WORKFLOW ANALYSIS FOR LAPAROSCOPIC SURGERY

Yitong Zhang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Computer Science
University College London

Sept. 2024

Acknowledgement

I would like to express my deepest appreciation to my supervisor, Dr. Francisco Vasconcelos, for the guidance, support, and invaluable insights throughout the course of my research. I am also grateful to Prof. Danail Stoyanov and Dr. Sophia Bano, for their helpful feedback and encouragement.

Special thanks to my family and friends for their unwavering support and encouragement. A special thanks to my cat, Burrata and Croissant, for keeping me company during the long nights of writing and providing much-needed comfort. Lastly, I wish to acknowledge the financial support from Amazon CDI AWS Doctoral Scholarship, which made this work possible.

I, Yitong Zhang confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

This study investigated surgical workflow analysis by comparing frame-based and event-based methodologies. Initial research focused heavily on frame-wise classification and metrics. At the outset, we assessed traditional models against a proposed frame-based sequence-to-sequence model using both frame-based and event-based metrics across multiple datasets, including Cholec80 and an in-house Sacrocolpopexy dataset. Although conventional frame-based techniques generally performed well, they struggled with lengthy and complex surgical videos when evaluated on event-based metrics. Despite our proposed sequence-to-sequence model achieving superior event-based metrics, it still had limitations. Therefore, we introduced a new transition-retrieval configuration incorporating several innovative models: the TRN model for offline segmentation, and the enhanced ATRN model providing both offline/online segmentation and anticipation tasks. These methods showed improved performance in segmentation tasks by integrating transitions and minimizing frame-level noise. The study also underscored the importance of event-based metrics in capturing long-term temporal patterns and phase continuity, which is essential in the medical field.The Sacrocolpopexy dataset, which involves a unique type of surgery to current surgical workflow analysis benchmarks, contains surgeries of longer duration than those in existing benchmarks, thus increasing the likelihood of transition over-detection. We observed that the transition-retrieval configuration yields better results for this dataset. The research concluded that while traditional frame-based approaches are effective for quick evaluations, event-based metrics offer more detailed and accurate segmentation, which is essential for downstream surgical applications.

Impact Statement

The expertise and findings presented in this thesis can significantly impact both the medical field and broader technological advancements. By enhancing the accuracy and efficiency of laparoscopic surgery workflows through advanced machine learning models, our research offers several substantial benefits.

Firstly, the development of automated surgical workflow segmentation and anticipation tools can improve intraoperative decision-making, providing surgeons with real-time, data-driven insights. This leads to increased accuracy, reduced error rates, and enhanced patient outcomes. Furthermore, these tools streamline surgical processes, potentially shortening operation times and reducing patient recovery periods. This can translate to cost savings for healthcare systems and improved patient satisfaction.

Moreover, the adoption of Computer Assisted Intervention (CAI) systems incorporating our models facilitates continuous surgical training and proficiency assessments. By offering detailed, phase-specific feedback on surgical performance, these systems empower surgeons to refine their techniques and learn from retrospective analyses. As a result, the surgical community can achieve consistent improvements in skill levels and procedural standards.

Beyond the immediate benefits within the medical field, the insights gained from this research can stimulate further advancements in machine learning and artificial intelligence. The novel approaches to workflow segmentation and anticipation detailed in this work can be applied in other domains requiring temporal data analysis, such as industrial automation, robotics, and surveillance systems. By advancing the state-of-the-art in supervised deep learning, our research supports the ongoing development of intelligent, au-

tonomous systems capable of performing complex tasks with minimal human intervention.

In conclusion, the impact of this research extends from enhancing the quality and efficacy of minimally invasive surgeries to driving innovations in artificial intelligence and machine learning. These advancements foster a collaborative and more efficient healthcare environment, ultimately benefiting society by improving medical outcomes and supporting technological growth.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

- **CAI:** Computer Assisted Intervention
- **AI:** Artificial Intelligence
- **CNN:** Convolutional Neural Network
- **MLP:** natural language processing
- **LSTM:** Long Short-Term Memory

- **TCN:** Temporal Convolution Network
- **seq2seq:** sequence-to-sequence
- **m2m:** many-to-many
- **TRN:** Transition Retrieval Networ
- **EAD:** event analysis diagram
- **RL:** Reinforcement Learning
- **DQN:** Deep Q-Learning Network
- **FI:** Fixed Initialization
- **RMI:** ResNet Modified Initialization Initialization
- **PPO:** Proximal Policy Optimization
- **SAC:** Soft Actor-Critic
- **DDPG:** Deep Deterministic Policy Gradient
- **ATRN:** Aligned Transition Retrieval Network
- **TRA:** Transition Retrieving Agent

1 Introduction

Laparoscopic surgery, often referred to as a type of minimally invasive surgery, involves performing procedures via small cuts with the aid of a camera and specialized instruments. Due to the fact that minimally invasive surgery does not allow for direct access to the surgical site as traditional open surgery does, the procedure can be more complicated for the surgeon, and in some instances, it might take a longer time to complete. Thus, Computer Assisted Intervention (CAI) has the potential to become crucial in modern laparoscopic surgery. It provides surgeons with improved visualization, higher accuracy, and better operative control. The integration of CAI into laparoscopic procedures has revolutionized the field by facilitating minimally invasive surgeries that reduce patient recovery durations and enhance surgery success rates. [26, 33, 62, 66, 69, 92, 96]

During CAI, various types of data can be produced during operations, including instrument kinematics signals [47, 96], surgical gesture information [6, 8], and data from model-integrated force sensors [48, 70]. However, the most significant data generated is undoubtedly the vast amount of surgical video data.

With more retrospective surgical video data available for researchers, machine learning models and video-based surgical vision systems, which heavily rely on data, have made significant advancements. Generally, Machine learning methods can enhance surgical quality and efficiency by offering more accurate decisions [63, 89], quicker decision-making, and comprehensive patient status monitoring [36] during ongoing surgeries, or aid in retrospective analysis [12], quality control [54], and auditing of recorded surgical videos [39]. In addition to these general benefits, there are particular CAI applications that are highly advantageous in laparoscopic surgery.

In this study, we will focus on surgical workflow analysis, a key aspect of CAI that supports surgeons in various stages of an operation, including the preoperative, intraoperative, and postoperative phases. It offers a standardised timeline of an operation, as defined by its different phases. This helps in assessing surgical proficiency, improving surgical training, and providing essential data for audits and support systems [48]. As a consequence, these analyses provide surgeons with detailed insights into their performance, allowing for technique refinement, better patient outcomes, and providing real-time feedback during surgery to enhance decision-making and potentially lower the risk of complications [23] .

We identify two main computational tasks for analyzing surgical workflows: segmentation and anticipation. Surgical workflow segmentation breaks down a procedure into distinct phases or stages and can be classified into online and offline tasks. Online video segmentation processes frames as they are received in real-time, ideal for applications needing instant feedback. Offline video segmentation occurs after recording is complete, allowing for more complex algorithms aimed at higher accuracy and quality. Different methods are chosen based on the context. Surgical workflow anticipation forecasts and prepares the sequence of tasks during an operation, ensuring the necessary instruments, equipment, and staff are ready and keeping the team informed. Initially, this PhD research focused on segmentation in the first two studies and developed hybrid methods for both segmentation and anticipation in the third study.

The state-of-the-art in this domain is based on supervised deep learning. Given the temporal nature of this problem, the majority of the state-of-the-art models for surgical workflow segmentation can be decomposed into two components: feature extractor and feature classifier. The feature extractor

normally is a convolutional neural network (CNN) backbone that converts images or batches of images (clips) into feature vectors. Figure 1 provides an illustration of this setup.

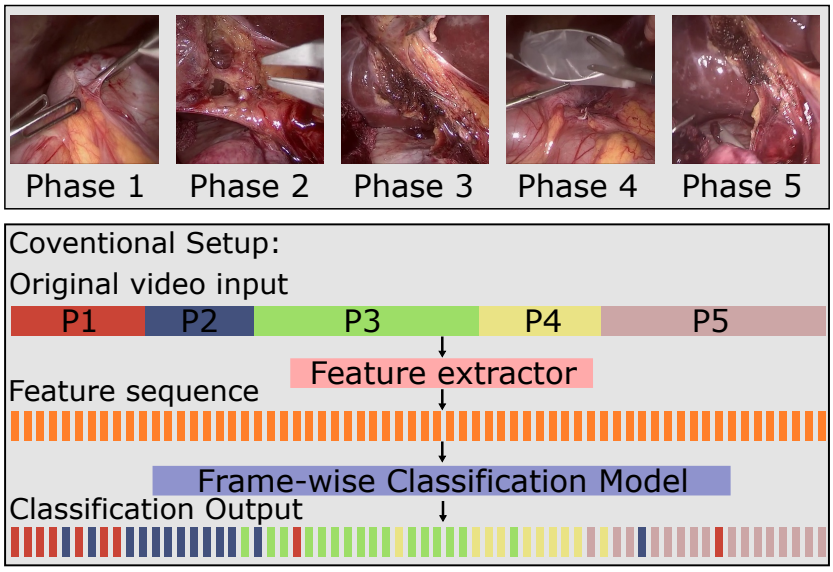


Figure 1: The conventional model configuration for workflow segmentation task

Most of the features extracted at this stage are spatial features or fine-level temporal features depending on the type of the input. Considering that long-term information in surgical video sequences aids the classification process, the following feature classifier predicts phases based on a temporally ordered sequence of extracted features.

At the beginning of this PhD research, the leading methods for automated workflow segmentation primarily relied on frame-wise multi-label classification. We assessed the performance of the then state-of-the-art model on an in-house laparoscopic sacrocolpopexy dataset (a different surgery that has not been evaluated in public benchmarks) comprising videos significantly longer than typical surgical datasets, using standard frame-wise eval-

uation metrics and commonly-used segmental metrics for general activity segmentation tasks. It was demonstrated that models with high accuracy are still prone to predict erroneous transitions randomly distributed within sequences. Specifically unexpected transitions can be easily detected at actual phase transitions. This phenomenon is particularly pronounced in long video sequences. Therefore, the subsequent work introduced an event-based phase-wise configuration, which was thoroughly validated against the state-of-the-art methods using both public benchmarks and our large-video in-house video dataset. The key contributions of this research are summarized in the later section.

1.1 Report Outline

This report is divided into 5 major chapters. Chapter 2 contains the literature review which provides the background information for the datasets and evaluation metrics related to this research. Also, a summary for the state-of-the-art AI techniques applied in this field is included in this chapter.

In Chapter 3, we conducted experiments on large-scale surgical video datasets using both frame-wise metrics (including accuracy, precision, recall, and F1-score) and event-based segmental metrics (a modified Ward Metric [97]), which consider the number of erroneous classification transitions. These experiments address the issue that conventional methods often underperform on event-based metrics, particularly with large videos. We proposed the use of a sequence-to-sequence architecture, which to our knowledge, is the first application of such an approach to workflow segmentation. An ablation study was performed on the proposed model. The results were presented using the Cholec80 dataset and the in-house dataset.

Although the sequence-to-sequence network showed improved performance on event-based metrics, it remains essentially a frame-wise classification

model and cannot completely avoid over-segmentation, particularly near transitions. In Chapter 4, we propose the Transition Retrieval Network (TRN) with a novel workflow segmentation configuration that emphasizes event-based metrics. This model focuses on identifying transitions between phases directly, rather than labeling each individual frame. Reinforcement learning is applied in this model. To maintain consistency with activity segmentation tasks, we use standard segmental metrics such as EDIT and F1@k to evaluate performance on the Cholec80 and in-house datasets.

Chapter 4 has demonstrated the efficacy of transition detection over frame-specific classification in surgical workflow segments. Nonetheless, the arrangement remains rudimentary. The output of the proposed approach is restricted to discrete action spaces, which is less computationally efficient, and the method can only handle offline video processing. In chapter 5, we have enhanced the transition-retrieval configuration with improvements to address these concerns. Additionally, by enabling the model to act within a continuous action space, similar to the format used in workflow anticipation tasks,we designed a hybrid model capable of concurrently performing workflow segmentation and anticipation. The performance of this hybrid model has been comprehensively evaluated on three datasets using frame-based and event-based segmental metrics.

Finally, a conclusion on current work is presented in Chapter 6.

1.2 Contribution

Chapter 3 have contributed to the following publication at IJCARS and will be referenced as [J1].

- ZHANG, Yitong, et al. Large-scale surgical workflow segmentation for laparoscopic sacrocolpopexy. International Journal of Computer Assisted Radiology and Surgery, 2022, 1-11.

The contributions of [J1] are:

- A general sequence-to-sequence temproal model formulation of the surgical workflow segmentation problem and several implementations with different configurations (time-synchronous and time-shifted), architectures (LSTM [41] and transformer [95]) and learning strategies. The time-shifted configuration has the advantage of not requiring a fine-level initialisation beyond the first few frames of a video.

- Introducing workflow segmentation in the context of laparoscopic sacrocolpopexy, with its significant challenges in terms of large and highly varying phase duration. These differences are also highlighted in comparison with the widely used benchmark Cholec80.

- An event-based evaluation methodology for surgical workflow that complements standard classification metrics to inform on potential workflow applications such as automated time-stamping of events.

Chapter 4 have contributed to the following publication at MICCAI 2022 and is referenced as [C1]. The contributions of [C1] are:

- We propose a novel formulation for surgical workflow segmentation based on phase transition retrieval. This strictly enforces that surgical phases are continuous temporal intervals, and immune to frame-level noise.

- We propose Transition Retrieval Network (TRN) that actively searches for phase transitions using multi-agent reinforcement learning. We describe a range of TRN configurations that provide different trade-offs between accuracy and amount of video processed.

- We validate TRN both on the public benchmark Cholec80 and on an in-house dataset of laparoscopic sacropolpopexy, where we demonstrate a single phase detection application.

Chapter 5 is under review and is referenced as [J2]. The contributions of [J2] are:

- We introduce the novel architecture ATRN, to simultaneously retrieve all phase transition timestamps in a surgical video, while only processing a fraction of its frames. The multi-purpose nature of ATRN enables application to both online/offline phase segmentation and online phase anticipation.

- Our online/offline segmentation models achieved superior performance in segment-level metrics (edit score) compared to the state-of-the-art methods, enabling more accurate prediction of the exact start and end timestamps of each phase, while keeping competitive frame-level performance (f1-score).

- Our offline segmentation model significantly reduces computational costs with respect to state-of-the-art by processing only a fraction of the whole video.

- Our online anticipation model has superior performance when compared to the state-of-the-art. We also demonstrate that accurate anticipation can be achieved from a small window of frames.

- We showcase the effectiveness of our method across three datasets: two public benchmarks (Cholec80, Cataract101) and an in-house dataset of laparoscopic sacropolpopexy (Sacro56).

2 Background

In this chapter, we provide a comprehensive background on surgical workflow analysis. We provide a overview of the essential machine learning configurations for analyzing surgical workflows and examined the latest literature in this area. In addition, we review and compare the datasets and evaluation metrics used in both general activity recognition tasks and the surgical workflow analysis domain, highlighting and identifying the differences between them.

Activity or action segmentation refers to the process of dividing a continuous sequence of actions or activities into distinct segments, each representing a specific activity. This technique is widely used in various fields. Applications of activity segmentation include video surveillance for security purposes, activity monitoring for elderly or patients in healthcare, sports analytics, and improving user experience in interactive systems. [1,2,34,82]. Surgical workflow segmentation, being a subset of activity segmentation, utilizes similar techniques and some of the same evaluation metrics as those employed in general activity segmentation tasks.

This chapter begins with a literature review that encapsulates the current advancements in surgical workflow analysis, associated with sections summarize both the datasets and evaluation metrics employed to assess performance across various methodologies. The chapter concludes with a discussion on the limitations of existing approaches, datasets, and metrics, which serve as the impetus for this research.

It is noteworthy that genuine clinical evidence in the field of surgical workflow analysis remains scarce. While metrics for activity segmentation tasks are often presented, it is uncommon for researchers to assess their methods us-

ing clinical metrics. Certain economic factors and ethical consent challenges could serve as major barriers in assessing methods based on clinical metrics. While our study focuses on comparing activity segmentation metrics, we aim to highlight the significance of incorporating actual clinical metrics—such as complication rates, blood loss, mortality, and readmission rates—as a crucial future direction for workflow analysis in this domain.

2.1 Surgical workflow analysis

In the introduction, two primary tasks of surgical workflow analysis were identified: surgical workflow segmentation and surgical workflow anticipation. Segmentation seeks to divide the entire surgical procedure into various temporal phases, whereas anticipation predicts the time remaining until the forthcoming surgical phases. The segmentation task is further categorized into offline and online approaches. Clinically, offline segmentation allows for the automation of retrospective audits and the review of recorded surgical procedures [103]. On the other hand, online segmentation coupled with phase anticipation can deliver real-time contextual data to the surgical team and support other subsequent algorithms, such as instrument navigation, risk assessment, and skill analysis [68, 80].

In the majority of image-guided surgeries, the main source of data for workflow analysis is video. The standard model architecture to handle this task in computer vision can be split into two primary components: a feature extractor that transforms frames of images into high-level, predominantly spatial feature vectors, and a subsequent sophisticated model that interprets a sequence of feature vectors to generate the necessary segmentation or prediction output. The configuration specifics were outlined in the introduction section with the Fig. 1.

Numerous studies have leveraged additional data sources alongside video, such as instrument trajectories obtained from robotic joint kinematics [7,57] or activity signals from surgical tools [98]. Some research relies on tool detection as an initial step for segmenting surgical phases. Despite the efficacy observed in surgeries like cholecystectomy [13, 44, 94] and cataracts [106], this approach demands precise tool labels, which necessitate laborious annotation efforts or extra tracking procedures in the operating room. The applicability of tool/kinematic signals is limited to specific types of surgeries, whereas video-only sources can be generalized for all surgeries, including our in-house sacrocolpopexy dataset. Therefore, this study restricts itself to utilizing purely video-based input signals.

2.1.1 Surgical Workflow Phase segmentation

Early studies in surgical phase segmentation relied on handcrafted feature extraction. Linear statistical models such as Hidden Markov Models (HMMs) [75], Conditional Random Fields (CRFs) [59], and Dynamic Time Warping (DTW) [11, 55] were then employed to capture the temporal relationships between extracted features across the entire surgical video. However, these approaches had limited generalisability and ability to represent the complex temporal dynamics.

With the advent of deep learning, Convolutional Neural Networks (CNNs) became the predominant technique to extract high-level spatial features from visual frames, and in particular ResNet has been widely adopted as a feature extractor for workflow segmentation. Recent research has mainly focused on exploring different temporal classification models that operate on top of extracted CNN features to provide temporal context, including

Long Short-Term Memory (LSTM) [44,94], Temporal Convolution Networks (TCN) [17, 27], and Transformers [18, 31]. More recently, Gated Recurrent Unit (GRU) has been demonstrated to compete with state-of-the-art in this task, despite being a relatively simple and old model [19, 40].

Recent research has increasingly focused on the hierarchical structures of temporal information in workflow analysis, proposing various methodologies to improve surgical workflow recognition by capturing complex multi-scale temporal patterns and enhancing feature learning. For instance, TMRNet [45] leverages a long-range memory bank and a temporal variation layer, while the segment-attentive hierarchical consistency network (SAHC) [24] emphasizes high-level segment information and a hierarchical segment-frame attention module. Additionally, a multi-stage architecture [100] and SKiT [65], a fast Key Information Transformer, have been introduced to further enhance performance and capture global information efficiently.

Additionally, timestamp supervision and uncertainty-aware temporal diffusion have been utilized to reduce manual annotation costs and improve surgical phase recognition performance by generating trustworthy pseudo labels from single timestamp annotations and diffusing them to adjacent frames based on uncertainty scores. [25] Moreover, GLSFormer, a transformer-based model for surgical step recognition in videos, employs spatio-temporal attention, a two-stream model, and a gating module to capture long-range dependencies, outperforming existing methods on cataract surgery video datasets [87]. Lastly, a weakly supervised temporal convolutional network approach has been proposed for fine-grained surgical activity recognition by utilizing phase annotations to train an end-to-end spatio-temporal model for step recognition and introducing a phase-step dependency loss to enforce weak supervision. [78] Together, these advancements highlight the significant strides being made in improving the accuracy and efficiency of surgical

workflow recognition through sophisticated hierarchical and temporal analysis techniques.

2.1.2 Surgical Workflow Phase Anticipation

Compared to workflow segmentation, there has been far less work on workflow anticipation. Since the outputs for anticipation are typically continuous values, a regression model is generally used for this task instead of a classification model. This distinction adds more challenges to both the training and evaluation processes compared to the segmentation task. Early anticipation studies predicted the remaining duration of the ongoing phase [29,50], while recent works forecast different upcoming phases [80, 101]. Some recent studies have combined surgical phase segmentation and anticipation in joint frameworks. TransSV employed multi-task learning, with a shared encoder and separate decoders for each task [46]. Attention mechanisms have been used to integrate contextual information. While offering both functionalities, these hybrid models still rely on frame-by-frame classification. On the other-hand, this paper introduces further accuracy gains with our model focused on phase transitions.

