity. Additionally, the real-world RAS time series data is highly noisy, containing diverse nuisance factors

and surgical technique variances. The proposed model - StiseNet - achieves effective invariance induction to such factors through a novel adversarial model design and training schematics. Chapter 6 also further improved the efficiency of the endoscopic video feature extraction. A semantic mask of the current endoscopic view assigns each pixel of an endoscopic video frame to three categories, which greatly reduces the noise and diversity of the complex surgical backgrounds. The incorporation of such semantic masks to endoscopic video feature extraction reduced noises in the surgical background and further improved the fine-grained surgical state estimation performance.

Finally, Chapter 7 explores concurrent hierarchical surgical state estimation at multiple levels of temporal granularity. Through the incorporation of multiple types of input data and sophisticated fine-grained surgical state estimators developed in previous chapters, HESS-DNN and CHASSEN - the proposed models - achieved significant improvements to state-of-the-art methods in the surgical state estimations at two levels of temporal granularity when evaluated on the HERNIA-40 dataset. As the current surgical task/step and the current fine-grained surgical state are highly correlated, the usage of surgical state estimation results at one level of temporal granularity on the state estimation process at another level was also explored. The incorporation of both direct data sources (endoscopic video, robot kinematics, and system events) and inferred data sources (surgical state estimation results from other levels of temporal granularity) was shown to further improve the hierarchical surgical state estimation accuracy.

8.2 Opportunities for future work

While the previous chapters have presented novel contributions to temporal understanding and perception during an RAS procedure, which is an indispensable prerequisite for numerous AI applications in RAS, there are three areas of opportunities for future work that would further the contributions of this thesis work.

Over the past decade, enormous efforts have been devoted to computational-heavy fields of research such as machine learning, computer vision, artificial intelligence, autonomy, and many others. The field of surgical robotics research could benefit greatly from techniques, models, and algorithms resulting from the advancement in the aforementioned fields. This would allow robot-assisted surgeries to go beyond its current form of teleoperation and improve the quality of healthcare to many. Specifically, many AI applications could be developed to assist medical professionals

during and after RAS procedures. As mentioned previously, a comprehensive and accurate temporal awareness of the current surgical scene by the surgical robotic system is cardinal to many surgeon-assisting functionalities. The accuracy of the hierarchical surgical state estimation could therefore be further improved through the incorporation of more sophisticated and innovative deep learning-based model architectures and training strategies.

While this thesis propose a new hierarchical RAS modeling strategy with discrete surgical states, this strategy is mostly descriptive and has plenty room for refinement. Through our first attempt at using the hierarchical correlations between surgical estimation results at two levels of temporal granularity to improve the surgical state estimation performance, it has been shown that that surgical states at different levels of temporal granularity are highly correlated with each other. Our initial effort in Chapter 7 has also shown that more efficient hierarchical surgical state estimations could be achieved through the incorporation of such hierarchical correlations. Currently, however, the incorporation method implemented in this thesis work is elementary. More sophisticated methods that utilizes the correlations across hierarchy in a surgical HSFM could greatly benefit the surgical state estimation effort. The concepts of Graph Convolutional Network and hypergraph neural network have gained recent attentions in the field of deep learning research [10, 29]. Similar concepts and models, as they are being developed, have the potential of being applied to a more formal definition of a hierarchical HFSM.

Learning-based surgical state estimation efforts rely heavily on the datasets for training and evaluation. Although new and more realistic datasets were collected and used in this thesis work and have shown their significant contributions, their sizes and level of diversity is still extremely limited. Continuous effort in real-world RAS data collection, curation, and annotation is highly beneficial for the advancement in learning-based surgical state estimator development. So far, the only real-world RAS procedure used for hierarchical surgical state estimation is the hernia repair surgery. The application and expansion of surgical state estimations to other types of RAS procedures such as prostatectomy is highly beneficial and evaluates the state estimators in a more comprehensive manner. This is especially important in the safety-critical field of RAS research.

Our work has numerous future extensions and applications. The hierarchical surgical state estimation framework could be applied to diverse applications during and after an RAS procedure. These applications range from user interface integration,

to surgeon-assisting functionalities, to supervised semi-autonomous, or even autonomous surgical robotic systems. The future work on improving the hierarchical surgical state estimation accuracy and efficiency could therefore benefit the field of robot-assisted surgery greatly.