2.2 Datasets

The tasks of activity recognition and segmentation have been explored across various fields, such as cooking, manufacturing, and sports, using several well-known datasets like 50 Salads [90], Breakfast [53], EPIC-Kitchen [20], Assembly101 [86] and MultiThumos [99]. However, surgical procedures possess distinct features and challenges that necessitate a focused study. Many

types of surgeries exhibit clearly defined temporal phases (e.g., cholecystectomy [74], robotic prostatectomy [5], cataracts [93]), in which specific actions are performed with certain instruments targeting specific anatomical regions at particular times. The same procedure may be carried out by surgeons with varying expertise at different times (e.g., a consultant might take over from a junior trainee during a more sensitive part of the operation). These scenarios create a unique temporal context vocabulary for surgical procedures, which allows for specialized modeling of temporal phases and events for workflow analysis [32]. Some datasets that capture these surgical characteristics include cholecystectomy (Cholec80) [94], cataracts (CATRACT101) [106], micro-surgical anastomosis on artificial blood vessels (MISAW) [42], or general surgeon action detection (ESAD) [9]. We have compiled the details of these datasets in Table 1 to summarize and compare the similarities and differences across the datasets used in these domains.

Additionally, an overview of the Cholec80, Cataract, and Sacrocolpopexy datasets is presented below, as these are the primary datasets employed in this study for assessing the models' performance.

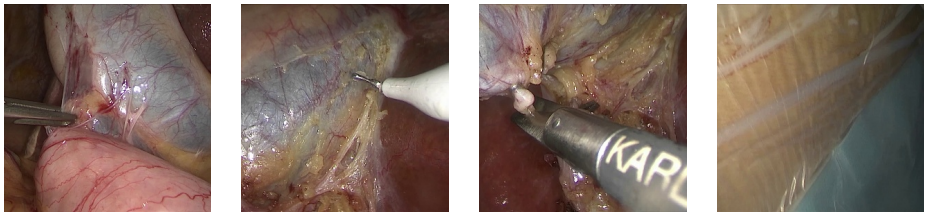
2.2.1 Cholec80

Cholec80 is one the most widely used datasets for surgical workflow analysis [13, 94] and tool segmentation [81] . It contains 80 videos of cholecystectomy surgeries performed by 13 surgeons. The videos are captured at 25 fps. There are 7 phases in this surgery which are: 1)Preparation; 2) Calot triangle dissection; 3) Clipping and cutting; 4) Gallbladder dissection; 5) Gallbladder packaging; 6) Cleaning and coagulation; 7) Gallbladder retraction. The average duration of this surgery is 38min. Figure 2 shows some

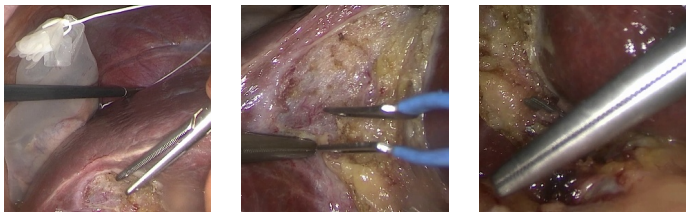
Table 1: Summary of datasets information for surgical workflow and general action segmentation

	Dataset	Number of videos	Average duration	Number of phases/actions	Benchmark Metrics	Type of Surgery/ Dataset infomation
Surgical Workflow	Cholec80	80	38min26s	7	Acc.,F1	Cholecystectomy
	M2CAI16	41	37min	7	Acc.,F1	Cholecystectomy
	JIGSAWS	39 Suturing, 36 Knot-Tying, and 28Needle-Passing	2min	15-20	EDIT, Acc.	Skill Assessment Working Set
	CATARACTS	50	10min56s	21	Acc.,F1	Cataract
	Cataract101	101	8min9s	11	Acc.,F1	Cataract
	Pitvis	25	1h20min	14	EDIT, Acc.,F1	Pituitary surgery
	HeiChole	30	35min	7	F1	Cholecystectomy
In-house Surgical Dataset	Sacrocolpopexy	56	3h19min57s	7	Acc.,F1,Ward, EDIT, F1@k	Sacrocolpopexy
General Action Segmentation	Breakfast	52	1h29min50s	10	F1@k, Acc., EDIT	webcams, standard industry cameras and stereo camerauncaliberated camera
	MultiThumos	400	4min30s	65	mAP	1.5 labels per frame, 10.5 actions per video
	EPIC-Kitchen-100	700	8min34s	97 verb classes, 300 noun classes	mAP, nDCG	Head-mounted camera, Multi-language narrations
	50 Salads	50	4min48s	3 high-level activity, 17 Low-level activity	F1@k, Acc., EDIT	accelerometer and RGB-D video data
	Assembly101	362 recodings with 12 views per recording	7min6s	Fine-grained actions: 24 verbs and 90 objects, Coarse actions: 11 verbs and 61 objects	F1@k, EDIT, MoF	multi-view action dataset

example images for each phase in cholecystectomy surgery.



(a) Phase 1 (b) Phase 2 (c) Phase 3 (d) Phase 4

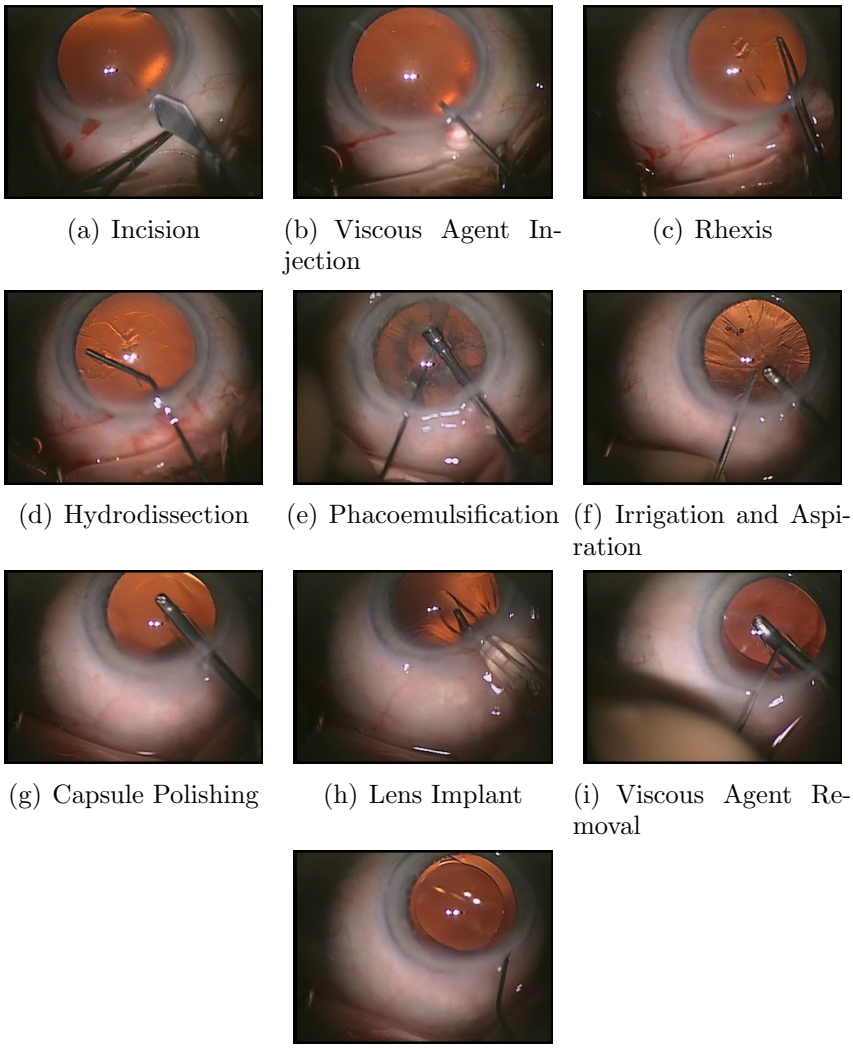


(e) Phase 5 (f) Phase 6 (g) Phase 7

Figure 2: Surgical phases of cholecystectomy surgery: 1)Preparation; 2) Calot triangle dissection; 3) Clipping and cutting; 4) Gallbladder dissection; 5) Gallbladder packaging; 6) Cleaning and coagulation; 7) Gallbladder retraction.

2.2.2 Cataract-101

Cataract101 [84] is a public dataset consisting of 101 cataract surgery videos performed by different surgeons over a 9-month period. The surgical sites and anatomy structures involved in this dataset are very different from those of laparoscopy. In cataract surgery, the surgical site involves the front part of the eye, known as the cornea. A small incision is made at the edge of the cornea to allow access to the lens. Through this incision, the clouded lens (cataract) is removed and replaced with a clear artificial lens. Displayed below are the illustrative images depicting the stages of this surgical procedure, as mentioned in [84]:



(a) Incision (b) Viscous Agent Injection (c) Rhexis

(d) Hydrodissection (e) Phacoemulsification (f) Irrigation and Aspiration

(g) Capsule Polishing (h) Lens Implant (i) Viscous Agent Removal

(j) Tonifying and Antibiotics

Figure 3: Surgical phases of cataract surgery

This surgery is divided in up to 10 phases and their duration is significantly shorter compared to the other utilised datasets. After trimming out the non-phase portions of the surgical video, the average procedure duration is 8 minutes.

2.2.3 Laparoscopic Sacrocolpopexy

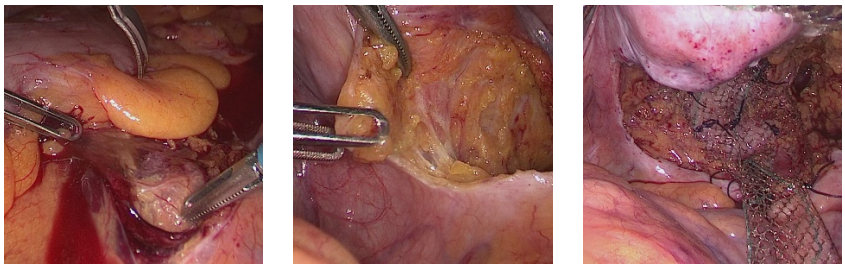
In addition to public benchmarks, we examined the workflow analysis using an in-house dataset focusing on laparoscopic sacrocolpopexy surgery. Laparoscopic sacrocolpopexy is considered the gold standard for treating vaginal vault prolapse [15]. All videos in this dataset were collected by the same team of clinicians from UZ Leuven Hospital, Belgium. In scenarios where trainee surgeons work under the guidance of a trainer surgeon, they gain experience as they participate in more procedures.

This learning process prompted the clinician team to ask: How long does it take to train a surgeon to perform sacrocolpopexy at the expert level? The clinician team investigated the learning curve for proficiency in laparoscopic sacrocolpopexy surgery [16]. The results revealed that the most time-consuming step is the dissection of the vault, which required 31 procedures for the trainee to attain an operation time comparable to the instructor. In addition, the quality of the dissection improved with time. The suturing of the implant to the vault and peritonealisation required only 10 and 6 procedures, respectively. The measurement of durations in lengthy surgical videos is labor-intensive, and automation through machine learning can help reduce the effort required.

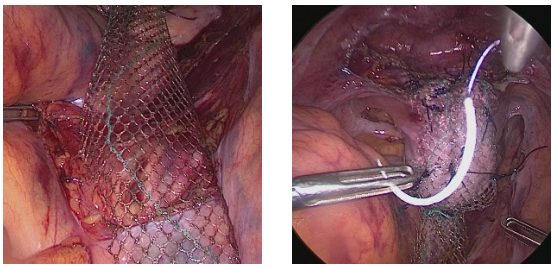
When compared to public datasets like Cholec80, which has an average duration of 38 minutes, and Cataract101 with an average of 8 minutes, the sacrocolpopexy surgeries are substantially longer, averaging 3 hours and 10 minutes. This extension in surgical duration raises several rarely discussed issues: (1) The conventional workflow segmentation configuration, in which the feature extractor processes each frame, becomes progressively more computationally expensive for longer surgeries, particularly when a large volume

of videos is processed for offline tasks. Minimizing the computational demands associated with feature extraction can substantially lower the computational expenses for offline activities.(2) Frame-wise classification outputs inevitably contain noise. Even with the same performance in frame-wise metrics (e.g., accuracy), the issue of overpredicting the presence of transitions may be more pronounced in lengthy sequences. Therefore, a more comprehensive evaluation metric is required to analyze performance in large surgical videos. (3) In practice, surgeons need more preparation and longer rest periods during lengthy surgeries, which may include frames where the laparoscopic camera is removed from the patient, entirely altering the visual context. Properly handling these non-phase frames is another open question that could affect the performance of methods for segmenting long surgical videos.

This in-house dataset contains 14 videos (Sacro14) of laparoscopic sacrocolpopexy surgery in the first research of Sequence-to-sequence network in Chapter 3. The dataset expanded to 38 videos (Sacro38) during the TRN study as discussed in Chapter 4, and ultimately reached 56 videos (Sacro56) in the third study on ATRN presented in Chapter 5. Two surgeons operated simultaneously with one of them manipulating the laparoscopic instruments and the other controls the tissues.



(a) Phase 1 (b) Phase 2 (c) Phase 3



(d) Phase 4 (e) Phase 5

Figure 4: Surgical phases of laparoscopic sacrocolpopexy: 1) promontory preparation; 2) dissection of vault and gutter; 3) mesh fixation to vault; 4) mesh fixation to promontory; 5) peritonealisation.

The videos are acquired at 24 fps resolution with a display resolution of 1920×1080 pixels. Most videos captured complete procedures, where the average duration was 3 hours 13 minutes with the shortest video of 1 hours and 47 minutes and the longest video of 4 hours 56 minutes. Each video was annotated by an expert Gynaecologist to indicate the start, the end and any pausing and resuming of each phase as timestamps.

Recently, the clinical team has introduced several enhancements in performing laparoscopic sacrocolpopexy. Surgical robots are employed for this procedure. Furthermore, traditional suturing has been substituted with a new gluing technique. The impact of these changes on patient outcomes and the reliability of the surgical workflow analysis model is still uncertain. While this research does not allow sufficient time to address these issues compre-

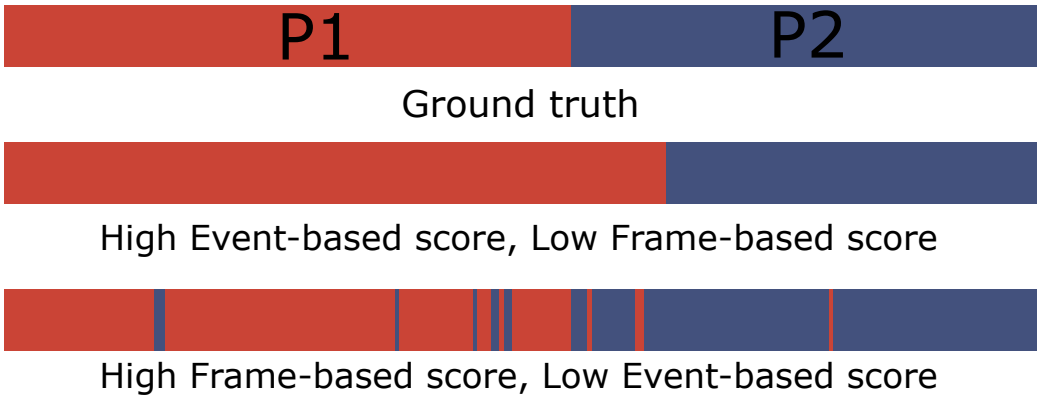


Figure 5: Example in differences between frame-based metrics and event-based metrics

hensively, they warrant further investigation based on the findings of this study.

2.3 Evaluation metric

In general, evaluation metrics are classified into two types: frame-based segmental evaluation metrics and event-based evaluation metrics. Frame-based metrics gauge model predictions using frame-level performance indicators such as accuracy and F1 score. On the other hand, segmental metrics assess sequences by considering the continuity of phases or steps. Funke, I. et.al. [30] highlights the inconsistencies found in the evaluation processes of various surgical phase recognition methods, particularly those evaluated on the Cholec80 benchmark. An example showing the difference between the two types of metrics is shown in Figure 5. In our research, we chose accuracy and F1 score, which are the metrics most commonly used in surgical workflow segmentation, and we examined a series of segmental(event-based) metrics to evaluate the segmental behavior of the sequences.

2.3.1 Frame-based metric

The most common evaluation metrics for workflow analysis are the measurement of video-based accuracy, phase-based precision and recall. These measurements are easy to calculate and provide a quick intuition on the performance of the models. In our work, we provide macro-averaged (per phase) precision and recall, F1-score calculated through this precision and recall, and micro-averaged accuracy. Macro-average treats all phases equally by computing the metric independently for each class and then averaging the results. Micro-average aggregates the contributions of all classes to compute the metric globally by considering all instances together. The equations for these metrics are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

TP denotes true positives, TN denotes true negatives, FP denotes false positives, and FN denotes false negatives.

2.3.2 Event-based metric

In light of the preceding study, it has been observed that the assessment and comparison of surgical workflow segmentation, viewed as an action segmentation task, predominantly rely on frame-level metrics. These metrics, however, do not entirely capture the effectiveness of the methods used in this time-based action segmentation task. Human understanding to a se-

quence is a complex problem as we are not only analysing the precision and recall but also the continuity of actions. Segmentation metrics outperform frame-level metrics in scenarios such as: 1. Evaluating the structure and order of actions in surgical procedures, offering a holistic view unlike frame-level metrics, which focus on individual frames [77]. 2. Tasks requiring clear transitions between actions, where event-based metrics capture changes more effectively than frame-level metrics, which consider frames separately [105]. 3. Long-duration activities with dispersed action points, where segmentation metrics provide a more accurate assessment over the entire period, in contrast to frame-level metrics that may focus on specific moments [3]. 4. Reverting to an earlier phase is an unfavorable indication in surgery, but it is difficult to recognize within the frame-based evaluation metric. In this study, we utilize three segmental metrics to address the previously mentioned drawbacks of the frame-based metric. These metrics include Ward's metric, EDIT score, and F1@50.

An event in action segmentation can be defined as continuous positive labels with a start time and a stop time. Ward define 5 types of error by comparing the predicted sequence to the ground truth sequence as: deletion(D), insertion(I′), merge(M, M′), fragmentation(F, F′), and Fragmented and Merged(FM, FM′), where the prime symbol indicates the segment in the predicted sequence. Figure 6 shows an example sequence with the each type of error where C stands for correctly predicted.

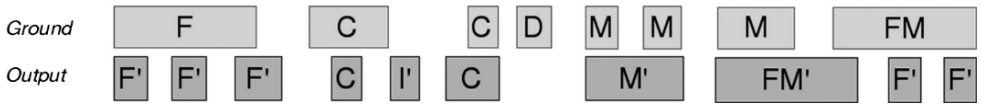


Figure 6: Event-based Ward Metric Error Definition [97]

Ward Metric The Ward metric counts each type of error individually and summarizes them in an event analysis diagram (EAD). In our research, as a multi-class classification problem, we implemented the ward metric phase-by-phase and added them up to obtain the final evaluation of a single sequence. We define event ratio as $\frac{E_{gt}}{E_{det}}$ where E_{gt} is the number of ground truth events, and E_{det} is the number of detected events by each method. We define a second ratio based on the event evaluated as correct in event-based Ward metric. We denote the Ward event ratio as $(\frac{C}{E_{gt}})$. For both of these ratios, values closer to 1 indicate better performance.

Edit Score In various fields, there is considerable uncertainty about when one action ends and another begins. In applications such as surgical skill assessment, the order of actions might be more crucial than exact temporal segmentation. In practice, the edit score metric [60] penalizes less for the timing offset of an action. For each sequence, the segmental labels G' and P' are designated for the original ground truth G and the prediction sequence P. For example, if $G = \{[A], [BBBB], [CC]\}$, then $G' = \{ABC\}$. The segmental edit score is described as a normalized edit distance, denoted $s_e(G', P')$, involving insertions, deletions, and substitutions. where

$$s_e(G', P') = \frac{e(G', P')}{\max(|G'|, |P'|)} \tag{5}$$

The function e is the Levenshtein distance. Levenshtein distance is a metric for measuring the difference between two sequences. It is the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one segment into the other. Finally, the edit score is cal-

culated as $(1 - s_e(G', P')) \cdot 100$, with 100 indicating the best score and 0 representing the worst outcome.

F1@k When delving into action detection research, it's quite common to encounter works that rely on mean Average Precision (mAP) paired with an intersection over union (IoU) overlap criterion, often denoted as mAP@k [79]. This metric, mAP@k, is determined by comparing the IoU overlap score for each action segment against the ground truth action of the same category. If an IoU score is above a threshold of k percent it is considered a true positive otherwise, it is deemed a false positive. Researchers calculate the average precision for each class and then average these results to derive the mAP value. This metric proves to be particularly beneficial for information retrieval tasks such as video searching. However, it has limitations, where the mAP is very sensitive to a confidence score assigned to each segment prediction. Colin L et al. [58] improved the mAP@k metric into the F1@k to address this problem. Similarly to mAP detection scores, F1@k compares the IoU with the ground truth using a specified threshold k. If more than one correct detection exists within the span of a single true action, only one is considered a true positive, whereas all others are marked as false positives. Then the precision and recall for true positives, false positives, and false negatives aggregated across all classes are computed. In the end, F1@k is calculated by $F1@k = \frac{2 * Precision@k * Recall@k}{Precision@k + Recall@k}$.

The F1@k score possesses several key characteristics: (1) it penalizes errors related to over-segmentation, (2) it does not penalize slight temporal misalignments between predictions and the actual data, which might occur due to annotation inconsistencies, and (3) it bases scoring on the number of actions rather than the length of each action instance. Although this metric

is similar to mAP@k, it does not require confidence in individual predictions. As shown in [58], qualitatively, the F1@k is better at capturing the quality of specific segmentations than mAP@k.

2.3.3 Anticipation Metric

The workflow anticipation task enables the model to predict the exact remaining time until the next phase begins.Consequently, we follow the methodology detailed in [80], using frame-based evaluation metrics like the mean absolute error (MAE) along with its variations: iMAE and eMAE. The representations for iMAE and eMAE are as follows:

$$iMAE = \frac{1}{T}\sum_{i}^{T} MAE(f_i, y_i), 0 < y < h \tag{6}$$

$$eMAE = \frac{1}{T}\sum_{i}^{T} MAE(f_i, y_i), 0 < y < 0.1h \tag{7}$$

Given T as the sequence length, f_i representing the estimated remaining time, and y_i being the actual remaining time at the present timestamp, h denotes the anticipation threshold.

2.4 Discussion

As indicated in Table 1, the distinctions between surgical datasets and general action segmentation datasets are evident. Generally, general action segmentation datasets are more extensive than surgical datasets, offering a larger number of training samples due to the limited number of videos in surgical datasets. Furthermore, surgical datasets feature fewer actions compared to general action segmentation datasets. While surgical datasets usually segment videos into predefined steps or phases of surgeries, general

action segmentation datasets aim to make complex predictions by identifying both actions (verbs) and objects (nouns) under certain conditions. The video content also varies; in surgical datasets, the background is often similarly coloured to the operating area, with blood or smoke that may cause occlusions. Conversely, in general action segmentation datasets, scenes differ greatly for various actions and have distinct features for identification, though this can vary based on the specific task and dataset context.

Meanwhile, the commonly used benchmark metrics differ between general action segmentation and surgical phase segmentation. In action segmentation, segmental metrics like EDIT or F1@k are often employed for network performance evaluation. Contrarily, in the surgical phase segmentation domain, accuracy and F1 score remain the predominant metrics for performance assessment. This frame-by-frame approach is generally simpler to implement and can be more efficient since each frame is processed independently. However, it may fail to capture the temporal relationships between frames, which are crucial for understanding the progression of a surgical procedure. This might deem detected erroneous transitions as a less significant issue, while the offset of predicted phases can be heavily penalized in the metrics. Conversely, event-based methods treat a sequence of frames as a single entity (or event), making predictions based on the entire sequence. This approach captures the temporal dynamics and dependencies between surgical phases or steps, offering a more comprehensive understanding of the surgical procedure. Therefore, in the subsequent studies within this thesis, event-based metric performance is also considered a key factor in method evaluation.

The vision-task-based metrics reviewed in this section serve to assess the effectiveness of models used in the analysis of surgical videos. These metrics are essential to determine how effectively models can identify and predict

surgical phases and actions, ensuring that the sequence and timing of steps are meticulously recorded. Nevertheless, analyzing surgical workflows, as a task with clinical significance, demands more than mere model performance evaluation in a broad context. Clinical metrics such as operative duration, blood loss, complication rates, readmission rates, and patient-reported outcomes are centered on the overall quality and safety of surgical procedures. These metrics evaluate the efficiency, safety, and effectiveness of surgeries, offering insights into patient outcomes and the caliber of care provided. Both categories of metrics are vital for advancing surgical practices, with vision-task-based metrics enhancing real-time analysis and training, while clinical metrics guarantee patient safety and superior surgical outcomes. This discussion primarily focuses on vision-task-based metrics, but it is beneficial to consider incorporating clinical metrics in future studies on this subject.

In conclusion, it is evident that surgical datasets and general action segmentation datasets differ significantly in terms of their structure, content, and the metrics used for performance evaluation. The limited number of videos and the uniformity in surgical datasets pose challenges that are inherently difficult to alter. However, by focusing on methods that prioritize event-based metric performance, we can better capture the temporal dynamics and dependencies within surgical procedures. This approach promises to provide a more comprehensive understanding and assessment of surgical phase segmentation, paving the way for more robust and accurate models in this specialized domain.

3 Sequence-to-sequence architectures

Through a case study on our in-house Sacrocolpopexy dataset, we identified the discrepancies in segmental performance across different surgeries and surgical datasets, particularly when the videos in a specific dataset are large. It was observed that although the standard Endo3D model reached an stat-of-the-art accuracy of 85.9, it experienced significant fragmentation issues based on the ward metric, with an event ratio of 0.266. To address this issue, we propose using sequence-to-sequence (seq2seq) models for coarse-level phase segmentation to manage the highly variable phase durations in Sacrocolpopexy. Various architectures (LSTM and transformer), configurations (time-shifted, time-synchronous), and training methods are evaluated within this proposed framework to assess its adaptability. It is important to note that this research was conducted just as transformers were about to gain popularity in the following months, coinciding with the first implementation work in surgical workflow segmentation, TransSV [31]. This study confirmed the practicality of using transformers in workflow segmentation, and showcased a possible configuration for implementing transformers in a seq2seq model.

We perform 7-fold cross-validation on the Sacro14 dataset. We perform both a frame-based (accuracy, F1-score) and an event-based (Ward metric) evaluation of our algorithms and show that different architectures present a trade-off between higher number of accurate frames (LSTM, Mode average) or more consistent ordering of phase transitions (Transformer). We compare the implementations on the widely used Cholec80 dataset and verify that relative performances are different to those in Sacro14.

3.1 Methods

Surgical workflow segmentation can be modeled as a sequential multi-label classification problem with inherent temporal constraints. Considering the most recent state-of-the-art deep learning approaches, these temporal constraints can be modeled at a fine-level with 3D convolutional neural networks (3D CNN's), and at a coarse level with a temporal model, such as LSTM. In this section, we assume that a fine-level model estimates a sequence of feature vectors (input sequence) and an initial workflow segmentation prediction from them (target sequence). We will now explore different ways of processing these sequences at a coarse level to produce an output sequence that represents our final workflow segmentation.

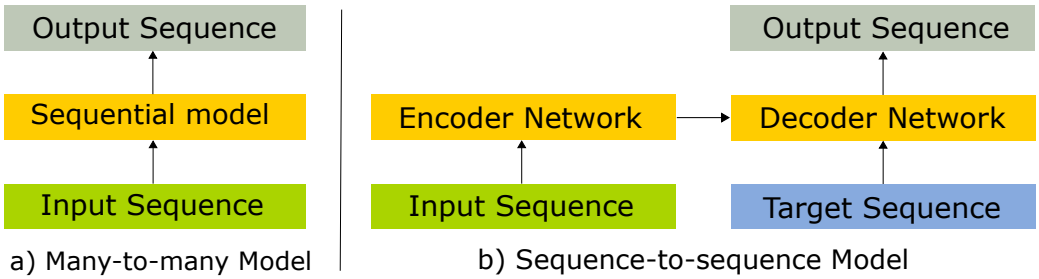


Figure 7: Network architectures for coarse-level sequential models. The main differences from the sequence-to-sequence to the many-to-many model are: 1) the presence of an encoder-decoder structure, allowing input/output sequences to have different sizes; 2) In addition to a sequence of feature vectors (input sequence), the input to this model also includes a sequence of label classifications (target sequence). The colour legend can be referred to Figure 8

We refer to conventional recurrent models in this domain as many-to-many (many2many) models, since both their input and output are sequences with

the same dimension [38, 64], where each unit in the sequential model processes a single input unit from the input sequence and produces a corresponding output, maintaining a one-to-one relationship. Although techniques like padding can be used to align the input and output sequences, these methods often result in inefficient implementations or require the use of a large model with superfluous parameters. In contrast, a seq2seq model can have input and output sequences of different sizes [28] that are linked by an encoder-decoder architecture. Additionally, a seq2seq model uses the fine-level predictions (target sequence) to guide feature selection at the the decoder level [67, 72]. These differences are summarised in Fig. 7. Recent works have also used related strategies for feature selection though attention mechanisms in the context of cholecystectomy workflow segmentation [18,31].

3.1.1 Network Architecture

Our proposed network (Fig.8) has two main components: a 3D convolutional neural network (Conv3D) for fine-level phase classification and a seq2seq model for coarse-level refinement. Conv3D takes clips x_t of 16 consecutive RGB images with 112×112 pixel resolution. Our 3D convolution architecture follows the Endo3D network architecture [13] and refers to the hyperparameters used in this work, which is based on Alexnet [52]. The Conv3D Network utilizes three-dimensional convolutional layers to process data with three spatial dimensions, typically height, width, and time. This architecture is particularly advantageous for handling video data and volumetric images, where temporal information is critical. By applying convolution operations over both spatial and temporal dimensions, Conv3D is able to capture not only the spatial features within each frame but also the temporal dynamics across consecutive frames. This allows the network to effectively model changes over time, making it suitable for tasks such as action recognition in videos, where understanding the sequence of movements is essential. A final fully connected layer is added to output 6 classifications (phases in

sacrocolpopexy) and 7 classifications (phases in Cholec80). Both the final classification as well as the 1200 dimensional feature vector from the previous fc8 layer is fed into the seq2seq model (Fig.9)

The seq2seq model analyses a larger video segment, consisting of 100 Conv3D clips. The base unit of seq2seq sequences are clips, not frames, and therefore at a coarse-level, we refer to the label of an entire clip as the most frequent label in its 16 frames. This technique generally has no impact on the clips during the middle of the phase. However, for clips at the transition, it establishes a distinct phase transition. During network training, the target sequence can be defined differently, e.g. as the groundtruth labels.

We implemented seq2seq with two base architectures, LSTM [41] and transformer [95].Both of these models can be adapted to the position of the seq2seq model as shown in Fig.8, without having to alter the input or output structure of the model. Note that LSTM has been already extensively used for surgical phase segmentation [43,73,106], but only as a conventional many2many sequential model. In this paper, we refer to LSTM adapted to the seq2seq structure. We additionally consider two configurations: time-synchronous and time-shifted.

Time-synchronous Configuration (100 series) The time-synchronous (Fig.8.a) configuration takes as the target sequence the 100 labels corresponding to the same clips as the fc8 feature vectors. Hence, these networks are named as LSTM100(L100) and Transformer100(T100) for simplicity.

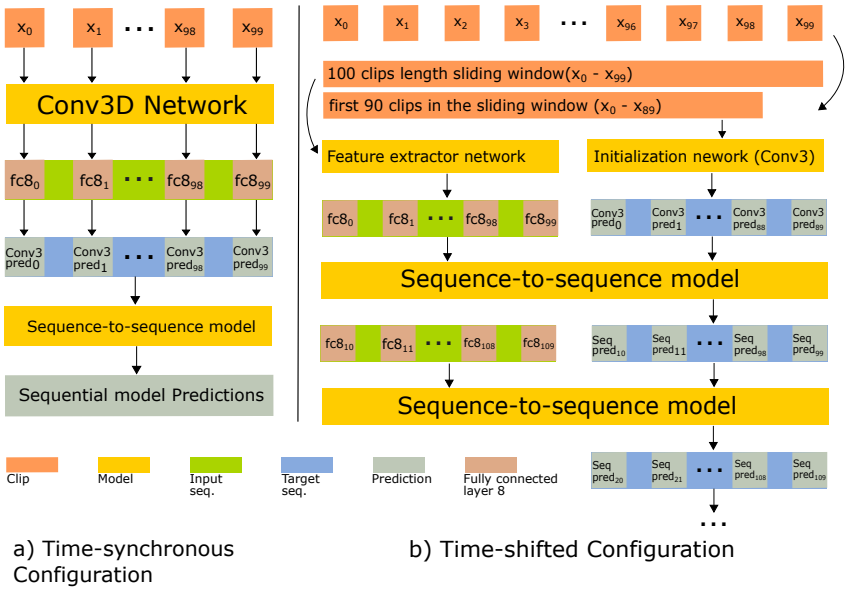


Figure 8: Seq2seq Network Architecture with a sequential input consists of 100 clips. The length of the target and output sequence depends on the configuration of the network: a) in the time-synchronous configuration the target, input and output sequences correspond to the same time interval of 100 clips; b) in the time-shifted configuration the target and output sequences have a length of 90 time steps with a shift of 10 between them. Together they span a length of 100 clips which corresponds to the size of the input sequence that is obtained from the Conv3D feature extractor. To obtain segmentations for consecutive sequences in a video, the seq2seq predictions become the target sequence of the next prediction iteration

Time-shifted Configuration (90 series) The time-shifted configuration (Fig.8.b) takes as the target sequence only the first 90 labels corresponding to the 100 fc8 feature vectors to predict the last 90 labels of that sequence. Hence, there are 10 labels in the prediction that act as 'future' labels relative to the target sequence with only 80 overlapping labels, caused the target sequence to shift by 10 timesteps relative to the input sequence. By having this shift between target sequence and prediction, the detection can have

the first 90 target sequence to be initialized by the Conv3D network and the prediction of all following labels in the video relies completely on the seq2seq model by treating the previous predictions as the current target sequence recursively. These type of networks are named as LSTM90(L90) and Transformer90(T90) for simplicity.

3.1.2 Network Parameters

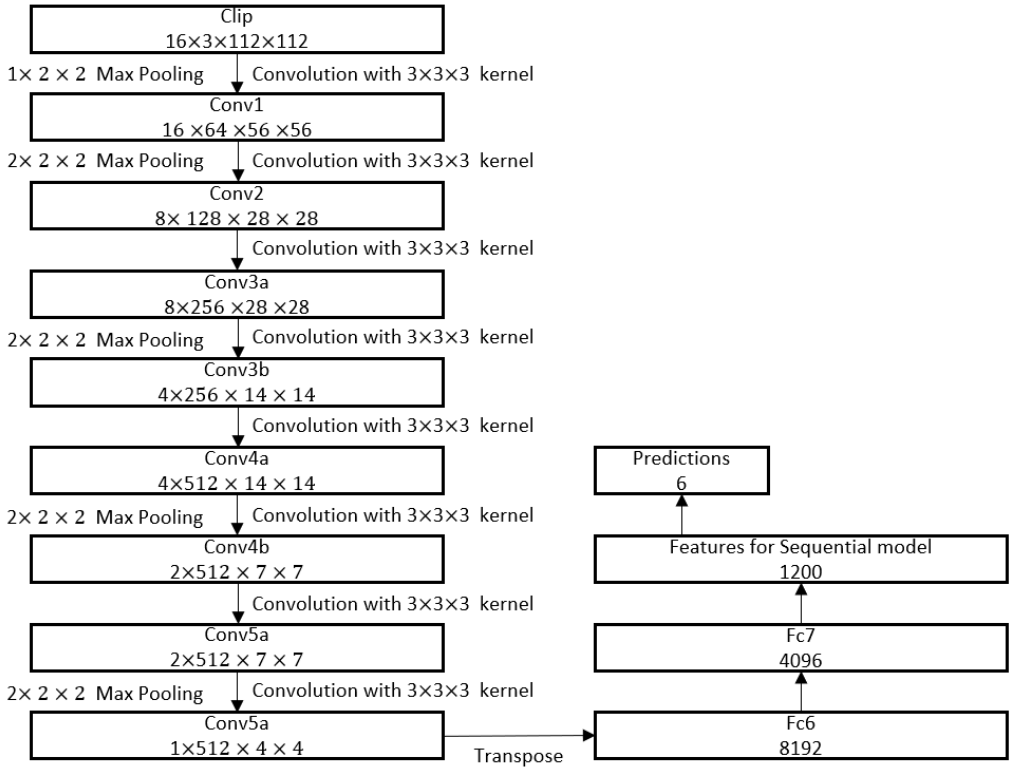


Figure 9: The C3D Network Architecture with each box represting a tensor with the labeled size

The detailed network parameters of Conv3D are presented in Fig.9. The fc8 layer that is extracted for sequential model input has a dimension of 1200. This dimension is used as the hidden dimensions of the LSTM model

for convenience and each separate LSTM model (many-to-many, LSTM encoder and LSTM decoder) has 3 hidden layers.

The transformer setup is analogue to the originally proposed default settings [95] with 6 layers of the 8 head encoder-decoder pairs. The input dimensions are adjusted to fit our input sequence with a sequence length of 100 and the d_{model} of 1200. The inner layer dimension for the feed forward network is reduced to 1000 to decrease the model size. And the sine and cosine functions of different frequencies are used for positional encoding as used in the original transformer setup [95].

3.1.3 Network Training Strategies

Conv3D and Seq2seq are trained separately. The Conv3D model was fine-tuned based on the parameters that have been pre-trained on the Cholec80 dataset, as Cholec80 and our dataset have roughly similar tool usages and the tissue shares some spatial features. For seq2seq we defined different training strategies in terms of sampling policy and usage of the target sequence. All strategies are independently modified from a baseline so that an ablation study can evaluate them independently. These training strategies are defined as following:

- **Standard method (baseline)**: During training, the input to the target sequence are the groundtruth labels. When deployed, the network uses Conv3D (time-synchronous) or past Seq2Seq (time-shifted) predictions instead. This is the standard approach for training seq2seq models in previous works [91]. The entire video is sampled for training in sequence. For balancing the videos of different lengths in the training data, a fixed number of sliding windows (200) is sampled from each video, with their interval changing depending on total video time. During training, the sliding windows with the same indices will be extracted from each video and assembled into a batch. Hence, each

batch contains the samples that are at the same relative positions in all videos.

- **Target Sequence with injected noise (noised)**: We inject noise into the groundtruth target sequence to simulate prior classification errors during training and enabling seq2seq to learn a filtering action. Noise is injected by randomly replacing 40% with correct labels which is the average accuracy of applying a pure C3D network to the dataset.

- **Target Sequence with Predicted Labels (pred)**: Similarly to the previous strategy, we introduce classification errors by using Conv3D predicted labels as the target sequence. This method may preserve some internal structures between the predicted labels.

3.1.4 Loss Function

Cross entropy loss is utilized in training the network. The general form of the loss function for the Conv3D network is:

$$L_{Conv}(\boldsymbol{y}, \boldsymbol{x}) = -\frac{1}{d}\sum_{j=1}^{d}\sum_{i=1}^{n} w_i \boldsymbol{y}_{i,j} log(\boldsymbol{x}_{i,j}), \tag{8}$$

where $\boldsymbol{x}$ is the softmax output from the network and $\boldsymbol{y}$ is the one-hot label for that particular clip. There are n classes of labels that represent the phases and each label has a corresponding weight w_i in evaluation. Multiple samples are trained together with a batch size d and the average loss for all samples are considered as the general loss for that batch.

The loss function for the sequential model is similar but with an extra time dimension t for the sequence length:

$$L_{sequential}(\boldsymbol{y}, \boldsymbol{x}) = -\frac{1}{td}\sum_{k=1}^{t}\sum_{j=1}^{d}\sum_{i=1}^{n} w_i \boldsymbol{y}_{i,j,k} log(\boldsymbol{x}_{i,j,k}). \tag{9}$$

3.2 Experiment Setup

3.2.1 Post-processing

Both the time-synchronous configuration and the time-shifted configurations have fixed-length input and output sequences. The length is designed to be short enough for extracting sufficient amount of sliding windows from the videos. Hence, it is necessary for composing the output sequence together for a final predicted sequence. For the time-shifted configuration, there are overlaps existing between the sliding windows. A single time step in the video can have multiple predictions throughout the sliding windows. The mode of the predicted labels among the multiple predictions is taken as the final prediction for that time step. For the time-synchronous configuration, the sliding windows can be assembled in sequence as there are no overlaps between them.

3.2.2 Comparison with the state-of-the-art

With the sacrocolpopexy dataset, we compare our seq2seq results against raw predictions from [13] (C3D), a filtered version with mode averaging, and the many-to-many models LSTM and TCN. C3D+LSTM can take sequences of arbitrary length, and thus it is normal to perform predictions based on all past frames. However, our seq2seq models require a fixed sized sequence and perform predictions using a sliding window. To understand how this affects the performance, we test C3D+LSTM with both all-past-frames input mode and with a sliding window input mode. All above methods are also tested on the Cholec80 public dataset [94] to which we add for completeness the state-of-the-art results as reported in [13], [44], [17]. The major difference between our dataset and Cholec80 is the overall duration of each phase, which can be significantly larger in Sacrocolpopexy. Notably, this significantly changes the relative performance between different algorithms, as we show in Sec. 3.3.2.

3.2.3 Training Details

The captured videos are downsampled to 2.4 fps, centre cropped, and resized into a square of resolution 300×300 pixels before them are augmented into 112×112 pixels to match the input requirement of the Conv3D network to prevent extra information loss. Then, 16 consecutive frames are assembled into a clip as the basic unit of input for the Conv3D network. The most common label (mode value) for all the frames in a clip is assigned as the label for that clip. The sequential model takes a continuous sequence of 100 clips (1600 frames) as input, where the clips are processed by the Conv3D network first and its last fully connected layer of 1200 neurons for those 100 clips are assembled into a tensor as one training sample.

Data augmentation is applied to each clip along with sampling [10] by performing horizontal and/or vertical flip, rotation in the range of 0 to 360 degrees, crop with a minimum factor of $\frac{1}{9}$ of the original image and then resizing, blur with a Gaussian filter of 5×5 kernel with 1.5 standard deviation and luminance variation in the range of 0.6 to 1.4. These augmentations are selected randomly with a uniform distribution within the indicated ranges. The same augmentation is applied to all the frames in a single clip for consistency. Finally, all frames are resized to 112×112 pixels to match the input requirement of the Conv3D network. The proposed network is implemented in PyTorch using a single Tesla V100-DGXS-32GB GPU of an NVIDIA DGX station.

The training is performed using 7-fold cross-validation for Sacro14. The 14 videos that constitute our dataset are divided into seven pairs where five of them are used for training, 1 pair is used for validation and 1 pair is used for testing. Cholec80 has sufficient amount of videos, we use 40 videos for training, 20 for validation and 20 for test. Adam [51] optimiser with a learning rate (l_r) of $1e^{-5}$ and a decay set to $0.93 \times l_r$ for every fifth epoch is used

for the Conv3D network training.

Each epoch contains 600 samples of batch size 10 with each phase sampled to a same number. The average accuracy without the transition phase and non-phase is calculated on the validation set 4 times per epoch, and network parameters with the best accuracy in history are saved as the final parameters. The output (fc8 and prediction) of the trained Conv3D are then used as input for sequential models.