BIBLIOGRAPHY

[1] Alessandro Achille and Stefano Soatto. "Emergence of invariance and disentanglement in deep representations". In: *J. Machine Learning Research* 19.1 (2018), pp. 1947–1980.

[2] Narges Ahmidi et al. "A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery". In: *IEEE Transactions on Biomedical Engineering* 64.9 (2017), pp. 2025–2041.

[3] Alexandre Alahi et al. "Social lstm: Human trajectory prediction in crowded spaces". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 961–971.

[4] Alexander A Alemi et al. "Deep variational information bottleneck". In: *arXiv preprint arXiv:1612.00410* (2016).

[5] Max Allan et al. "3-D pose estimation of articulated instruments in robotic minimally invasive surgery". In: *IEEE transactions on medical imaging* 37.5 (2018), pp. 1204–1213.

[6] Beatrice van Amsterdam et al. "Weakly Supervised Recognition of Surgical Gestures". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 9565–9571.

[7] Michael A Arbib. "Theories of abstract automata". In: *Journal of Symbolic Logic* 37.2 (1972).

[8] George Awad et al. "Trecvid 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search". In: 2018.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[10] Song Bai, Feihu Zhang, and Philip HS Torr. "Hypergraph convolution and hypergraph attention". In: *Pattern Recognition* 110 (2021), p. 107637.

[11] Gabriel I Barbash. "New technology and health care costs–the case of robot-assisted surgery". In: *The New England journal of medicine* 363.8 (2010), p. 701.

[12] Debabrata Basu. "On the elimination of nuisance parameters". In: *Selected Works of Debabrata Basu*. Springer, 2011, pp. 279–290.

[13] Andrew P Bradley. "The use of the area under the ROC curve in the evaluation of machine learning algorithms". In: *Pattern recognition* 30.7 (1997), pp. 1145–1159.

[14] Fabian Caba Heilbron et al. “Activitynet: A large-scale video benchmark for human activity understanding”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 961–970.

[15] Preetham Chalasani et al. “A Computational Framework for Complementary Situational Awareness (CSA) in Surgical Assistant Robots”. In: *2018 Second IEEE International Conference on Robotic Computing (IRC)*. IEEE. 2018, pp. 9–16.

[16] Girish Chandrashekar and Ferat Sahin. “A survey on feature selection methods”. In: *Computers & Electrical Engineering* 40.1 (2014), pp. 16–28.

[17] Kyunghyun Cho et al. “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. In: *arXiv preprint arXiv:1406.1078* (2014).

[18] Kyunghyun Cho et al. “On the properties of neural machine translation: Encoder-decoder approaches”. In: *arXiv preprint arXiv:1409.1259* (2014).

[19] Tobias Czempiel et al. “TeCNO: Surgical Phase Recognition with Multi-Stage Temporal Convolutional Networks”. In: *Int. Conf. Med. Image Comp. and Comp.-Assist. Intervention*. Springer. 2020, pp. 343–352.

[20] Daniel DeMenthon and Remi Megret. *Spatio-temporal segmentation of video by hierarchical mean shift analysis*. Citeseer, 2002.

[21] Simon P DiMaio et al. *Interactive user interfaces for minimally invasive telesurgical systems*. US Patent App. 15/725,271. Feb. 2018.

[22] Li Ding and Chenliang Xu. “Tricornet: A hybrid temporal convolutional and recurrent network for video action segmentation”. In: *arXiv preprint arXiv:1705.07818* (2017).

[23] Yuan Ding et al. “Surgical Workflow Recognition Using Two-Stream Mixed Convolution Network”. In: *Int. Conf. Adv. Elec. Materials, Comp.s and Soft. Eng.* 2020, pp. 264–269.

[24] Robert DiPietro et al. “Recognizing surgical activities with recurrent neural networks”. In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2016, pp. 551–558.

[25] Robert DiPietro et al. “Segmenting and classifying activities in robot-assisted surgery with recurrent neural networks”. In: *International journal of computer assisted radiology and surgery* (2019), pp. 1–16.

[26] Yong Du, Wei Wang, and Liang Wang. “Hierarchical recurrent neural network for skeleton based action recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1110–1118.