3.3 Results and discussion

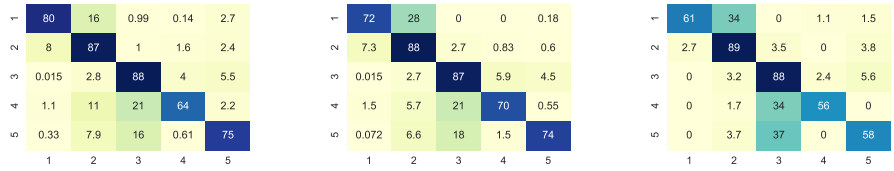
3.3.1 Ablation Study of Seq2Seq On Sacrocolpopexy

Architectures			Precision (Macro)	Recall (Macro)	F1-Score	Accuracy (Micro)
LSTM(L)	100	baseline	61.6±6.7	74.8±9.7	0.68	70.7±9.0
		pred	72.8±12.8	69.6±17.6	0.71	80.4±13.0
		noised	74.6±11.8	**78.8±11.5**	**0.77**	**82.8±9.8**
	90	baseline	53.7±24.1	54.4±17.5	0.54	67.2±22.3
		pred	57.7±16.0	59.0±15.1	0.58	75.5±20.2
		noised	53.5±16.3	58.8±11.7	0.56	76.5±16.0
Transformer(T)	100	baseline	64.6±13.7	63.2±14.7	0.64	73.1±13.4
		pred	**75.4±14.3**	69.4±14.2	0.72	80.6±16.1
		noised	72.9±14.2	68.6±15.7	0.71	82.7±13.5
	90	baseline	**76.4±12.6**	**71.7±15.5**	**0.74**	81.1±15.5
		pred	71.7±14.2	65.1±13.1	0.68	80.4±14.1
		noised	74.9±13.6	71.2±15.5	0.73	**81.9±14.1**

Table 2: Ablative phase recognition results(%) over different proposed architectures on Sacrocolpolpexy dataset the best among each configuration are bolded in different colour (green for 100 series and blue for 90 series)

Table 5 shows an ablation study of our different seq2seq implementations, and Fig.11 shows its results on a particular video sequence. The noised training strategy overall performed best for both time-synchronous (100 series) and time-shifted (90 series) configurations, with respectively LSTM100

(L100) and Transformer90 (T90) being the best performing in terms of accuracy. The baseline strategy using groundtruth labels for the target sequence is generally the worst, with a single exception (T90). In this case the network suffers from the exposure bias [83] as there is a strong dependency between the groundtruth and the predictions.



(a) Mode Averaged C3D (b) L100 noised (c) T90 noised

Figure 10: Sacrocolpopexy per phase results: averaged confusion matrices(%) over all cross-validation folds normalized by the sample number of each phase with the two best methods in sequential models. (Note: transition phase is eliminated from the graph)

Figure 10 (b) and (c) provides the confusion matrices of the selected methods. Most of the misclassifications for this type of surgery happens between the two consecutive phases as phase 1-2, phase 3-4 and phase 3-5. The mesh is introduced in phase 3 which separates the following phases from the first 2 phases. The same tools are also used in phase 3, 4 and 5 but applied to different positions with phase 3 (promontory) and 4 (vault). Phase 5 can be started from either phase 3 or 4 but in most cases is phase 3, hence it has more misclassifications with phase 3 rather than phase 4.

3.3.2 Comparison With the State-of-the-art

	Sacrocolpopexy (average of 1389 clips per video)				Cholec80 (average of 360 clips per video)		
Method	Pre. (Macro)	Rec. (Macro)	F1-Score	Acc. (Micro)	Pre. (Macro)	Rec. (Macro)	Acc. (Micro)
C3D+LSTM+Tool(Endo3D)* [13]					81.3	**87.7**	91.2
ResNet-50+LSTM+PKI (SV-RCNet)* [44]					**90.6±8.1**	86.2±15.3	**92.4±5.2**
ResNet-50+LSTM * [44]					80.7±7.0	83.5±7.5	85.3±7.3
ResNet-50+TCN(TeCNO Stage I)* [17]					**82.44±0.46**	**84.71±0.71**	**88.35±0.3**
C3D	58.5±6.8	68.6±10.1	0.63	69.2±8.8	67.5±8.1	74.7±7.4	71.0±8.5
C3D + Mode average	**78.1±9.5**	**79.7±12.6**	**0.79**	**82.8±9.5**	73.9±10.6	81.2±9.9	79.5±8.1
C3D+TCN	76.6± 12.6	74.3± 15.3	0.72	82.6±12.4	**81.3±5.9**	**82.0±8.4**	83.8±7.8
C3D+LSTM	71.6±22.6	64.8±19	0.68	77.1±18.8	80.1±10.0	**82.0±8.3**	**85.9±7.9**
C3D+LSTM+Sliding Window	71.2±17.5	65.8±15.7	0.68	79.2±14.7			
C3D+T90 noised(Proposed)	74.9±13.6	71.2±15.5	0.73	81.9±14.1	43.7±18.7	48.1±16.0	71.1±13.9
C3D+L100 noised(Proposed)	74.6±11.8	78.8±11.5	0.77	**82.8±9.8**	64.9±9.6	73.5±10.6	81.1±5.3

Table 3: Comparison of the phase recognition results(%) with other methods on the Sacrocolpopexy and Cholec80 datasets. Asterisk (*) denotes cholec80 results were directly extracted from respective publications, while the others are our own implementations. This table is grouped by (row 1-2) methods that use models specific to cholecystectomy (tools or priors), as reported in previous literature; (row 3-4) models with ResNet-50 backbone, as reported in previous literature; (row 5-11) models with a C3D backbone, as proposed in this work. Note: In this table, the green color highlights the optimal performance for Sacrocolpopexy, while the blue color indicates the top performances among Cholec80.

Our best performing seq2seq time-synchronous and time-shifted models (T90, L100 noised) are also compared with previously proposed approaches on our Sacrocolpopexy dataset (Table 3). First, we can observe that performing predictions on a sliding window does not affect the general performance of Endo3D, slightly increasing its accuracy. This suggests that the loss of input information from using a fixed sliding window is not negatively affecting performance and therefore this should not be a limiting factor in our seq2seq architectures that always operate on a sliding window. Both seq2seq models (T100,L90) outperform the many-to-many approach (Endo3D). Surprisingly, the best performance in terms of F1-score is the simple mode average on C3D results which narrowly beats the seq2seq L100 noised. However, an analysis purely based on F1-scores disregards how accurately are we capturing a time ordered sequence of events. To further interpret these results we also

perform an event-based evaluation. (Sec.3.3.3)

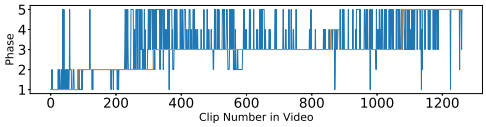
Table 3 also compares the performance of networks on the Cholec80 dataset. Our own implementation of Endo3D (no-tool) achieves a close result to the original Endo3D + LSTM, where the slight decrease in performance is explained by not using tool signal information. The average number of clips per video in Cholec80 is 360 which is much smaller than in Sacrocolpopexy (1389). Furthermore, the relative proportions of each phase is also generally different. Taking these factors into consideration, it is worth noting that the relative performances between our implemented methods is almost reverted in Cholec80, with Endo3D performing the best and mode average the second worst. This shows that the specific characteristics of a given surgery greatly affects algorithm performance. More specifically, we verify that our seq2seq models outperform conventional LSTM on Sacrocoplopexy but this is not the case on Cholec80. We should also highlight that more recent approaches such as TeCNO (based on Temporal Convolutional Network) and SV-RCNet+PKI (uses surgery-specific priors) still outperform both conventional LSTM and seq2seq models on Cholec80 according to their reported results. Even though we have clearly shown that we should not draw firm conclusions on how they would perform in Sacrocolpopexy data, they are still worth considering as promising options.

3.3.3 Event-based Analysis

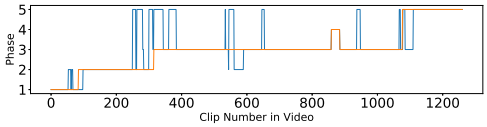
Method	F	C	F'	event ratio
C3D	79	4	2299	0.015
Mode Average	49	33	218	0.172
Endo3D(no-tool)	30	41	123	0.266
Endo3D+slinding window	39	35	150	0.238
L100 noised	63	19	415	0.097
T90 noised	28	42	98	0.313
LSTM avg.	40	24	347	0.217
Trans avg.	36	34	188	0.215
100 series avg.	54	22	427	0.123
90 series avg.	22	36	107	0.309

Table 4: Ward Metric results summed over all Sacrocolpopexy cross-validation folds. F and F$'$ represents the fragmentation label where an event F in the groundtruth is fragmented into multiple F$'$ events in the predictions. C represents the correct labels for the events in predictions that are matched with the corresponding events in ground truth.

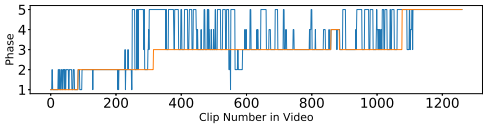
Table 4 shows the sum of the Ward metric results over the 7 cross validation folds. The event ratio, number of correct (C) events and number of the fragmentation errors (F, F$'$) are presented in the table. A higher event ratio means that the temporal order of phase transitions is better preserved. Filtering very noisy predictions generally leads to better results in this evaluation due to eliminating a significant number of false phase transitions (e.g. comparing C3D with its mode average). Seq2seq models can further increase the event ratio in most cases. T90 noised has a slightly worse F1-score and accuracy than the mode average, but it has a significantly better Ward metric, specifically in terms of its event ratio and low fragmentation number. This effect can be visualised in the example results in Fig 11, where even if overall accurate, mode average tends to have many incorrect transitions, while T90 performs all transitions in correct order but accumulates errors near the phase transitions. This may be a desirable outcome, since phase



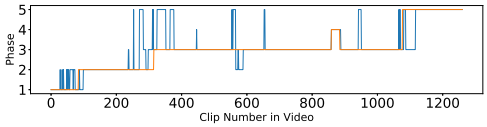
(a) Pure Conv3D output sequence



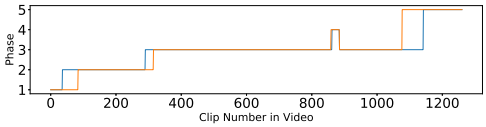
(b) C3D with mode average output sequence



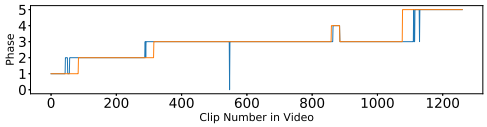
(c) L100 baseline output sequence



(d) L100 output sequence with injection noise target sequence



(e) T90 baseline output sequence



(f) T100 output sequence with injection noise target sequence

Figure 11: Phase diagrams from the best Sacrocolpopexy fold. Orange is ground-truth label and blue is predicted label.

transitions are by definition more subject to annotation ambiguity than the middle of the phases. The trade-off between F1-score and event ratio can also be observed by comparing the overall performance of time-synchronous configurations (100 series) with the time-shifted configurations (90 series). The first one tends perform better in terms of F1-score, while second better preserves number and order of transitions.

3.4 Conclusion

In this chapter, we introduce frame-based seq2seq models as a novel coarse-level sequential model for surgical workflow segmentation. We validated the approach on a challenging dataset of Sacrocolpopexy surgery where phase duration has a very high variability. We experimentally highlight the differences between this dataset and the widely studied benchmark Cholec80,

showing that the same set of algorithms have different relative performances on each dataset. Additionally, the inclusion of an event-based analysis (Ward metric) to complement more standard accuracy metrics (F1-score,accuracy) revealed a trade-off between different seq2seq configurations. While L100 (and more generally, seq2seq 100 series) performs accurate predictions on a higher number of frames, T90 (and more generally, the 90 series) produces a temporally more consistent workflow prediction. How each criteria should be weighted will invariably be application-specific. Nevertheless, accurate time-stamping of phase transitions requires both standard and event metrics to perform well. Nevertheless, despite the high accuracy achieved by seq2seq, it is not entirely possible to eliminate random transitions between phases during a single phase, which leads to a lower event ratio.The intrinsic weaknesses of this architecture complicate the determination of precise transition points between phases and hinder the accurate identification of the phase order as well. On the other hand, this error can be alleviated If phases can be detected in one shot rather than in independent frames. The next chapter introduces a method to do so.

4 Transition Retrieval Network

The preceding chapter offers insights into analyzing surgical workflow segmentation by incorporating an event-based metrics (Ward metric) alongside frame-based analysis. Additionally, a novel frame-based method is proposed to improve the segmental-level behavior of the predictions.

Nevertheless, the frame-based approach inherently encounters issues with erroneous transitions, as the predictions unavoidably produce random noise. To adress this issue, we introduce a novel reinforcement learning formulation for offline phase transition retrieval. Instead of attempting to classify every video frame, we identify the timestamp of each phase transition. By construction, our model does not produce spurious and noisy phase transitions, but contiguous phase blocks. We investigate two potential configurations that this new model setup can accomplish: one is focused on reducing computational cost, and the other is aimed at achieving optimal performance The first does not require processing all frames in a video (only $< 60\%$ and $< 20\%$ of frames in Cholec80 and Sacro38 respectively), while producing results slightly under the state-of-the-art accuracy. The second configuration processes all video frames and outperforms the state-of-the art at a comparable computational cost.

We compare our method with recent top performing frame-based approaches TeCNO and Trans-SVNet on the public dataset Cholec80 and also the inhouse Sacro38 dataset. Besides the earlier metrics, we altered the Ward metric to a simplified version called Ward event ratios as explained in Chapter 2 to provide a clearer understanding of the segmental performance of the methods.

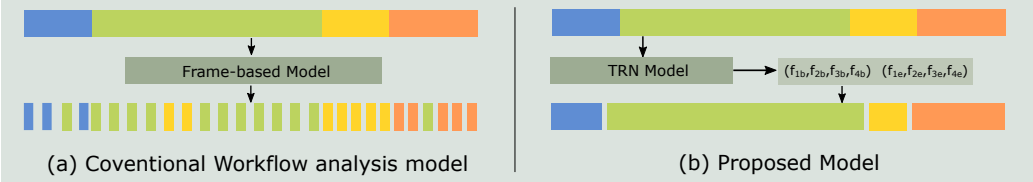


Figure 12: Comparison of network architecture between (a) conventional model and (b) our proposed model with potential error illustration. The conventional model assigns labels for each individual frames and our proposed model predicts frame indices for the starts and end position of phases.

4.1 Methods

In this chapter, we introduced an innovative method for segmenting surgical workflows by pinpointing the transition points between different phases directly. The main feature of our proposed formulation can be visualised in Fig.12. While previous work attempts to classify every frame of a video according to a surgical phase label, we attempt to predict the frame index of phase transitions. More specifically, for a surgical procedure with N different phases, our goal is to predict the frame indices where each phase starts $\{t_{1b}, t_{2b}...t_{Nb}\}$, and where each phase ends $\{t_{1e}, t_{2e}...t_{Ne}\}$. Assuming surgical phases occur as continuous events without abrupt shifts to different phases and then returning, which is frequently observed, our method naturally supports this assumption. In contrast, conventional frame-based methods inadvertently identify transitions at any point within a phase, resulting in incorrect insertion between phases or the fragmentation of a single phase. To solve this problem we propose the Transition Retrieval Network (TRN), which we described next.

4.1.1 Architecture of Transition Retrieval Network (TRN)

Figure 13 shows the architecture of our TRN model. It has three main modules: an averaged ResNet feature extractor, a multi-agent network for

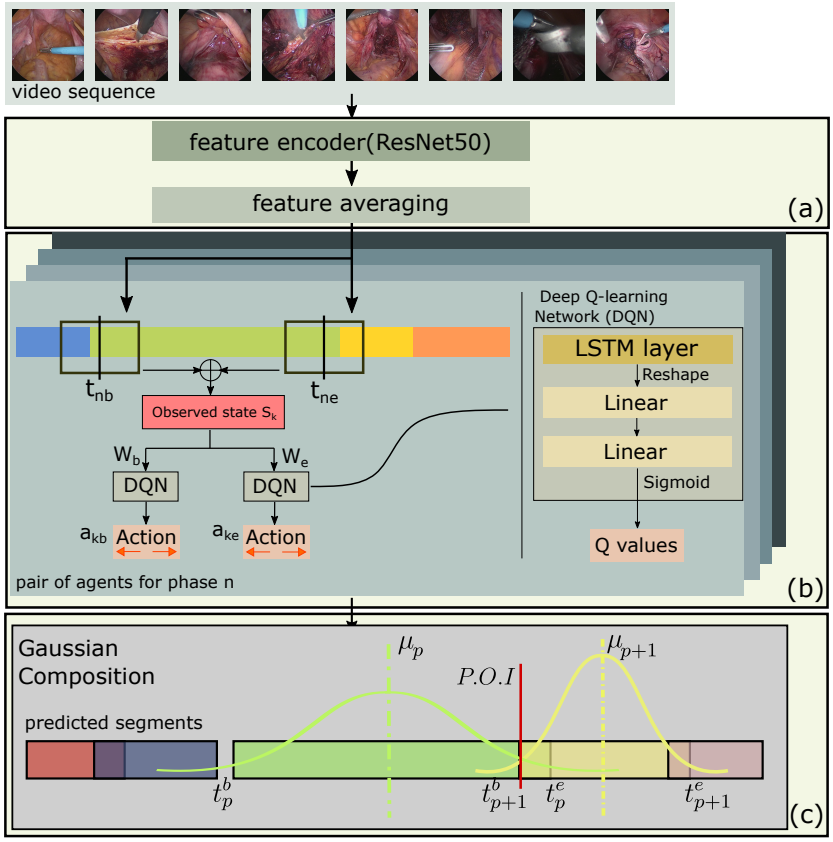


Figure 13: TRN architecture with (a) averaged ResNet feature extractor, (b) multi-agent network for transition retrieval and (c) Gaussian composition operator

transition retrieval, and a Gaussian composition operator to generate the final workflow segmentation result

Averaged ResNet feature extractor: We first train a standard ResNet50 encoder (outputs 2048 dimension vector) with supervised labels, in the same way as frame-based models. For a video clip of length M, features are averaged into a single vector. We use this to temporally down-sample the video through feature extraction. In this work we consider $M = 16$.

DQN Transition Retrieval: At this stage, we process phases independently denoted by n. We treat it as a reinforcement learning problem with 2 discrete agents W_b and W_e, each being a Deep Q-Learning Network (DQN) [4, 49]. These agents iteratively move a pair of search windows centered at frames t_{nb} and t_{ne}, with length L. The state of the agents s_k is represented by the $2L$ features within the search window, obtained with the averaged ResNet extractor. As the agents progress through their steps, the index k denotes the count of steps taken. Based on their state, the agents generate actions $a_{kb} = W_b(s_k)$, and $a_{ke} = W_e(s_k)$, which move the search windows either one clip to the left or to the right within the entire video. During network training, we set a +1 reward for actions that move the search window center towards the groundtruth transition, and -1 otherwise. During training, we discovered that providing an additional reward for 'halting precisely at the transition point' led the agent to stop indiscriminately, making convergence difficult. Therefore, we streamlined the reward function by removing the stop reward. Instead, the agent maneuvers freely until reaching a predetermined step that guarantees oscillation around the target transition. Therefore, we learn direction cues from image features inside the search windows. As our input to DQN is a sequence of feature vectors, a 3-layer LSTM of dimension 2048 is introduced to DQN architecture for encoding the temporal features into action decision process. The LSTMs are followed by 2 fully connected layer of dimension $20L$ and 50 respectively that maps temporal features to the final 2 Q-values of 'Right' and 'Left'. We implemented the standard DQN training framework for our netwrok. [71] At inference time, we let the agents explore the video until they converge to a fixed position (i. e. cycling between left and right actions). Two important characteristics of this solution should be highlighted: 1) we do not need to extract clip features from the entire video, just enough for the agent to reach the desired transition; 2) the agents need to be initialised at a certain position in the video, which we discuss later.

Agent initialization configurations: We propose two different approaches to initialise the agents: fixed initialization (FI) and, ResNet modified initialization (RMI). FI initializes the search windows based on the statistical relative position (frame index average) of each phase transition on the entire training data. With FI, TRN can make predictions without viewing the entire video and save computation time. On the other hand, RMI initialises the search windows based on the averaged-feature ResNet-50 predictions by averaging the indices of all possible transitions to generate an estimation. In this way, we are very likely to have more accurate initialization positions to FI configuration and yield better performance.

4.1.2 Merging different phases with Gaussian composition:

So far, we have only explained how our DQN transition retrieval model segments a single phase. To generalise this, we start by running an independently trained DQN transition retrieval model for each phase. If we take the raw estimations of these phase transitions, we inevitably create overlapping phases, or time intervals with no phase allocated, due to errors in estimation. we used the Gaussian composition (shown in Figure 13 (C)). We obtain the middle point μ_i of each phase from its beginning and end transition predictions. For each phase, we define a Gaussian curve with mean μ_i and standard deviation σ_i equal to half of the phase duration divided by a slackening factor C. These values can be obtained by the following expressions:

$$\mu_j = \frac{t_j^e - t_j^b}{2} + t_j^b, \quad \sigma_j = \frac{t_j^e - t_j^b}{2C} \tag{10}$$

where t_j^b, t_j^e denote the beginning and end transitions of phase j respectively.

We now define our final estimation of phase transitions as the point of intersection (P.O.I.) between adjacent Gaussian curves $X_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$ and

$X_{j+1} \sim \mathcal{N}(\mu_{j+1}, \sigma^2_{j+1})$ as follows:

$$t^e_j = t^b_{j+1} = P.O.I(X_j, X_{j+1}) \tag{11}$$

This results in a workflow segmentation defined by timestamps in continuous time domain. For comparison with frame-based methods and computing performance metrics, we discretise our workflow segmentation by assigning each frame label $\mathbf{n}$ to its corresponding transition:

$$\mathbf{Y}_{t^b_j:t^e_j} = j \tag{12}$$

4.2 Training details

The DQN model is trained in a multi-agent mode where W_b, W_e for a single phase are trained together. In this scenario, the agents engage with the same environment but do not pursue a shared reward; instead, they optimize their individual rewards. This situation is known as independent multi-agent reinforcement learning [22, 61]. The input for individual DQNs in each agent shares a public state concatenated from the content of both search windows, allowing the agents to be able to aware information of others. The procedures of training the DQN are showing in pseudo code in Algorithm 1. For one episode, videos are trained one by one and the maximum number of steps an agent can explore in a video is 200 without early stopping. For every steps the agents made, movement information (s_k, s_{k+1}, a_k, r_k) are stored in its replay memories, and sampled with a batch size of 128 in computing Huber loss [71]. This loss is optimized with gradient descent algorithm, where α is the learning rate and $\nabla_{W_k}\mathcal{L}_k$ is the gradient of loss in the direction of the network parameters. The detailed equations for updating parameters are explained below.

Algorithm 1 The procedures of training DQN

Initialize parameters of agents W_b and W_e as W_{0b} and W_{0e}
Initialize individual replay memories for agents W_b and W_e
for $episode \leftarrow 0$ to $episode_{MAX}$ **do**
  Initialize search window positions (FI or RMI)
  for $video \leftarrow 0$ to $range(videos)$ **do**
    for $k \leftarrow 0$ to 200 **do**
      $s_k \leftarrow$ read ResNet features in search window
      $a_{kb} \leftarrow W_{kb}(s_k)$ and $a_{ke} \leftarrow W_{ke}(s_k)$
      $s_{k+1} \leftarrow$ update search window position by (a_{kb}, a_{ke}) , read new features
      $r_{kb}, r_{ke} \leftarrow$ compare s_k and s_{k+1} with reward function
      Save $(s_k, s_{k+1}, a_k, r_{kb})$ and $(s_k, s_{k+1}, a_k, r_{ke})$ into agent memory
      Compute loss $(\mathcal{L}_{kb}, \mathcal{L}_{ke})$ from random 128 samples from each memory
      Optimize W_{kb}: $W_{k+1b} \leftarrow W_{kb} + \alpha \nabla_{W_{kb}} \mathcal{L}_{kb}$
      Optimize W_{ke}: $W_{k+1e} \leftarrow W_{ke} + \alpha \nabla_{W_{ke}} \mathcal{L}_{ke}$
    end for
  end for
end for

DQN is a Q-function approximator that maps input features s_k and action a_k into Q-value where an ideal Q-function maps the highest Q-value with the best action to take for a known state. A policy $\pi()$ is the process of choosing action to maximize the reward with the best Q-value. An ideal Q-function satisfies the Bellman equation:

$$Q(s_k, a_k) = r + \gamma Q\left(s_{k+1}, \pi\left(s_{k+1}\right)\right) \tag{13}$$

where k represents the current state, $k+1$ is the next state after taking an action chosen from policy $\pi()$ and r is the reward for taking that action.

In our experiment, the reward is defined by the movement of the central position of W. The reward is set to 1 if the agent is moving closer to its target transition and to -1 if the agent in moving away from it in this step. In

real situations, the fact that Q-function is not perfect leading to a difference δ between the two sides of Bellman equation:

$$\delta = Q_P(s_k, a_k) - \left(r + \gamma \max_{a_{k+1}} Q_T\left(s_{k+1}, a_{k+1}\right) \right) \tag{14}$$

Our training purpose is to minimize this difference δ. We applied the Huber loss to it on a batch B sampled from a memory of the past taken steps.

$$\begin{aligned} \mathcal{L} &= \tfrac{1}{|B|} \textstyle\sum_{(s_k, a_k, s_{k+1}, r_k) \in B} \mathcal{L}(\delta) \\ \text{where} \quad \mathcal{L}(\delta) &= \begin{cases} \frac{1}{2}\delta^2 & \text{for } |\delta| \leq 1 \\ |\delta| - \frac{1}{2} & \text{otherwise.} \end{cases} \end{aligned} \tag{15}$$

Noticeably, the difference δ is calculated with two separate networks of same architecture. The $Q_P()$ part called policy net that the parameters are updated for every step and $Q_T()$ is called target net [71]. As the samples in batch are discrete in time having less correspondence with each other, the optimization of the policy net may forget the learnt features catastrophically. By coping the parameters from policy net to target net periodically improves the robustness of the optimization process [71].

4.3 Experiment setup and Dataset Description

The proposed network is implemented in PyTorch using a single Tesla V100-DGXS-32GB GPU of an NVIDIA DGX station. For the ResNet-50 part, PyTorch default ImageNet pretrained parameters are loaded for transfer learning. The videos are subsampled to 2.4 fps, centre cropped, and resized into resolution 224*224 to match the input requirement of ResNet-50. We train both ResNet-50 and DQN with Adam [51] at a learning rate of 3e-4. For ResNet-50, we use a batch size of 100, where phases are sampled with equal probability. For DQN, the batch size is 128.

We evaluated the performance of the TRN model on both the Cholec80 and Sacro38 datasets. At the time of this study, the laparoscopic sacrocolpopexy dataset has been increased to 38 videos. The Sacro38 contains up to 8 phases (but only 5 in most cases) at this stage, however, here we consider the simplified binary segmentation of the phases related to suturing a mesh implant (2 contiguous phases), given that suturing time is one of the most important indicators of the learning curve [16] in this procedure. As indicated in the literature, the timing of these two phases is of clinical interest for assessing a surgeon's performance in performing this surgery. We performed a 2-fold cross-validation with 20 videos for training, 8 for validation, and 10 for testing. For Sacrocolpopexy, we train our averaged ResNet extractor considering all phases, but train a single DQN for retrieving the suturing phase as the second stage of the whole network. We also do not require to apply Gaussian composition since we're interested in a single phase classification.

4.3.1 Evaluation metrics:

Apart from the evaluation metrics introduced in Sec.2.3, we also provide a coverage rate for the fixed initialisation (FI) configuration, indicating the average proportion of the duration for each video that was processed to perform the segmentation. Lower values indicate fewer features need to be extracted and thus lower computation time.

4.4 Results and Discussion

4.4.1 Ablative Study of TRN on Cholec80

Window size	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7	Overall F1-score
TRN21 FI	0.854	0.917	0.513	0.903	0.687	0.549	0.83	0.782
TRN41 FI	0.828	0.943	0.636	0.922	0.558	0.694	0.85	0.808
TRN21 RMI	0.852	0.942	0.619	0.939	0.727	0.747	0.837	0.830
TRN41 RMI	0.828	0.940	0.678	0.945	0.753	0.738	0.861	0.846

Table 5: TRN ablation in the Cholec80 dataset (F1-scores). The values per-phase are computed before Gaussian Composition, while the overall F1-score is for the complete TRN method.

We first provide an ablation of different configurations of our TRN model in Table 5, for Cholec80. It includes two search window sizes (21 and 41 clips) and two initialisations (FI, RMI). The observations are straightforward. Larger windows induce generally better f1-scores, and RMI outperforms FI. This means that heavier configurations, requiring more computations, lead to better accuracies. Particular choice of a TRN configuration would depend on a trade-off analysis between computational efficiency and frame-level accuracy.

4.4.2 Comparison With Other Works

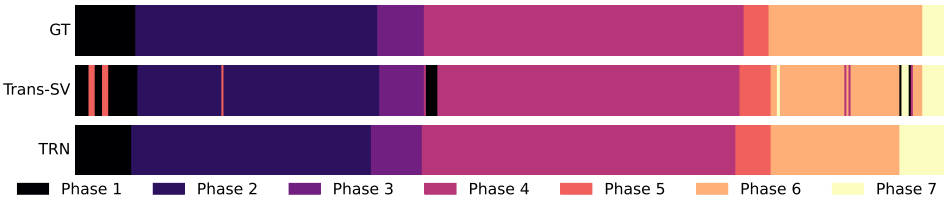
Dataset	Method	Accuracy	Precision	Recall	F1-Score	Event ratio	Ward Event Ratio	Coverage rate(%)
Cholec80	ResNet-50	79.7±7.5	73.5±8.4	78.5±8.9	0.756	0.120	0.375	full
	TeCNO	88.3±6.5	78.6±9.9	76.7±12.5	0.774	0.381	0.691	full
	Trans-SVNet	89.1±5.7	81.7±6.5	79.1±12.6	0.800	0.316	0.566	full
	TRN21 FI	85.3±9.6	78.1±11.1	78.9±13.5	0.782	1	0.934	57.6
	TRN41 FI	87.8±8.1	80.3±9.1	81.7±12.4	0.808	1	0.956	59.1
	TRN41 RMI	90.1±5.7	84.5±5.9	85.1±8.2	0.846	1	0.985	full
Sacrocol--popexy	ResNet-50	92.5±3.8	94.9±2.8	84.5±8.4	0.892	0.029	0.016	full
	TeCNO	98.1±1.7	97.7±1.9	97.5±3.0	0.976	0.136	0.438	full
	Trans-SVNet	97.8±2.2	96.5±4.5	98.0±3.5	0.971	0.536	0.813	full
	TRN21 FI	89.8±6.2	88.6±11.7	85.3±11.1	0.860	0.971	0.875	14.6
	TRN81 FI	90.7±6.1	88.6±11.5	88.5±11.1	0.875	0.941	0.860	18.3

Table 6: Evaluation metric results summary of ResNet-50, our implementation of TeCNO and Trans-SV, and ablative selected TRN result on Cholec80 and Sacrocolpopexy.

Table 6 shows a comparison between TRN and state-of-the-art frame-based methods on both Cholec80 and Sacrocolpopexy. The utilised baselines are TeCNO [17], Trans-SVNet [31], which we implemented and trained ourselves. Instead of simple ResNet50, we use the same feature averaging process as the TRN for consistency.

For Cholec80, our full-coverage model (TRN41 RMI) surpasses the best baseline (Trans-SVNet) in all frame-based metrics, while having significantly better even-based metrics (event ratio, Ward event ratio). This can be explained by TRN's immunity to frame-level noisy predictions, which can be visualised on a sample test video in Fig. 14(a).

Still for Cholec80, our partial-coverage models (TRN21/41 FI) have frame-based metrics below the state-of-the-art baselines, however, they have the advantage of performing segmentation by only processing below 60% of the video samples. The trade-off between coverage and accuracy can be observed. Additionally, TRN21/41 FI also have substantially better event-based metrics than frame-based methods due to its formulation.



(a) An example of video77 from Cholec80 processed by Trans-SV and TRN41 RMI

(b) An example video from Sacrocolpopexy processed by Trans-SV and TRN81 FI

Figure 14: Color-coded ribbon illustration for two complete surgical videos from (a) Cholec80 and (b) Sacrocolpopexy processed by Trans-SV and TRN models.

For sacrocolpopexy, we display a case where our partial-coverage models (TRN21/41 FI) are at their best in terms of computational efficiency. These are very long procedures and we are interested in only the suturing phases, therefore, a huge proportion of the video can be ignored for a full segmentation. Our models slightly under perform all baselines in frame-based metrics, but achieve this result by only looking at under 20% of the videos on average.

4.5 Conclusion

In this chapter, We proposed a new formulation for surgical workflow segmentation based on phase transition retrieval (instead of frame-based classification), and a new solution to this problem based on multi-agent reinforce-

ment learning (TRN). This poses a number of advantages when compared to the conventional frame-based methods. Firstly, we avoid any frame-level noise in predictions, strictly enforcing phases to be continuous blocks. This can be useful in practice if, for example, we are interested in time-stamping phase transitions, or in detecting unusual surgical workflows (phases occur in a non-standard order), both of which are challenging to obtain from noisy frame-based classifications. In addition, our models with partial coverage (TRN21/41/81 FI) are able to significantly reduce the number of frames necessary to produce a complete segmentation result.

Nevertheless, the TRN model is limited to offline segmentation tasks, and because each agent corresponds to an individual model, it becomes significantly more challenging to deploy on datasets with a larger number of phases. These constraints will be addressed in the next chapter. Meanwhile, the evaluation metric remains open for discussion. The Ward metric offers detailed insights into the sequence's segmental information, it is not as straightforward for comparing different methods. More generalized and intuitive approaches would be advantageous to adopt.

5 ATRN: A multi-purpose model for retrieving and anticipating surgical phase transitions

In the previous chapter, a novel set-up for surgical workflow segmentation was proposed, focusing on retrieving transitions rather than conducting frame-level classification. Based on this set-up, a reinforcement learning-based approach was developed, demonstrating results comparable to state-of-the-art methods. However, enhancing the performance of the reinforcement learning-based approach is challenging when advanced algorithms such as PPO [85], SAC [37], or DDPG [88] are employed. Preliminary experiments conducted with these models indicate that the sophisticated nature of the input and continuous output action space making these models very hard to converge in training.

Even though the advanced reinforcement learning algorithms did not yield satisfactory performance, the network architecture remains highly beneficial for the transition-retrieving configuration which guarantees continuity in the predicted phases and obtains a favorable event-based metric by identifying precise transition points between phases. In this chapter, we reverted to supervised learning for algorithm training but retained the continuous output action space to explore the potential of this configuration. Consequently, the size of the receptive window step is no longer restricted to 1 frame and can span an arbitrary number of frames. This new output format simultaneously enables the model to perform the surgical workflow anticipation task. Surgical workflow anticipation is the task of predicting the timing of relevant surgical events from live video data during a surgical procedure. As it has been mentioned in Chapter 2, it is critical in Robotic-Assisted Surgery (RAS) as it can enhance preparation and coordination within the surgical

team, improving surgical safety and the efficiency of operating room usage.

Several improvements, including adapting the transition-retrieving configuration for online applications, have also been made to the TRN architecture, which will be detailed in this chapter. Additionally, while the Ward metric offers a comprehensive understanding of the models' segmental performance, it is less intuitive to compare different methods. In this chapter, we employed EDIT score and F1@k, the two most widely used benchmark metrics in general activity segmentation tasks, to evaluate our method and compare it with state-of-the-art techniques on three datasets: Sacro56, Cholec80, and Cataract101.

5.1 Methodology

Figure 15 illustrates the new purpose architecture, the Aligned Transition Retrieval Network (ATRN). The network is composed of three main steps that are independently trained: video preparation and feature encoding, ATRN transition retrieving agent (TRA) aggregation, and task-specific pipeline. Depending on the specific usage of the network in offline, online, or anticipation mode, the output from the ATRN transition retrieving agents (TRA) aggregation network will be interpreted differently and be processed with different pipelines. This section provides details on the implementation of ATRN.

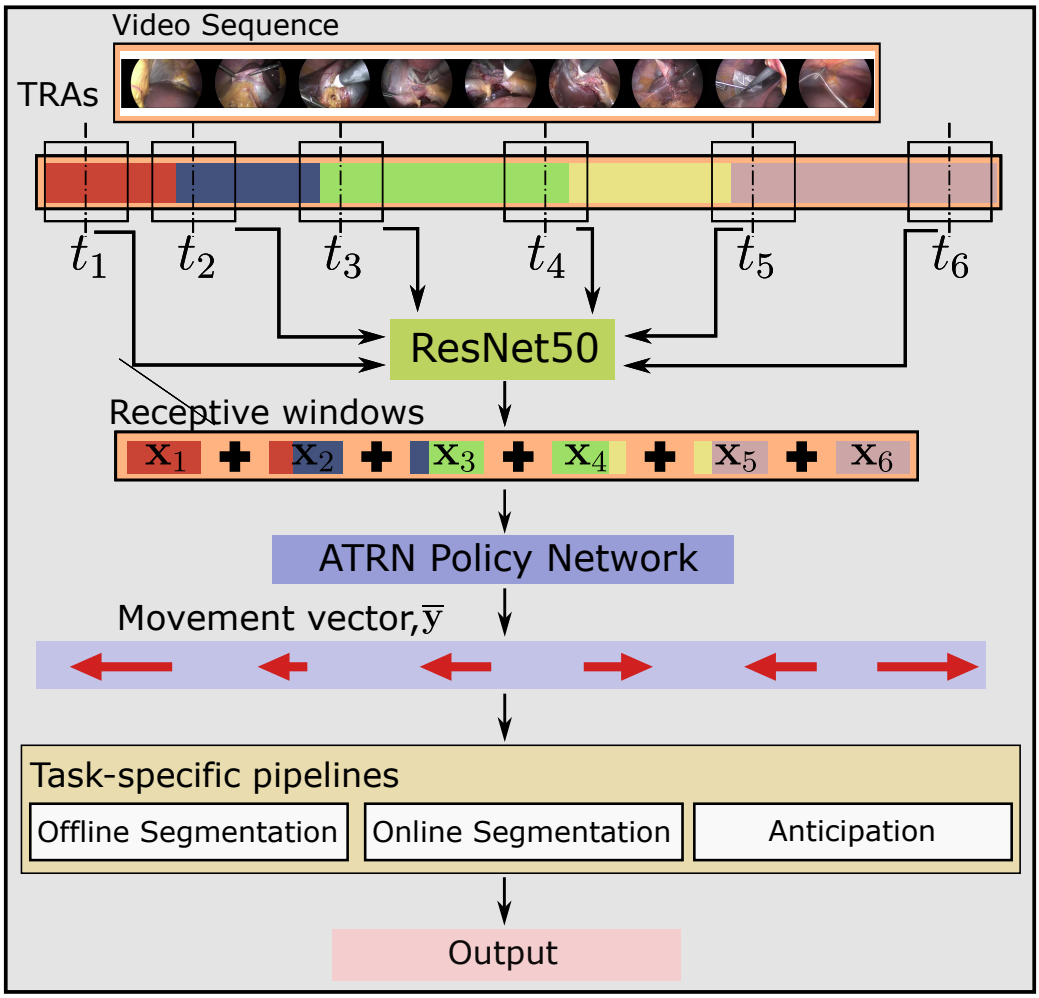


Figure 15: The overall architecture of ATRN. For each input video sequence, the ResNet50 is used to encode frames into a feature sequence. Each Transition Retrieval Agent (TRA) collects the content within its own receptive window and then feeds it into the ATRN policy network after concatenation. This process aims to obtain the movement vectors of TRAs, which are used to estimate movement/anticipation towards their target transitions based on the specific tasks. The movement vectors can be further processed by three different pipelines, each designed for a specific task (offline/online segmentation and anticipation), in order to make a final prediction.

5.1.1 Feature Encoding

We use ResNet50 as our base image feature encoder, which has been trained by a self-knowledge distillation algorithm [104], and has an output feature vector size of 2048. Instead of extracting features from every single video frame, we let a set of Transition Retrieval Agents (TRAs) to decide which frames are needed. This selection process is described in the remainder of this section.

5.1.2 Transition Retrieving Agent (TRA)

The Transition Retrieving Agent (TRA) is the fundamental unit in our architecture. Each TRA aims at detecting either the start or the end of each phase. Therefore, for a surgery with P phases we consider $I = 2P$ TRAs, each with a corresponding target phase transition. A TRA A_i receives as input a feature sequence $\mathbf{x}_i$ from its receptive field, i. e. features extracted from window of frames centred at index t_i, with length L. It produces as output the estimated distance $\overline{y}_i$ between t_i and the target transition.

$$\overline{y}_i = A_i(\mathbf{x}_i) \tag{16}$$

During inference, the centre of the receptive field t_i will be adjusted automatically, while L is a fixed hyperparameter of the architecture. In our implementation, the distance y_i is bounded by a maximum value S and is further normalised in the interval $[-1, 1]$. To better handle different receptive field sizes, all extracted feature sequences $\mathbf{f}$, regardless of their length L, are downsampled to a fixed 21-length sequence $\mathbf{x}_i$, so that A_i has an input with fixed size. The following equation outlines this downsampling process:

$$\mathbf{x}_i(t_i) = \begin{bmatrix} [\frac{21}{L}] \sum_{j=[t_i-\frac{L}{2}+\frac{L(m+1)}{21}]}^{[t_i-\frac{L}{2}+\frac{L(m+1)}{21}]} \mathbf{f}_j \\ \cdots \\ \cdots \\ [\frac{21}{L}] \sum_{j=[t_i-\frac{L}{2}+\frac{LM}{21}]}^{[t_i-\frac{L}{2}+\frac{L(M+1)}{21}]} \mathbf{f}_j \end{bmatrix} \forall M \in [0, 21] \tag{17}$$

In the downsampling stage outlined in equation 17, the receptive field of length L is initially partitioned into 21 equal segments. The feature vectors within each segment are then averaged, producing a single feature vector for each segment, culminating in a final receptive window consisting of 21 feature vectors.

5.1.3 Aligned Transition Retrieving Network (ATRN)

Instead of having TRAs operating independently in a similar way in Chapter 4.1.1, the Aligned Transition Retrieving Network (ATRN) processes the information from all agents simultaneously, which enables modeling interphase relationships. For this purpose, we concatenate all receptive windows $\mathbf{x}_{0:I}$ into a unified input $\mathbf{X}$. ATRN contains a set of TRAs with receptive fields of length L and centre positions t_i as the elements of vector $\mathbf{t}$. The output of ATRN is a movement vector $\overline{\mathbf{y}}$ of TRAs containing the distance from each t_i in to its corresponding target phase. Thus, equation (16) can be rewritten as:

$$\overline{\mathbf{y}} = \mathbf{A}(\mathbf{X}) \tag{18}$$

where $dim(\overline{\mathbf{y}})$ is (N, I), representing the batch number N and, the number of transitions required for retrieval respectively.