[27] Ian Endres and Derek Hoiem. “Category independent object proposals”. In: *European Conference on Computer Vision*. Springer. 2010, pp. 575–588.

[28] M. Everingham et al. “The Pascal Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338.

[29] Yifan Feng et al. “Hypergraph neural networks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 3558–3565.

[30] Edward W Forgy. “Cluster analysis of multivariate data: efficiency versus interpretability of classifications”. In: *Biometrics* 21 (1965), pp. 768–769.

[31] Isabel Funke et al. “Using 3d convolutional neural networks to learn spatiotemporal features for automatic surgical gesture recognition in video”. In: *Int. Conf. Med. Image Comp. and Comp.-Assisted Inter.* 2019, pp. 467–475.

[32] Yixin Gao et al. “Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling”. In: *MICCAI Workshop: M2CAI*. Vol. 3. 2014, p. 3.

[33] Felix A Gers and Jürgen Schmidhuber. “Recurrent nets that time and count”. In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. Vol. 3. IEEE. 2000, pp. 189–194.

[34] Bernard Gibaud et al. “Toward a standard ontology of surgical process models”. In: *Int. J. Comp. Assist. radiology and surgery* 13.9 (2018), pp. 1397–1408.

[35] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[37] Ian Goodfellow et al. “Generative adversarial nets”. In: *Adv. Neural Info. Process. Systems*. 2014, pp. 2672–2680.

[38] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. “Hybrid speech recognition with deep bidirectional LSTM”. In: *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE. 2013, pp. 273–278.

[39] Thomas R Gruber. “Toward principles for the design of ontologies used for knowledge sharing?” In: *International journal of human-computer studies* 43.5-6 (1995), pp. 907–928.

[40] Bilge Gunsel, Yue Fu, and A Murat Tekalp. “Hierarchical temporal video segmentation and content characterization”. In: *Multimedia Storage and Archiving Systems II*. Vol. 3229. International Society for Optics and Photonics. 1997, pp. 46–56.

[41] Richard HR Hahnloser et al. “Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit”. In: *nature* 405.6789 (2000), pp. 947–951.

[42] Sepp Hochreiter. “Untersuchungen zu dynamischen neuronalen Netzen”. In: *Diploma, Technische Universität München* 91.1 (1991).

[43] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[44] I Hsu, Ayush Jaiswal, Premkumar Natarajan, et al. “Niesr: Nuisance invariant end-to-end speech recognition”. In: *arXiv preprint arXiv:1907.03233* (2019).

[45] Zhiheng Huang, Wei Xu, and Kai Yu. “Bidirectional LSTM-CRF models for sequence tagging”. In: *arXiv preprint arXiv:1508.01991* (2015).

[46] Fumito Ito, David Jarrard, and Jon C Gould. “Transabdominal preperitoneal robotic inguinal hernia repair”. In: *J. Laparoendoscopic & Adv. Surg. Techn.s* 18.3 (2008), pp. 397–399.

[47] Ayush Jaiswal et al. “Unified adversarial invariance”. In: *arXiv preprint arXiv:1905.03629* (2019).

[48] Ayush Jaiswal et al. “Unsupervised adversarial invariance”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 5092–5102.

[49] Yueming Jin et al. “SV-RCNet: workflow recognition from surgical videos using recurrent convolutional network”. In: *IEEE transactions on medical imaging* 37.5 (2017), pp. 1114–1126.

[50] Ankur Kapoor, Anton Deguet, and Peter Kazanzides. “Software components and frameworks for medical robot control”. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE. 2006, pp. 3813–3818.

[51] Andrej Karpathy et al. “Large-scale video classification with convolutional neural networks”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2014, pp. 1725–1732.

[52] Peter Kazanzides et al. “An open-source research kit for the da Vinci® Surgical System”. In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, pp. 6434–6439.

[53] Zhamak Khorgami et al. “Extra Costs of Robotic Surgery in Minor and Major Surgeries: An Analysis of National Inpatient Sample”. In: *Journal of the American College of Surgeons* 225.4 (2017), e86.

[54] Peter C Kim et al. *Automated surgical and interventional procedures*. US Patent 9,220,570. Dec. 2015.

[55] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[56] Tom Ko et al. “Audio augmentation for speech recognition”. In: *Sixteenth annual conference of the international speech communication association*. 2015.