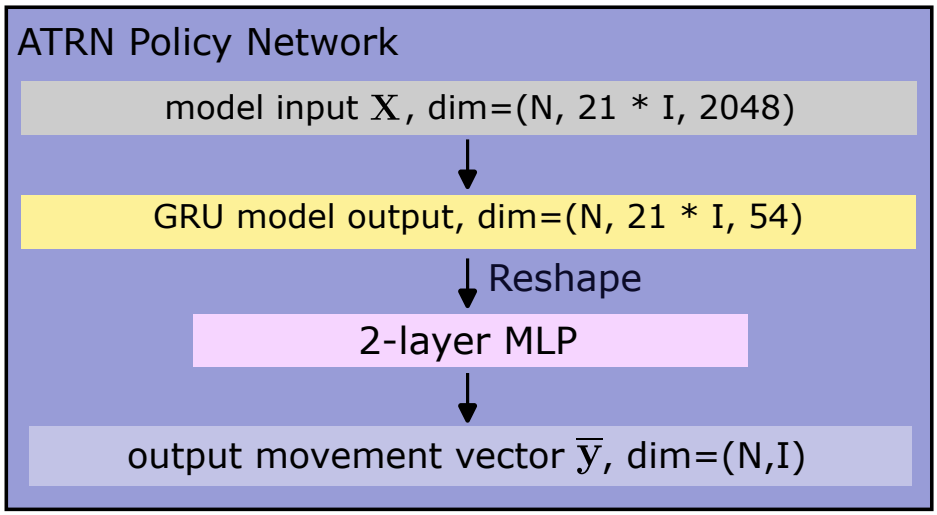


Figure 16: The detailed architecture of the ATRN policy network, this network takes the concatenated features from TRAs $\mathbf{X}$ of dimension (N,21*I,2048) and output the movement vector $\overline{\mathbf{y}}$ with dimension (N,I) where N represents the batch size used for ATRN training and I represents the number of transitions to retrieval.

In Figure 16, the architecture of ATRN policy network is shown, where the network $\mathbf{A}$ consists of a BiGRU sequential backbone, followed by two MLP layers and a tanh activation layer at the end. This final layer is used to convert the output into the range $[-1, 1]$. The same ATRN backbone is used for all 3 tasks (online/offline segmentation, anticipation), however, a dedicated task-specific pipeline is proposed for each of them.

5.1.4 Task-specific pipelines

ATRN predicts the normalised distance of each agent to their target transition. However, this prediction requires further interpretation to convert it into the desired format. We have designed three distinct pipelines for offline segmentation, online segmentation, and anticipation tasks. The illustraion of these three piplines can be found in Figure 17.

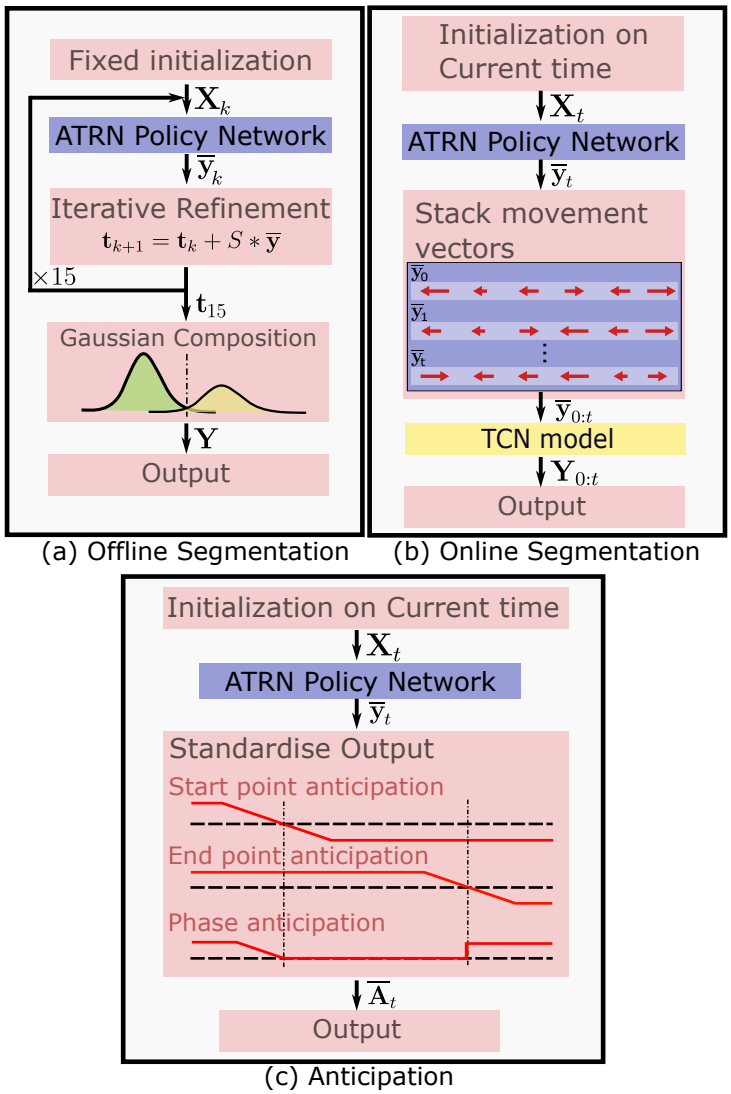


Figure 17: Task-specific pipelines for ATRN include (a) an offline segmentation pipeline with fixed initialization where TRAs are set at the average percentage positions for their target transitions. ATRN is used recursively to converge the TRAs to the target transitions, and Gaussian composition synthesizes the transitions into phase predictions. (b) An online segmentation pipeline initializes all TRAs at the current time step, utilizing ATRN output as features, which are then fed into a TCN model to predict phase performance online. (c) An anticipation pipeline directly employs ATRN output as anticipation predictions of transitions, transforming the transition anticipation signals (beginning and end of phases) into anticipation of each phase.

Offline phase segmentation We initialise the centre of receptive fields with a prior $\mathbf{t}_0$ based on the average timestamp of each transition in the training data. In the remainder of this paper, we denote this prior as fixed initialization. Our TRAs then predict the distances from $\mathbf{t}_0$ to the target transitions. We iteratively repeat this process according to

$$\mathbf{t}_{k+1} = \mathbf{t}_k + S * \overline{\mathbf{y}}_k \tag{19}$$

Vector $\mathbf{t}$ represents the positions of all agents, and k is the step index in the iteration.

We expect that $\mathbf{t}_k$ eventually converges to the correct position, and that a good initial $\mathbf{t}_0$ reduces the number of iterations needed. In our implementations on inference, we employ up to 15 iterations, and in most instances, our approach converges well before reaching this limit. The trajectory of the receptive fields throughout this process defines which frames in the video need to be passed through a ResNet for feature extraction, named activated frames, which in general corresponds to a fraction of the whole video. This is the main mechanism through which we reduce computational cost when compared to standard frame-based segmentation approaches.

After 15 iterations, the end of one phase and the beginning of the next one are not necessarily aligned due to estimation errors. To produce a workflow with seamless transitions from one phase to another, we used the Gaussian composition (shown in Figure 18) as explained in the previous Chapter 4.1.2.

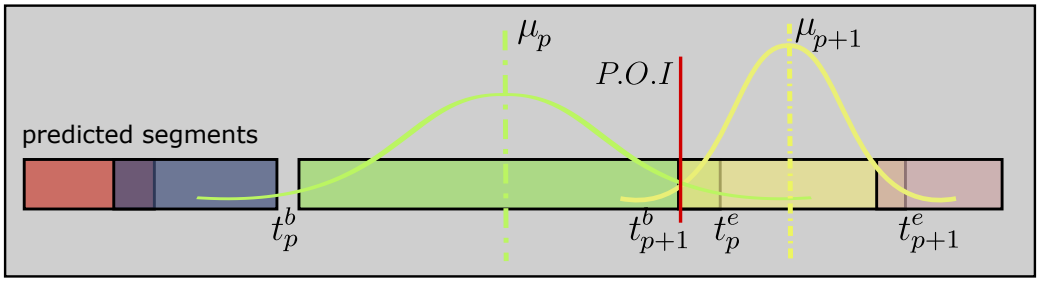


Figure 18: Illustration of Gaussian Composition

Online phase segmentation Unlike with our offline segmentation method, here we do not run ATRN recursively to refine over phase timestamps. Instead, we sequentially run ATRN once for each video frame, setting the receptive field of all TRAs to the current time step t. In addition, we feed the frame-based predicitons $\overline{\mathbf{y}}(t)$ to a temporal model B, a two-stage causal TCN [17, 58]. B receives as input a sequence of past and present vectors $\overline{\mathbf{y}}_{0:t}$, with the identical input dimension as the original TCN study, where the feature size is defined as the number of transitions. The output of B is the phase classification from time 0 to t.

$$\mathbf{Y}_{0:t} = B(\overline{\mathbf{y}}_{0:t}) \tag{20}$$

Online phase anticipation For anticipation, we aim at estimating the distance of each TRA from the current frame t to their target transition. In a similar manner to online segmentation, our anticipation pipeline also sequentially runs ATRN once at each frame, by setting the receptive field of all TRAs to the current time t. Consequently, at each time step t, the temporal distance to each phase transition can be computed as:

$$\overline{\mathbf{a}}(t) = \overline{\mathbf{y}}(t) * S \tag{21}$$

The anticipation result at time t of any phase occurring in the future is defined as the temporal distance to its beginning transition, a value contained in $\overline{\mathbf{a}}(t)$. To be consistent with prior work on phase anticipation [46,80,102] we bound our anticipation predictions by a maximum distance value T, smaller or equal to S. Please note that the S bound is a tunable hyper-parameter of our model, while the T bound is set to a fixed value that makes our anticipation results comparable to prior work. While we could simplify our formulation by making $S = T$, we observe empirically in our experiments that having $S > T$ achieves better performance. We further define that an ongoing phase should have a zero value for anticipation, and that past phases

that do not occur anymore should have a maximum value for anticipation (T). We therefore define our final anticipation result $\overline{A}_p(t)$ for phase p as:

$$\overline{A}_p(t) = \begin{cases} \min(a(t)_p^b, T) & a(t)_p^b > 0 \\ 0 & a(t)_p^b \leq 0 \end{cases} + \begin{cases} 0 & a(t)_p^e \geq 0 \\ T & a(t)_p^e < 0 \end{cases} \tag{22}$$

where $a(t)_p^b$, $a(t)_p^e$ stand for anticipation of the beginning and end of phase p respectively.

5.1.5 Training procedure

The training of our pipelines is done in the following order: 1) Training the ResNet50 encoder using self-knowledge distillation [104]; 2) Training ATRN; 3) For online segmentation only, training the TCN temporal model.

For training the ATRN model, we define a training sample as randomly selecting a video from the training set and then setting each TRA receptive field to a random position within the video. We further define a batch as 128 independent training samples. In each training iteration, a batch is sampled and ATRN is updated according to the objective function described in Section 5.1.5. For online phase segmentation, training the TCN follows equivalent methodology to other works [17, 31, 94], using a negative logarithmic likelihood loss function.

ATRN Objective Function Since ATRN distance predictions are bounded by the interval $[-S, S]$, any TRA outside this interval will output values saturated at S or $-S$. Assuming that the length $2S$ is very small compared to the entire length of the video, the random sampling of TRA receptive fields (described in section 5.1.5) introduces a significant training data imbalance towards saturated values. To counter this problem, we introduce a discount factor $\mathbf{d}$ that reduces the loss of each sample as its output becomes further from the interval $[-S, S]$. For a smooth loss function, we model this discount

factor with a Gaussian function decay:

$$\mathbf{d} = exp(-\frac{\mathbf{y}^2}{2\sigma^2}) \tag{23}$$

where σ determines how quickly the loss decreases when the sample output $\mathbf{y}$ increases. The choice of σ is directly dependent on the receptive field length $2S$. The complete ATRN objective function is the dot product of the standard MSE error and the discount factor, multiplied by the square of the maximum step size:

$$l(\mathbf{y}, \overline{\mathbf{y}}) = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{y} - \overline{\mathbf{y}})^2 \cdot \mathbf{d} \cdot S^2 \tag{24}$$

where N is the training batch size.

5.1.6 Implementation details

We utilize the pytorch framework to deploy our model on a single NVIDIA RTX A6000 48GB card for both training and inference. We train ResNet (Section 5.1.1) for 100 epochs using the SGD optimiser with a momentum of 0.9, weight decay of 1e-5, learning rate of 5e-4, and batches of 64 randomly sampled frames from all videos in the training set. Random clipping, color jittering, and flipping are employed as augmentation [35]. After training the ResNet, we proceeded to extract and save the features in order to expedite the subsequent training processes. We train ATRN (Section 5.1.3) for 8000 epochs using the ADAM optimiser with a learning rate of 1e-4 and a batch size of 128. This training takes around 24 hours on the above mentioned hardware. We train TCN (Section 5.1.4) using the ADAM optimiser with a learning rate of 1e-1, and a batch size of 1. The video input is randomly cut into arbitrary length clips for trainining. It is worth noting that all settings used for training this TCN are kept the same for training other state-of-the-art methods reported in the experiments.For both training and inference,

all videos are downsampled to 1fps and frames are resized into 224*224 for saving memory and reducing network parameters.

5.2 Experiments

5.2.1 Evaluation Metric

For validating our phase segmentation methods, we report two types of evaluation metrics: frame-wise (accuracy, F1 score) and segmental metrics (EDIT score, F1@k), as described in Chapter 2.3. Same as in the TRN work, for offline phase segmentation, we also report the coverage rate, i. e. the percentage of frames used as input to ATRN out of the total number of frames in a video. This metric is calculated per video and then averaged. This is an indicator of computational efficiency, since ATRN only requires extracting ResNet features for this fraction of input frames instead of the whole video. Please note that all conventional frame-based state-of-the-art methods have by definition a coverage of 100% for all cases, as all individual frames are processed by the feature extractor network. For brevity, the ablation experiments (Section 5.2.2) only report accuracy, Edit score, and coverage rate, while our comparisons with the state-of-the-art (Section 5.2.3) report all metrics. To evaluate phase anticipation, we report the same metrics utilised in previous works [80, 101], iMAE and eMAE, with a threshold h of 5 minutes. The detailed equation is provided in Section 2.3.3.

5.2.2 Ablation and hyper parameter selection

For all 3 tasks (online/offline segmentation and anticipation), we perform independent ablation experiments for the TRA receptive field length L, the discount factor parameter σ, and the maximum TRA output distance S. All reported results are on the validation set of Cholec80, which we then use to select the best parameters for all following experiments on all datasets.

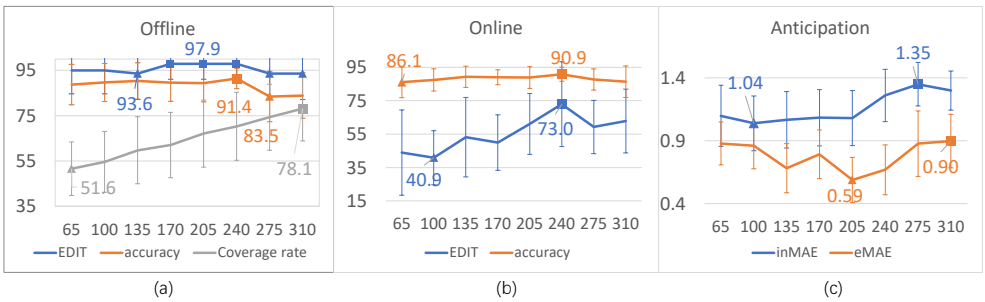


Figure 19: Ablation study on TRA receptive field length L (Eq. **??**)

TRA receptive field length (L) Figure 19 shows the ablation results for L. In principle, a larger L allows ATRN to gather wider temporal information around each TRA, but at the cost of losing low-level temporal resolution due to downsampling (as described in Euqation 17) and it also increases computation time at both training and inference. Results show that increasing L has little influence on accuracy, while increasing the Edit score for online segmentation only. On the other hand, increasing L has a clear impact in decreasing computational efficiency, as showed by the increasing coverage rate of offline segmentation. As a trade-off between accuracy and computational efficiency we pick a value $L = 135$ on the lower end of our ablation range and it is used for the rest of the ablation study.

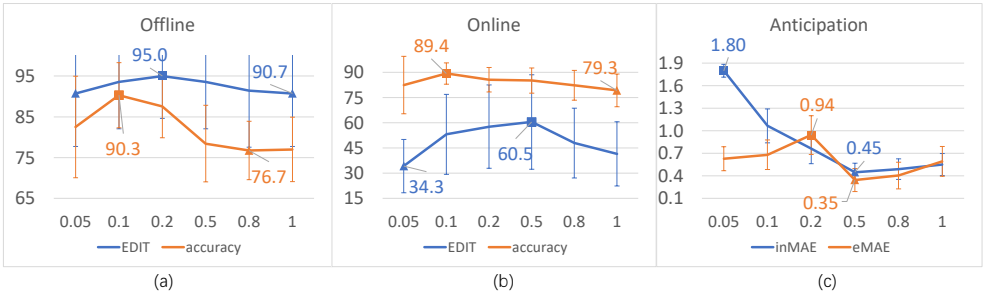


Figure 20: Ablation study on the discount factor decay σ (Eq. 23)

Discount factor decay (σ) Figure 20 shows the ablation results for σ. A lower σ helps alleviate the data imbalance caused by ATRN output saturation to the range $[-S, S]$. However, it also decreases prediction accuracy whenever a TRA is distant from its target phase transition, since any sample in these conditions results in a severe discount factor. A high σ value is especially beneficial for anticipation, since this pipeline is expected to perform accurate predictions when a TRA is still far away (up to time T) from its future target transition. On the other hand, phase segmentation models are expected to perform accurate predictions when TRAs are closer to their targets. In the offline case this is due to TRA positions iteratively converging to targets, and in the online case this is due to focusing only on the present phase (i. e. TRAs with lowest distance to target). Therefore, we observe that the optimal σ values for segmentation models are significantly lower than for anticipation. However, both the online and offline accuracies are largely degraded with the increase of the discount factor. Therefore, we select two different models respectively with $\sigma = 0.1$ (optimised for segmentation) and of $\sigma = 0.5$ (optimised for anticipation). In our remaining phase segmentation experiments, we report results with the first model, while in our phase segmentation experiments we report results with both models.

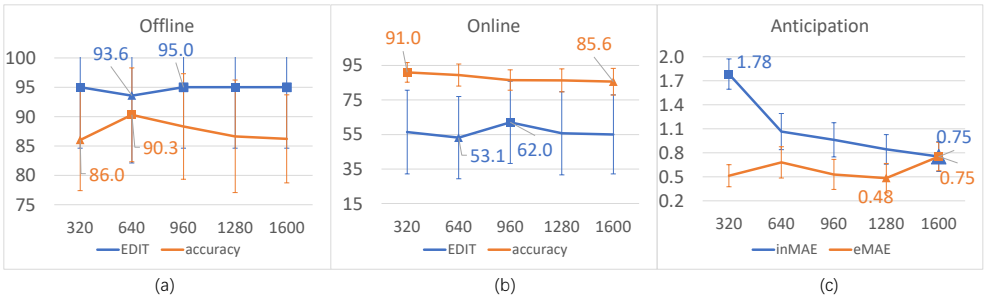


Figure 21: Ablation study on S, which denotes the maximum iterative step size for offline segmentation (eq. 19), and maximum predicted distance for remaining models (eq. 21).

Maximum distance (S) Figure 21 shows the ablation results for S. In the offline segmentation case, it determines the farthest distance a TRA can move in each iteration. For the other cases, it simply denotes the value range for elements of vector $\overline{\mathbf{a}(t)}$ by definition. in range of the ATRN output. We observe from experimental results that S has limited influence on phase segmentation performance. In the case of offline segmentation, this suggests that our proposed fixed initialisation is a good prior, as TRAs converge well to their targets with small iterative movements. For anticipation, increasing S seems to have a positive effect in inMAE, but limited impact on eMAE. Similar to σ, an S too small causes an anticipation performance drop for distant phases. For all further experiments, we select $S = 640$, which corresponds to the maximum accuracy in offline segmentation. While this is not the optimal value for anticipation, only modest gains can be obtained with a significantly larger S.

5.2.3 Comparison with State-of-the-art

For phase segmentation, we compare our method against state-of-the-art surgical phase segmentation models (TeCNO [17], TransSV [31]), a general-purpose temporal segmentation model (MSTCN [58]), and also a bi-directional GRU model (BiGRU) [14]. For fairness, all methods use features from the same ResNet encoder and we also report its standalone performance for reference. All methods are trained utilising the same video pre-processing pipeline described in 5.1.6.

All baselines are trained for 50 epochs using a learning rate of $1e-3$ and employing a weighted cross-entropy loss that is weighted by the reciprocal of the proportion of the frame number of each class in the total number of frames. The model parameters that yield the highest F1 score on the validation set are chosen as the final parameters for the model.

The architecture design hyper-parameters of each baseline are kept the same as their respective publications. Noticeably, TeCNO and MSTCN correspond to the same overall architecture, but with distinct hyper-parameters. MSTCN consists of 2 stages with 10 layers of temporal blocks in each stage and a feature mapping dimension of 64. In contrast, TeCNO also has 2 stages but with 8 layers of temporal blocks in a single stage and a feature mapping dimension of 32. Also, the TeCNO network serves as the module for encoding temporal features in TransSV, similar to the original work.

All baselines that incorporate TCN blocks (TeCNO, MSTCN, TransSV) are implemented with non-causal and causal TCNs for offline and online inference respectively. BiGRU performs offline inference when we pass it through all video frames in both directions. For online inference, we pass it only through previous frames. For online inference, we also compare our method with and without the final TCN refinement for ablation purposes. For anticipation, we compare our method against the original work of BayesianDL [80], IIA-Net [101], and Trans-SVNet [46].

Table 7: Offline Phase Segmentation Comparison

Dataset	Model	f1@50	Edit Score	Accuracy	f1 Score	Coverage Rate (%)
Cholec80	ResNet	0.016 ± 0.012	2.93 ± 1.45	78.4 ± 9.14	0.737 ± 0.078	100
	MSTCN	0.332 ± 0.223	25.5 ± 18.5	90.8 ± 5.95	0.865 ± 0.064	100
	TeCNO	0.325 ± 0.188	25.4 ± 14.5	90.7 ± 5.59	0.858 ± 0.064	100
	TransSV	0.338 ± 0.206	26.6 ± 17.2	89.6 ± 6.51	0.847 ± 0.075	100
	SAHC	0.581 ± 0.198	51.7 ± 19.2	91.5 ± 4.16	0.860 ± 0.068	100
	BiGRU	0.487 ± 0.241	41.4 ± 24.0	**92.1 ± 5.50**	**0.877 ± 0.070**	100
	ATRN	**0.818 ± 0.178**	**94.6 ± 9.94**	89.6 ± 7.20	0.836 ± 0.103	**57.9 ± 12.4**
Sacro56	ResNet	0.001 ± 0.001	0.535 ± 0.177	86.1 ± 7.23	0.874 ± 0.027	100
	MSTCN	0.253 ± 0.148	24.3 ± 15.9	**93.7 ± 9.66**	**0.948 ± 0.030**	100
	TeCNO	0.134 ± 0.105	11.1 ± 9.36	91.6 ± 10.1	0.927 ± 0.042	100
	TransSV	0.246 ± 0.160	20.6 ± 12.2	91.6 ± 9.73	0.926 ± 0.036	100
	BiGRU	0.356 ± 0.185	26.8 ± 11.6	91.4 ± 9.85	0.939 ± 0.030	100
	ATRN	**0.860 ± 0.189**	**93.8 ± 15.4**	90.4 ± 11.5	0.890 ± 0.108	**36.0 ± 11.2**
Cataract101	ResNet	0.118 ± 0.079	15.4 ± 6.29	73.2± 7.47	0.703 ± 0.089	100
	MSTCN	0.626 ± 0.200	66.3 ± 17.4	86.9 ± 8.47	0.844 ± 0.093	100
	TeCNO	0.655 ± 0.221	70.9 ± 18.3	86.4 ± 8.60	0.838 ± 0.099	100
	TransSV	0.664 ± 0.219	71.3 ± 17.8	86.2 ± 8.15	0.831 ± 0.088	100
	BiGRU	0.747 ± 0.160	88.1 ± 11.4	**88.0 ± 6.61**	**0.857 ± 0.082**	100
	ATRN	**0.769 ± 0.136**	**94.2 ± 7.14**	84.3 ± 5.20	0.820 ± 0.065	**97.3 ± 4.40**

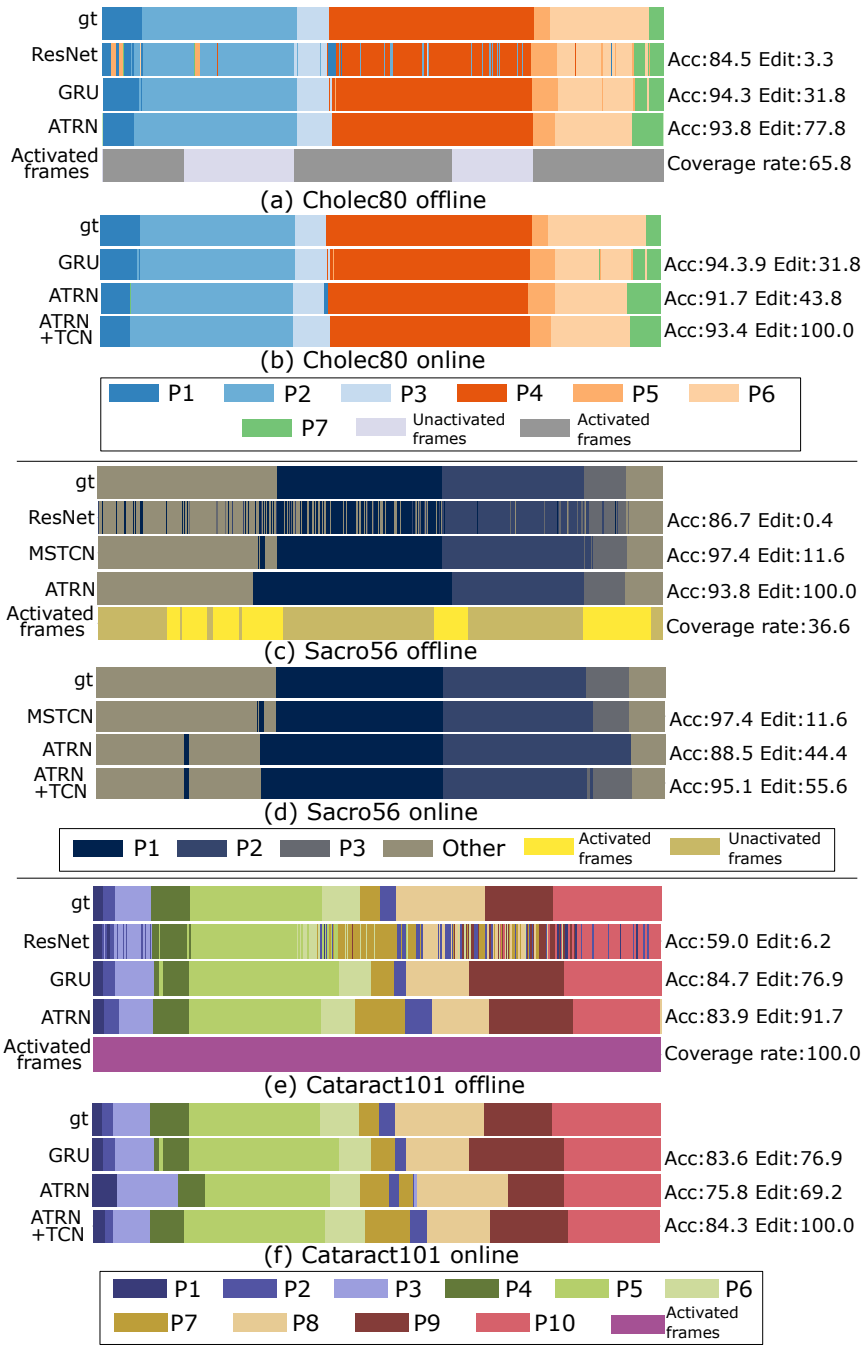


Figure 22: Color-coded ribbon illustration for the comparisons of workflow on three datasets, whose horizontal axis represents the time progression.The offline plots for each datasets also contain extra plots for the activated features used for ATRN predictions.

Offline Phase Segmentation Table 7 presents comparisons for offline workflow segmentation. BiGRU yields the best frame-wise performance on two datasets (Cholec80 and Cataract101), despite being the simplest architecture in our comparison. MSTCN has the best frame-wise performance on Sacro56. We note that this model has the same architecture of TeCNO, with a different selection of hyperparameters.

Especially on Cholec80 and Sacro56, our proposed method ATRN significantly outperforms all conventional frame-based methods on segmental metrics (f1@50 and Edit Score), at a cost of a slight under-performance in frame-wise metrics (accuracy and f1 score). We acknowledge a limitation in frame-wise evaluation. There is an inherent margin of error in frame-level ground truth labels. The precise frame where each phase starts or ends is subject to variations by human annotators. Consequently, when the performance differences between methods are minimal (as seen in Cholec80), the importance of "outperforming" or "underperforming" in f1-score may be negligible. Conversely, ground truth labels are much less prone to errors from a segmental perspective, as it would require an annotator to entirely miss or swap a whole phase. We believe the higher segmental performance of ATRN is due to its configuration, which considers phases or events as the basic units for the network, unlike previous methods that focused on individual frames.

To illustrate why and when this trade-off may be advantageous, we provide examples of full-video workflow predictions in Figure 22. Results with high Edit score correspond to predictions with the correct ordering of phase transitions, while low edit scores produce incorrect phase transition events. In Figure 22 (c) we can observe a case where ATRN has higher Edit score, but lower accuracy than MSTCN. The end result is that while MSTCN classifies correctly a higher number of frames, it introduces several incorrect transitions between phases 1 and 2, and between phases 2 and 3. ATRN

has thus a more accurate picture of the high-level workflow and ordering of phases. We therefore argue that for any end-applications where phase transitions matter (e. g. recognising normal vs. abnormal workflows) we need to analyse the trade-off between segmental and frame-wise metrics. However, in Cataract101, the advantage of ATRN in segmental metrics is much less pronounced. We believe this is a result of the much shorter phase durations in the procedure, which enable the baseline temporal models to capture sufficient context and predominantly segment entire phases as contiguous intervals.

In the context of surgery, challenging scenarios may arise in phase segmentation tasks, which we have classified into three types: missing phases, phase swapping, and phase repetition. For these scenarios, we have noticed that the model tends to predict transitions at the beginning and end of missing phases to be adjacent or very close, resulting in a phase duration that is zero or very short. It is important to note that such cases are present in the Cholec80 dataset and are reflected in the reported performance metrics. Therefore, missing phases have a minimal impact on the network's overall performance. Regarding phase swapping, ATRN demonstrates the ability to recognize such cases using the Cholec80 dataset. However, our current method cannot handle an arbitrary number of repeated phases, which is an area for future improvement in ATRN.

The offline ATRN architecture also has significant computational efficiency advantages since by design it only needs to extract image features for a fraction of all video frames during inference (measured by coverage rate in Table 7). The duration of the procedure and its phases is highly correlated with the magnitude of these computational gains. Sacro56 corresponds to the longest procedure (3 phases over >3 hours) and also the largest computational gains with ATRN only needing to process less than half of the entire

video on average (36%). In Cholec80 (7 phases over >30 minutes) ATRN needs to process only slightly more than half the video (58%). And finally, Catarac101 is the shortest procedure (10 phases over <20 minutes) and corresponds to marginal gains in computational efficiency, as ATRN processes on average 97% of the entire video.

Online Phase Segmentation The observations for online segmentation are similar to those of offline segmentation. Conventional frame-based baselines show little difference in performance when performing online inference. As Table 8 shows, for the Chlec80 and Sacro56 datasets, ATRN (without TCN) has a slight decrease in frame-wise metrics, while for Cataract101 there is a significant drop. We attribute this decrease in performance to the cataract workflow characteristics, with large number of very short phases. Coventional configurations typically display a low event-based metric, which is attributed to the fragmentation in the predicted phases. Figure 22 illustrates the presence of randomly distributed false negative predictions, with ResNet, GRU, and MSTCN predictions generally exceeding those of ATRN+TCN methods. It is surprising to observe that even without the Gaussian composition, which enforces strict continuity on predictions, the standalone ATRN network still boasts a superior event-based metric compared to most other methods except for SAHC and achieves optimal performance with the integration of an additional TCN network.

In online inference, ATRN (without TCN) has a relatively small short-term temporal information. When online ATRN is combined with TCN, longer temporal context becomes available to the model and thus ATRN's frame-wise performance becomes in line with state-of-the-art. Integrating TCN with ATRN markedly enhances segmental metrics, surpassing all baseline methods. In the architecture of our network, anticipation-related features (outputs from ATRN) represent fine-level temporal information, while TCN

Table 8: Online Phase Segmentation Comparison

Dataset	Model	f1@50	Edit Score	Accuracy	f1 Score
Cholec80	ResNet	0.016 ± 0.012	2.93 ± 1.45	78.4 ± 9.14	0.737 ± 0.078
	MSTCN	0.332 ± 0.223	25.5 ± 18.5	90.8 ± 5.95	0.865 ± 0.065
	TeCNO	0.324 ± 0.189	25.4 ± 14.4	90.6 ± 5.58	0.857 ± 0.064
	TransSV	0.340 ± 0.207	26.6 ± 17.2	89.6 ± 6.50	0.847 ± 0.075
	SAHC	0.578 ± 0.199	51.7 ± 19.2	91.3 ± 4.15	**0.857 ± 0.068**
	BiGRU	0.488 ± 0.240	41.4 ± 24.0	**92.2 ± 5.46**	**0.875 ± 0.070**
	ATRN	0.54 ± 0.187	49.9 ± 21.1	88.6 ± 5.65	0.820 ± 0.072
	ATRN + TCN	**0.768 ± 0.223**	**78.1 ± 21.9**	89.8 ± 6.81	0.855 ± 0.080
Sacro56	ResNet	0.001 ± 0.001	0.535 ± 0.177	86.1 ± 7.23	0.874 ± 0.027
	MSTCN	0.253 ± 0.148	24.3 ± 15.9	**93.7 ± 9.65**	**0.948 ± 0.030**
	TeCNO	0.133 ± 0.105	11.1 ± 9.37	91.6 ± 10.1	0.927 ± 0.042
	TransSV	0.246 ± 0.160	20.6 ± 12.2	91.6 ± 9.73	0.926 ± 0.036
	BiGRU	0.356 ± 0.185	26.8 ± 11.6	91.4 ± 9.85	0.939 ± 0.029
	ATRN	0.278 ± 0.165	21.3 ± 12.5	91.3 ± 8.50	0.930 ± 0.022
	ATRN + TCN	**0.498 ± 0.182**	**46.2 ± 15.9**	91.1 ± 8.99	0.891 ± 0.072
Cataract101	ResNet	0.118 ± 0.079	15.4 ± 6.29	73.2± 7.47	0.703 ± 0.089
	MSTCN	0.620 ± 0.190	66.5 ± 17.3	86.4 ± 8.79	0.834 ± 0.098
	TeCNO	0.650 ± 0.225	71.4 ± 18.5	85.7 ± 8.81	0.829 ± 0.102
	TransSV	0.660 ± 0.221	71.3 ± 17.8	86.1 ± 8.59	0.827 ± 0.095
	BiGRU	0.742 ± 0.168	88.6 ± 11.7	**87.7 ± 7.06**	**0.849 ± 0.088**
	ATRN	0.503 ± 0.187	64.2 ± 17.2	62.9 ± 18.0	0.669 ± 0.146
	ATRN + TCN	**0.776 ± 0.160**	**94.8 ± 8.22**	86.3 ± 6.52	0.830 ± 0.081

processes causal-level temporal data for segmentation. Combining anticipation and segmentation tasks into a hybrid model may be crucial to achieving superior performance in online segmentation metrics, as anticipation-related features could provide excellent guidance for the network in the phase segmentation task. For Cholec80 and Cataract101 datasets, online and offline segmental performance is similar, but for Sacro56 there is a performance decrease from offline to online.

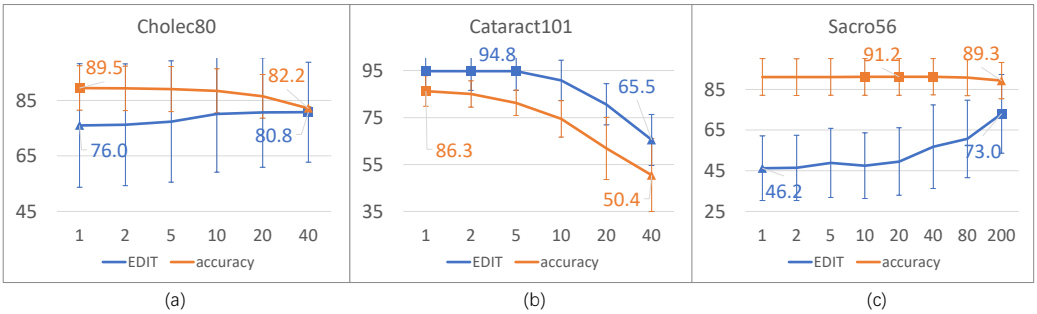


Figure 23: The influence of increasing frame skipping size on online segmentation performance for each dataset, where the x-axis denotes skipping size(in frame).

Besides the standard approach of processing each frame for online mode, our unique online segmentation pipeline enables the model to make predictions without the need to analyze every frame. We propose a frame skipping method with a skipping factor g, which involves using every g^{th} frame as input. This is akin to temporal downsampling of the video, thereby lowering the computational load. We tested the influence of varying the frame skipping size on network performance.

We used frame skipping factor g of 1, 2, 5, 10, 20, 40 frames for Cholec80 and Cataract101, and two additional frame skipping size of 80 and 200 frames for the longer videos in Sacro56. As illustrated in Figure 23, the impact of adjusting the frame skipping size varies depending on the length of the

Table 9: Computation Cost per Video

Method	ResNet	MSTCN	TeCNO	TransSV	SAHC	GRU	ATRN(Offline)	ATRN(Online)
Avg. offline processing time per video (s)	14.2	14.4	14.4	14.4	14.7	14.4	8.74	14.7

sequence. In the case of short videos (Cataract101), changing the frame skipping size significantly affects the performance. However, for moderate length videos (Cholec80), increasing the frame skipping size has minimal impact on the overall performance. Interestingly, for long videos (Sacro56), using a sparser frame skipping size can actually enhance perfromance on segmental metrics and significantly reduce the computational cost. Compared to conventional methods, where the input length is always increasing, ATRN can restrict the amount of data input to a predetermined size, thus reducing the amount of computing power needed and allowing for quicker interaction with online inference.

Computational cost The average processing times on Cholec80 for ATRN and state-of-the-art methods are detailed in Table 9. As indicated in the table, the offline version demonstrates superior computational efficiency compared to all other methods, as the feature encoding process is the most computationally intensive part, which the ATRN offline method avoids to process all features. Consequently, our offline method is 'faster' than the online method when measured in average FPS over entire videos due to its lower coverage rate.

It is worth to emphasize that Temporal Convolutional Networks (TCN) and all TCN-related methods (such as Trans-SV) utilize causal convolution, which leverages the low-level hidden layers of the network to ensure the model does not violate the ordering of the input data, thus enabling online detection. However, no previous work has explicitly calculated the real computational cost associated with online detection, as this is closely tied to

Table 10: Anticipation Comparison

Results	Comparison of state-of-the-art on Cholec80					ATRN performance on other datasets	
	BayesianDL	IIA-Net*	Trans-SVNet	ATRN(General)	ATRN(anticipation)	Cataract101	Sacro56
iMAE	1.17	1.08	1.07	0.905	**0.427**	1.68	0.321
eMAE	1.37	1.09	1.26	0.703	**0.369**	0.11	0.258

Note: The IIA-Net uses tool existing signal in trainning where other methods use pure vision input

the low-level code implementation of the models. Theoretically, the online computational cost should be comparable to the offline computational cost for these models. Given this complexity in assessing online computational costs and the lack of a universal standard, we have only provided the computational cost for offline detection.

To evaluate online performance, we use the input sequence from time 0 to t to make predictions at time t, thereby simulating an online processing scenario for these techniques. It is important to note that some baseline methods (such as BiGRU) are inherently offline and need to process all frames at once. Consequently, their 'online' version involves continuously increasing the number of input frames until the entire video is available at the end.

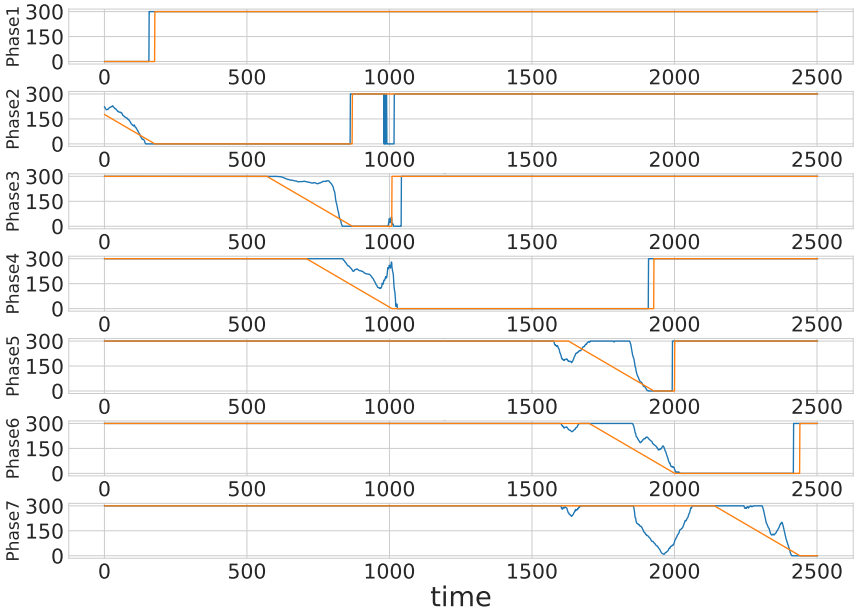


Figure 24: An example video of anticipation results for Cholec80 with a threshold of 300 seconds (5 mins). The vertical axis is the anticipation prediction in seconds and the horizontal axis is the time axis of the video

Anticipation We compare our anticipation model against the state-of-the-art baselines on the Cholec80 dataset since we can directly compare with results reported on [46, 80, 101] using the same test data. Table 10 shows that our ATRN(general) model, i. e. with same hyperparameters as phase segmentation models, already has slightly better performance than all baselines. It is important to note here that IIA-Net uses additional tool signal for model supervision and inference, and thus requires additional labels to train comparatively to all other models including ours. When we optimise hyperparameters for anticipation, as described in Section 5.2.2, the performance significantly improves as denoted by the results of ATRN(anticipation). An example output from our method's anticipation results is shown in Figure 24. We also provide the anticipation performance of ATRN(general) on

Cataract101 and Sacro56 datasets. The good performance of ATRN in phase anticipation suggests that, unlike with online segmentation, we do not need a very large temporal context of past frames, and thus an extra TCN model is not needed in this case.