[57] Julian Francisco Pieter Kooij et al. “Context-based pedestrian path prediction”. In: *European Conf. Computer Vision*. Springer. 2014, pp. 618–633.

[58] Sanjay Krishnan et al. “Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning”. In: *Robotics Research*. Springer, 2018, pp. 91–110.

[59] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).

[60] H. Kuehne et al. “HMDB: a large video database for human motion recognition”. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2011.

[61] Tian Lan et al. “Action recognition by hierarchical mid-level action elements”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4552–4560.

[62] Colin Lea, Gregory D Hager, and Rene Vidal. “An improved model for segmentation and recognition of fine-grained activities with application to surgical training tasks”. In: *2015 IEEE winter conference on applications of computer vision*. IEEE. 2015, pp. 1123–1129.

[63] Colin Lea, René Vidal, and Gregory D Hager. “Learning convolutional action primitives for fine-grained action recognition”. In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2016, pp. 1642–1649.

[64] Colin Lea et al. “Segmental spatiotemporal cnns for fine-grained action segmentation”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 36–52.

[65] Colin Lea et al. “Temporal convolutional networks: A unified approach to action segmentation”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 47–54.

[66] Erik Leijte et al. “Robot assisted versus laparoscopic suturing learning curve in a simulated setting”. In: *Surgical endoscopy* 34.8 (2020), pp. 3679–3689.

[67] Vladimir I Levenshtein. “Binary codes capable of correcting deletions, insertions, and reversals”. In: *Soviet physics doklady*. Vol. 10. 8. 1966, pp. 707–710.

[68] Junwei Liang et al. “Peeking into the future: Predicting future person activities and locations in videos”. In: *IEEE Conf. Computer Vision and Pattern Recognition*. 2019, pp. 5725–5734.

[69] Shujie Liu et al. *Hierarchical segmentation and quality measurement for video editing*. US Patent App. 15/173,465. Dec. 2016.

[70] Shugao Ma, Leonid Sigal, and Stan Sclaroff. "Learning activity progression in lstms for activity detection and early detection". In: *IEEE Conf. Computer Vision and Pattern Recog.* 2016, pp. 1942–1950.

[71] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. "Rectifier nonlinearities improve neural network acoustic models". In: *Proc. icml*. Vol. 30. 1. Citeseer. 2013, p. 3.

[72] Michael J Mack. "Minimally invasive and robotic surgery". In: *Jama* 285.5 (2001), pp. 568–572.

[73] Pietro Mascagni and Nicolas Padoy. "OR black box and surgical control tower: Recording and streaming data and analytics to improve surgical care". In: *Journal of Visceral Surgery* 158.3 (2021), S18–S25.

[74] M. Maschler et al. *Game Theory*. Cambridge University Press, 2013, pp. 176–180. ISBN: 9781107005488.

[75] Michael F Mathieu et al. "Disentangling factors of variation in deep representation using adversarial training". In: *Advv Neur, Inf. Process. Syst.s*. 2016, pp. 5040–5048.

[76] Effrosyni Mavroudi et al. "End-to-end fine-grained action segmentation and recognition using conditional random field models and discriminative sparse coding". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 1558–1567.

[77] Leland McInnes, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[78] Giovanni Menegozzo et al. "Surgical gesture recognition with time delay neural network based on kinematic data". In: *2019 International Symposium on Medical Robotics (ISMR)*. IEEE. 2019, pp. 1–7.

[79] Richard Meyes et al. "Ablation studies in artificial neural networks". In: *arXiv preprint arXiv:1901.08644* (2019).

[80] Riccardo Miotto et al. "Deep learning for healthcare: review, opportunities and challenges". In: *Briefings in bioinformatics* 19.6 (2018), pp. 1236–1246.

[81] Michael J Moore, Charles L Bennett, et al. "The learning curve for laparoscopic cholecystectomy". In: *The American journal of surgery* 170.1 (1995), pp. 55–59.

[82] Raj Mudunuri, Oliver Burgert, and Thomas Neumuth. "Ontological modelling of surgical knowledge". In: *Informatik 2009–Im Focus das Leben* (2009).

[83] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[84] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.

[85] Chinedu Innocent Nwoye et al. "Weakly supervised convolutional LSTM approach for tool tracking in laparoscopic videos". In: *International journal of computer assisted radiology and surgery* 14.6 (2019), pp. 1059–1067.

[86] Nicolas Padoy. "Machine and deep learning for workflow recognition during surgery". In: *Minimally Invasive Therapy & Allied Technologies* 28.2 (2019), pp. 82–90.

[87] Sofoklis Panteleimonitis et al. "Precision in robotic rectal surgery using the da Vinci Xi system and integrated table motion, a technical note". In: *Journal of Robotic Surgery* 12.3 (2018), pp. 433–436.

[88] Sahba Aghajani Pedram et al. "Autonomous suturing via surgical robot: An algorithm for optimal selection of needle diameter, shape, and path". In: *IEEE Int. Conf. Robotics and Automation*. 2017, pp. 2391–2398.

[89] Raghuveer Peri et al. "Robust speaker recognition using unsupervised adversarial invariance". In: *IEEE Int. Conf. Acoust., Speech Sig. Proc.* 2020, pp. 6614–6618.

[90] Mark Plutowski, Garrison Cottrell, and Halbert White. "Experience with selecting exemplars from clean data". In: *Neural Networks* 9.2 (1996), pp. 273–294.

[91] Yao Qin et al. "A dual-stage attention-based recurrent neural network for time series prediction". In: *arXiv preprint arXiv:1704.02971* (2017).

[92] Yidan Qin and Joel W Burdick. "Concurrent Hierarchical Autonomous Surgical State Estimation during Robot-assisted Surgery". In: (2022).

[93] Yidan Qin et al. "Autonomous Hierarchical Surgical State Estimation During Robot-Assisted Surgery Through Deep Neural Networks". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6220–6227.

[94] Yidan Qin et al. "davincinet: Joint prediction of motion and surgical state in robot-assisted surgery". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2921–2928.

[95] Yidan Qin et al. "Learning invariant representation of tasks for robust surgical state estimation". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3208–3215.

[96] Yidan Qin et al. "Temporal segmentation of surgical sub-tasks through deep learning with multiple data sources". In: *IEEE Int. Conf. Robotics and Automation*. 2020, pp. 371–377.

[97] Jeremy Rogers et al. “GALEN ten years on: Tasks and supporting tools”. In: *MEDINFO 2001*. IOS Press. 2001, pp. 256–260.

[98] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[99] Jacob Rosen et al. “Generalized approach for modeling minimally invasive surgery as a stochastic process using a discrete Markov model”. In: *IEEE Transactions on Biomedical engineering* 53.3 (2006), pp. 399–413.

[100] Peter J Rousseeuw. “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”. In: *J. Comp. Applied Math.* 20 (1987), pp. 53–65.

[101] Mohammad Sadegh Aliakbarian et al. “Encouraging lstms to anticipate actions very early”. In: *IEEE Int. Conf. Computer Vision*. 2017, pp. 280–289.

[102] Hiroaki Sakoe and Seibi Chiba. “Dynamic programming algorithm optimization for spoken word recognition”. In: *IEEE transactions on acoustics, speech, and signal processing* 26.1 (1978), pp. 43–49.

[103] Mario Selvaggio et al. “Passive virtual fixtures adaptation in minimally invasive robotic surgery”. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3129–3136.

[104] Azad Shademan et al. “Supervised autonomous robotic soft tissue surgery”. In: *Sci. Trans. Med.* 8.337 (2016), 337ra64–337ra64.

[105] Alexey A Shvets et al. “Automatic instrument segmentation in robot-assisted surgery using deep learning”. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2018, pp. 624–628.

[106] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[107] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. “UCF101: A dataset of 101 human actions classes from videos in the wild”. In: *arXiv preprint arXiv:1212.0402* (2012).

[108] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 1929–1958.

[109] C Staub et al. “Human-computer interfaces for interaction with surgical tools in robotic surgery”. In: *IEEE RAS & EMBS Int. Conf. Biomed. Robotics and Biomechatronics*. 2012, pp. 81–86.

[110] Ralf Stauder et al. “The TUM LapChole dataset for the M2CAI 2016 workflow challenge”. In: *arXiv preprint arXiv:1610.09278* (2016).

[111] Robert Stevens, Carole A Goble, and Sean Bechhofer. "Ontology-based knowledge representation for bioinformatics". In: *Briefings in bioinformatics* 1.4 (2000), pp. 398–414.

[112] Sarah B Stringfield et al. "Ten-Year Review of Robotic Surgery at an Academic Medical Center". In: *Journal of the American College of Surgeons* 225.4 (2017), S79.

[113] Lingling Tao et al. "Sparse hidden markov models for surgical gesture classification and skill evaluation". In: *International conference on information processing in computer-assisted interventions*. Springer. 2012, pp. 167–177.

[114] Lingling Tao et al. "Surgical gesture segmentation and recognition". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2013, pp. 339–346.

[115] Andru P Twinanda et al. "Endonet: a deep architecture for recognition tasks on laparoscopic videos". In: *IEEE transactions on medical imaging* 36.1 (2016), pp. 86–97.

[116] Andru Putra Twinanda et al. "RSDNet: Learning to predict remaining surgery duration from laparoscopic videos without manual annotations". In: *IEEE transactions on medical imaging* 38.4 (2018), pp. 1069–1078.

[117] Amin Ullah et al. "Action recognition in video sequences using deep bi-directional LSTM with CNN features". In: *IEEE Access* 6 (2017), pp. 1155–1166.

[118] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[119] Roy JJ Verhage et al. "Minimally invasive surgery compared to open procedures in esophagectomy for cancer: a systematic review of the literature". In: *Minerva chirurgica* 64.2 (2009), p. 135.

[120] Oliver Weede et al. "An intelligent and autonomous endoscopic guidance system for minimally invasive surgery". In: *IEEE Int. Conf. Robotics and Automation*. 2011, pp. 5762–5768.

[121] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big data* 3.1 (2016), pp. 1–40.

[122] Chenxia Wu, Ian Lenz, and Ashutosh Saxena. "Hierarchical Semantic Labeling for Task-Relevant RGB-D Perception." In: *Robotics: Science and systems*. 2014.

[123] Qizhe Xie et al. "Controllable invariance through adversarial feature learning". In: *Advances in Neural Information Processing Systems*. 2017, pp. 585–596.

[124] Takuma Yagi et al. "Future person localization in first-person videos". In: *IEEE Conf. Computer Vision and Pattern Recog.* 2018, pp. 7593–7602.

[125] Guang-Zhong Yang et al. “Medical robotics—Regulatory, ethical, and legal considerations for increasing levels of autonomy”. In: *Science Robotics* 2.4 (2017), p. 8638.

[126] Tong Yu et al. “Learning from a tiny dataset of manual annotations: a teacher/student approach for surgical phase recognition”. In: *arXiv preprint arXiv:1812.00033* (2018).

[127] Luca Zappella et al. “Surgical gesture classification from video and kinematic data”. In: *Medical image analysis* 17.7 (2013), pp. 732–745.

[128] Rich Zemel et al. “Learning fair representations”. In: *Int. Conf. Machine Learning*. 2013, pp. 325–333.

[129] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. “Mitigating unwanted biases with adversarial learning”. In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 2018, pp. 335–340.

[130] Alice Zheng and Amanda Casari. *Feature engineering for machine learning: principles and techniques for data scientists*. " O’Reilly Media, Inc.", 2018.

[131] Feng Zhou, Fernando De la Torre, and Jessica K Hodgins. “Aligned cluster analysis for temporal segmentation of human motion”. In: *2008 8th IEEE international conference on automatic face & gesture recognition*. IEEE. 2008, pp. 1–7.

[132] Feng Zhou, Fernando De la Torre, and Jessica K Hodgins. “Hierarchical aligned cluster analysis for temporal clustering of human motion”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.3 (2012), pp. 582–596.

[133] Xiao-Yun Zhou and Guang-Zhong Yang. “Normalization in training U-Net for 2-D biomedical semantic segmentation”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1792–1799.

[134] Aneeq Zia et al. “Surgical Activity Recognition in Robot-Assisted Radical Prostatectomy Using Deep Learning”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Ed. by Alejandro F. Frangi et al. Cham: Springer International Publishing, 2018, pp. 273–280. ISBN: 978-3-030-00937-3.

[135] Aneeq Zia et al. “Temporal clustering of surgical activities in robot-assisted surgery”. In: *International journal of computer assisted radiology and surgery* 12.7 (2017), pp. 1171–1178.