Model's superior anticipation performance compared to the state-of-the-art is its ability to simultaneously identifying all timestamps for all phases (both past and future) at each time instance t, rather than just predicting a single anticipation event. This method offers more contextual information for the final prediction than other baselines.

5.3 Conclusion

In this chapter, we thoroughly extended the transition-retrieving configuration with a new designed Aligned Transition Retrieval Network (ATRN). The ATRN is an enhancement of the TRN, designed for online/offline phase segmentation and phase anticipation, and has been evaluated in three distinct surgical procedures. The model resulting from this design choice (ATRN) closely follows the state-of-the-art in terms of low-level frame-level performance (f1-score, accuracy), while showing significantly better performance in detecting correct phase ordering and transitions (as measured by segmental metrics Edit score, F1@50). The ATRN offline segmentation model also introduces computational advantages, by producing complete video results while only using a fraction of its frames. We observe that the advantages of ATRN (both in segmental accuracy and computational efficiency) is more pronounced for procedures with larger phase durations.

As evidenced by our ablation study, ATRN's performance is not significantly affected by its key hyper-parameters, with a notable exception for the loss function discount factor when performing phase anticipation. This may justify training separate ATRN weights for segmentation and anticipation for

optimised performance. At present, ATRN does not handle instance segmentation, where a phase can occur a variable number of times during a single surgery. While this is not the case for any of the 3 datasets utilised in this work, it could be a relevant scenario in other application contexts. Exploring this extension is a potential future work avenue for further generalising our method. It is important to highlight that surgical workflow analysis methods have not yet been validated using clinically relevant metrics like complication rates, blood loss rates, mortality rates, or readmission rates. Although these metrics might be challenging to assess at present, they are worth considering in future advances in surgical workflow analysis.

6 Conclusions

This research explored surgical workflow analysis by comparing frame-based and event-based approaches. Past studies mainly relied on frame-wise classification and metric evaluation. Chapter 3 evaluated various frame-wise metrics using the public Cholec80 dataset and the in-house Sacro18 dataset, which features longer videos of laparoscopic sacrocolpopexy surgeries not included in public datasets. We employed both frame-based and event-based metrics, noting that while frame-based methods yielded strong results, event-based metrics struggled, especially with longer surgical videos.

In Chapter3, we introduced frame-based seq2seq models, such as LSTM and Transformer networks, for surgical workflow segmentation at a coarse level. This was the first time transformers, specifically TransSV, were used in this field, reflecting their general rise in machine learning. We have identified the drawback of frame-level detection, which generates numerous incorrect transitions. We then developed a direct method to detect transitions in Chapter 4, improving segmental performance by reducing frame-level noise and making phases continuous. We presented the multi-agent reinforcement learning model TRN for offline surgical workflow segmentation and expanded the Sacro18 dataset to Sacro38, comprising 38 videos. This offline segmentation method lowers computational costs since agents detect transitions using partial sequence ranges, reducing the need to process all frames. Specifically, less than 60% of frames were processed for Cholec80 and less than 20% for Sacro38. However, this model is specifically created for offline surgical workflow segmentation. Chapter 5 explores the transition-retrieving mechanism in detail, enhancing versatility and performance by introducing ATRN, a model featuring a continuous action output space. ATRN surpasses TRN by supporting both offline and online segmentation and anticipation, integrating all agents into one network to tackle scalability issues. The Sacro-

colpopexy dataset now has 56 videos. We use EDIT score and F1@k for event-based metrics. Comparing ATRN with state-of-the-art methods across three datasets (Cholec80, Cataract, and Sacro56), ATRN excelled in event-based metrics and matched frame-based metrics, also outperforming other models in anticipation tasks.

We point out that surgical workflow literature has almost exclusively focused on frame-level metrics in the past. This is misaligned with state-of-the-art in general activity segmentation outside the surgical field, where event-based metrics have been proposed and are used more often.

This research examined performance discrepancies in surgical workflow analysis between frame-based and event-based metrics. Frame-based metrics allow for a rapid and straightforward assessment of a network's performance. However, this approach often disregards the continuity of events and long-term temporal patterns. This omission is especially critical in surgical videos, where phases or steps may be reordered or repeated several times. Event-based evaluations effectively capture the accuracy of methods in detecting such information. Some challenges have already begun incorporating event-based evaluations into their result analyses [21]. We propose that evaluating these extra metrics offers valuable insights crucial for downstream applications, where accurate phase ordering is as important, if not more so, than individual frame accuracy. This is especially true in scenarios where particular phase sequences indicate the complexity of cases, the surgeon's experience, or potential complications.

In Chapters 4 and 5 of this study, a new methodology for detecting transitions in surgical workflow segmentation has been explored. It reveals a clear trade-off between transition-based detection and traditional frame-based approaches. Transition-based methods show superior performance in event-

based metrics, while frame-based methods excel in frame-based metrics. The selection of method depends on which type of performance is prioritized for the intended use. Notably, from Chapter 4 to Chapter 5, the frame-based performance of our transition-based approach has seen significant improvements. Moreover, transition-based detection aligns better with tasks involving surgical workflow anticipation. This approach can also be expanded into hybrid models for multi-purpose applications.

We have also introduced a new surgical workflow analysis dataset, known as the Sacrocolpopexy dataset. This specific surgery tends to be significantly longer than the current benchmark datasets for surgical workflow analysis, which exacerbates the issue of over-detecting transitions. The transition-based detection method may provide a better model for this dataset. Additionally, in clinical settings, suturing time serves as a crucial performance metric for surgeons. Conventional frame-based approaches struggle with accurately defining the start and end points of the suturing phases due to random errors in detecting transition points. Conversely, transition-based detection allows for a precise description of transition points, either during or after the surgery. Therefore, to simplify the model in the initial study discussed in Chapter 4 of TRN, we concentrated solely on identifying the mesh implant suturing phase within this dataset. After upgrading the model to ATRN in Chapter 5, the scope was expanded to encompass three primary phases: anatomy dissection, mesh implant suturing, and reperitonealisation.

6.1 Limitations and Future work

There are several improvements that can be pursued in future research. In the current architecture, the convolution neural network and the following transition-retrieving models are trained separately, but could be fine-tuned in an end-to-end fashion. Modelling surgery-specific priors can improve predictions on Cholec80 (SV-RCNet+PKI [44]) and similar strategies could be

developed for Sacrocolpopexy.

While it is uncommon in public datasets, there could be instances where phases repeat an unknown number of times, making our formulation inadequate. A specialized mechanism or model architecture in the transition-retrieving pipeline needs to be developed to address this issue. Furthermore, ATRN is sensitive to agent initialization, as a closer initialization to the target transition results in more precise predictions. Though we propose two working strategies (FI, RMI), they could be optimized further. Indeed, these two constraints are linked by a common concept: the network should grasp a more comprehensive global overview of the entire sequence instead of focusing solely on the local temporal characteristics near the target transition location.

From a clinical perspective, integrating surgical workflow segmentation and anticipation remains challenging. Firstly, surgeries are intricate and can exhibit unexpected variations or accidents not represented in current datasets. Furthermore, even when some rare instances are recorded, their infrequent occurrence can prevent the model from learning these scenarios during training. Second, prolonged efforts and a substantial number of surgeries are necessary to confirm if workflow segmentation effectively lessens surgeons' workload and reduces complications, particularly when additional annotated data might be essential for validation.

Finally, we have only evaluated partial performance with Sacrocolpopexy on its combined phases. Given that our Sacrocolpopexy is a datasets still in devleopment, with further cases being recorded. There may be some unusual phases or long rest times between phases in some videos that needed further clinical consensus toward a complete classification of phases in the procedure. It is necessary to explore a more detailed labeling strategy to

systematically examine new phases and atypical workflows in the enlarged Sacrocolpopexy dataset. In addition, robotic surgery has been integrated into laparoscopic sacrocolpopexy [76], and recently, glue mesh fixation has been adopted in place of sutures to secure the mesh in these procedures [56]. It is worthwhile to examine the behavior and performance of the network in handling these previously unseen techniques without extensive further training or with minimal transfer learning.

References

[1] Adel, B., Badran, A., Elshami, N.E., Salah, A., Fathalla, A., Bekhit, M.: A survey on deep learning architectures in human activities recognition application in sports science, healthcare, and security. In: The International Conference on Innovations in Computing Research, pp. 121–134. Springer (2022)

[2] Ahad, M.A.R., Antar, A.D., Shahid, O.: Vision-based action understanding for assistive healthcare: A short review. In: CVPR Workshops, pp. 1–11 (2019)

[3] Ahmidi, N., Tao, L., Sefati, S., Gao, Y., Lea, C., Haro, B.B., Zappella, L., Khudanpur, S., Vidal, R., Hager, G.D.: A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery. IEEE Transactions on Biomedical Engineering **64**(9), 2025–2041 (2017)

[4] Alavizadeh, H., Alavizadeh, H., Jang-Jaccard, J.: Deep q-learning based reinforcement learning approach for network intrusion detection. Computers **11**(3), 41 (2022)

[5] Badani, K.K., Kaul, S., Menon, M.: Evolution of robotic radical prostatectomy: assessment after 2766 procedures. Cancer **110**(9), 1951–1958 (2007)

[6] Baghdadi, A., Hussein, A.A., Ahmed, Y., Cavuoto, L.A., Guru, K.A.: A computer vision technique for automated assessment of surgical performance using surgeons' console-feed videos. International journal of computer assisted radiology and surgery **14**, 697–707 (2019)

[7] Balicki, M., Kyne, S., Toporek, G., Holthuizen, R., Homan, R., Popovic, A., Burström, G., Persson, O., Edström, E., Elmi-Terander, A., Patriciu, A.: Design and control of an image-guided robot for spine

surgery in a hybrid or. The International Journal of Medical Robotics and Computer Assisted Surgery p. e2108 (2020)

[8] Bartoli, A., Collins, T., Bourdel, N., Canis, M.: Computer assisted minimally invasive surgery: is medical computer vision the answer to improving laparosurgery? Medical hypotheses **79**(6), 858–863 (2012)

[9] Bawa, V.S., Singh, G., KapingA, F., Skarga-Bandurova, I., Oleari, E., Leporini, A., Landolfo, C., Zhao, P., Xiang, X., Luo, G., et al.: The saras endoscopic surgeon action detection (esad) dataset: Challenges and methods. arXiv preprint arXiv:2104.03178 (2021)

[10] Bloice, M.D., Stocker, C., Holzinger, A.: Augmentor: an image augmentation library for machine learning. arXiv preprint arXiv:1708.04680 (2017)

[11] Blum, T., Feußner, H., Navab, N.: Modeling and segmentation of surgical workflow from laparoscopic video. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010: 13th International Conference, Beijing, China, September 20-24, 2010, Proceedings, Part III 13, pp. 400–407. Springer (2010)

[12] Cheikh Youssef, S., Haram, K., Noël, J., Patel, V., Porter, J., Dasgupta, P., Hachach-Haram, N.: Evolution of the digital operating room: the place of video technology in surgery. Langenbeck's archives of surgery **408**(1), 95 (2023)

[13] Chen, W., Feng, J., Lu, J., Zhou, J.: Endo3d: Online workflow analysis for endoscopic surgeries based on 3d cnn and lstm (2018). URL https://link.springer.com/chapter/10.1007/978-3-030-01201-4_12

[14] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using

rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)

[15] Claerhout, F., Roovers, J.P., Lewi, P., Verguts, J., De Ridder, D., Deprest, J.: Implementation of laparoscopic sacrocolpopexy—a single centre's experience. International urogynecology journal **20**(9), 1119–1125 (2009)

[16] Claerhout, F., Verguts, J., Werbrouck, E., Veldman, J., Lewi, P., Deprest, J.: Analysis of the learning process for laparoscopic sacrocolpopexy: identification of challenging steps. International urogynecology journal **25**(9), 1185–1191 (2014)

[17] Czempiel, T., Paschali, M., Keicher, M., Simson, W., Feussner, H., Kim, S.T., Navab, N.: Tecno: Surgical phase recognition with multi-stage temporal convolutional networks. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 343–352. Springer (2020)

[18] Czempiel, T., Paschali, M., Ostler, D., Kim, S.T., Busam, B., Navab, N.: Opera: Attention-regularized transformers for surgical phase recognition. arXiv preprint arXiv:2103.03873 (2021)

[19] Czempiel, T., Sharghi, A., Paschali, M., Navab, N., Mohareri, O.: Surgical workflow recognition: From analysis of challenges to architectural study. In: European Conference on Computer Vision, pp. 556–568. Springer (2022)

[20] Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: Scaling egocentric vision: The epic-kitchens dataset. In: European Conference on Computer Vision (ECCV) (2018)

[21] Das, A., Khan, D.Z., Psychogyios, D., Zhang, Y., Hanrahan, J.G., Vasconcelos, F., Pang, Y., Chen, Z., Wu, J., Zou, X., et al.: Pitvis-2023 challenge: Workflow recognition in videos of endoscopic pituitary surgery. arXiv preprint arXiv:2409.01184 (2024)

[22] De Witt, C.S., Gupta, T., Makoviichuk, D., Makoviychuk, V., Torr, P.H., Sun, M., Whiteson, S.: Is independent learning all you need in the starcraft multi-agent challenge? arXiv preprint arXiv:2011.09533 (2020)

[23] Demir, K.C., Schieber, H., Weise, T., Roth, D., May, M., Maier, A., Yang, S.H.: Deep learning in surgical workflow analysis: a review of phase and step recognition. IEEE Journal of Biomedical and Health Informatics **27**(11), 5405–5417 (2023)

[24] Ding, X., Li, X.: Exploring segment-level semantics for online phase recognition from surgical videos. IEEE Transactions on Medical Imaging **41**(11), 3309–3319 (2022)

[25] Ding, X., Yan, X., Wang, Z., Zhao, W., Zhuang, J., Xu, X., Li, X.: Less is more: Surgical phase recognition from timestamp supervision. IEEE Transactions on Medical Imaging (2023)

[26] Dingemann, J., Kuebler, J., Ure, B.: Laparoscopic and computer-assisted surgery in children. Scandinavian Journal of Surgery **100**(4), 236–242 (2011)

[27] Farha, Y.A., Gall, J.: Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 3575–3584 (2019)

[28] Fırat, O.: Connectionist multi-sequence modelling and applications to multilingual neural machine translation (2017)

[29] Franke, S., Neumuth, T.: Adaptive surgical process models for prediction of surgical work steps from surgical low-level activities. In: 6th Workshop on Modeling and Monitoring of Computer Assisted Interventions (M2CAI) at the 18th International Conference on Medical Image Computing and Computer Assisted Interventions (MICCAI), Munich, Germany (2015)

[30] Funke, I., Rivoir, D., Speidel, S.: Metrics matter in surgical phase recognition. arXiv preprint arXiv:2305.13961 (2023)

[31] Gao, X., Jin, Y., Long, Y., Dou, Q., Heng, P.A.: Trans-svnet: Accurate phase recognition from surgical videos via hybrid embedding aggregation transformer. arXiv preprint arXiv:2103.09712 (2021)

[32] Garrow, C.R., Kowalewski, K.F., Li, L., Wagner, M., Schmidt, M.W., Engelhardt, S., Hashimoto, D.A., Kenngott, H.G., Bodenstedt, S., Speidel, S., et al.: Machine learning for surgical phase recognition: a systematic review. Annals of surgery **273**(4), 684–693 (2021)

[33] Gettman, M.T., Peschel, R., Neururer, R., Bartsch, G.: A comparison of laparoscopic pyeloplasty performed with the davinci robotic system versus standard laparoscopic techniques: initial clinical results. European urology **42**(5), 453–458 (2002)

[34] Gong, D., Lee, J., Jung, D., Kwak, S., Cho, M.: Activity grammars for temporal action segmentation. Advances in Neural Information Processing Systems **36** (2024)

[35] Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)

[36] Guni, A., Varma, P., Zhang, J., Fehervari, M., Ashrafian, H.: Artificial intelligence in surgery: the future is now. European Surgical Research **65**(1), 22–39 (2024)

[37] Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning, pp. 1861–1870. PMLR (2018)

[38] Hao, S., Lee, D.H., Zhao, D.: Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system. Transportation Research Part C: Emerging Technologies **107**, 287–300 (2019)

[39] Haque, A., Milstein, A., Fei-Fei, L.: Illuminating the dark spaces of healthcare with ambient intelligence. Nature **585**(7824), 193–202 (2020)

[40] He, Z., Mottaghi, A., Sharghi, A., Jamal, M.A., Mohareri, O.: An empirical study on activity recognition in long surgical videos. In: Machine Learning for Health, pp. 356–372. PMLR (2022)

[41] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**, 1735–80 (1997). DOI 10.1162/neco.1997.9.8.1735

[42] Huaulmé, A., Sarikaya, D., Le Mut, K., Despinoy, F., Long, Y., Dou, Q., Chng, C.B., Lin, W., Kondo, S., Bravo-Sánchez, L., et al.: Micro-surgical anastomose workflow recognition challenge report. Computer Methods and Programs in Biomedicine **212**, 106452 (2021)

[43] Jin, Y., Dou, Q., Chen, H., Yu, L., Heng, P.A.: EndoRCN: recurrent convolutional networks for recognition of surgical workflow in cholecystectomy procedure video. IEEE Trans. Med. Imaging (2016)

[44] Jin, Y., Dou, Q., Chen, H., Yu, L., Qin, J., Fu, C., Heng, P.: SV-RCNet: Workflow recognition from surgical videos using recurrent convolutional network. IEEE Transactions on Medical Imaging **37**(5), 1114–1126 (2018)

[45] Jin, Y., Long, Y., Chen, C., Zhao, Z., Dou, Q., Heng, P.A.: Temporal memory relation network for workflow recognition from surgical video. IEEE Transactions on Medical Imaging **40**(7), 1911–1923 (2021)

[46] Jin, Y., Long, Y., Gao, X., Stoyanov, D., Dou, Q., Heng, P.A.: Trans-svnet: hybrid embedding aggregation transformer for surgical workflow analysis. International Journal of Computer Assisted Radiology and Surgery **17**(12), 2193–2202 (2022)

[47] Jowett, N., LeBlanc, V., Xeroulis, G., MacRae, H., Dubrowski, A.: Surgical skill acquisition with self-directed practice using computer-based video training. The American Journal of Surgery **193**(2), 237–242 (2007)

[48] Kadkhodamohammadi, A., Sivanesan Uthraraj, N., Giataganas, P., Gras, G., Kerr, K., Luengo, I., Oussedik, S., Stoyanov, D.: Towards video-based surgical workflow understanding in open orthopaedic surgery. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization **9**(3), 286–293 (2021)

[49] Kathamuthu, N.D., Chinnamuthu, A., Iruthayanathan, N., Ramachandran, M., Gandomi, A.H.: Deep q-learning-based neural network with privacy preservation method for secure data transmission in internet of things (iot) healthcare application. Electronics **11**(1), 157 (2022)

[50] Ke, Q., Fritz, M., Schiele, B.: Time-conditioned action anticipation in one shot. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9925–9934 (2019)

[51] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[52] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)

[53] Kuehne, H., Arslan, A., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 780–787 (2014)

[54] Lalys, F.: Automatic recognition of low-level and high-level surgical tasks in the operating room from video images. Ph.D. thesis, Université Rennes 1 (2012)

[55] Lalys, F., Bouget, D., Riffaud, L., Jannin, P.: Automatic knowledge-based recognition of low-level tasks in ophthalmological procedures. International journal of computer assisted radiology and surgery **8**, 39–49 (2013)

[56] Lamblin, G., Dubernard, G., de Saint Hilaire, P., Jacquot, F., Chabert, P., Chene, G., Golfier, F.: Assessment of synthetic glue for mesh attachment in laparoscopic sacrocolpopexy: A prospective multicenter pilot study. Journal of Minimally Invasive Gynecology **24**(1), 41–47 (2017). DOI https://doi.org/10.1016/j.jmig.2016.10.008. URL https://www.sciencedirect.com/science/article/pii/S1553465016311505

[57] Lavallee, S., Troccaz, J., Gaborit, L., Cinquin, P., Benabid, A., Hoffmann, D.: Image guided operating robot: a clinical application in

stereotactic neurosurgery. In: Proceedings 1992 IEEE International Conference on Robotics and Automation (1992)

[58] Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 156–165 (2017)

[59] Lea, C., Hager, G.D., Vidal, R.: 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, pp. 1123–1129. IEEE (2015)

[60] Lea, C., Vidal, R., Hager, G.D.: Learning convolutional action primitives for fine-grained action recognition. In: 2016 IEEE international conference on robotics and automation (ICRA), pp. 1642–1649. IEEE (2016)

[61] Lee, K.M., Ganapathi Subramanian, S., Crowley, M.: Investigation of independent reinforcement learning algorithms in multi-agent environments. Frontiers in Artificial Intelligence **5**, 805823 (2022)

[62] Lee, S.L., Lerotic, M., Vitiello, V., Giannarou, S., Kwok, K.W., Visentini-Scarzanella, M., Yang, G.Z.: From medical images to minimally invasive intervention: Computer assistance for robotic surgery. Computerized Medical Imaging and Graphics **34**(1), 33–45 (2010)

[63] Lindegger, D.J., Wawrzynski, J., Saleh, G.M.: Evolution and applications of artificial intelligence to cataract surgery. Ophthalmology Science **2**(3), 100164 (2022)

[64] Liu, S., Cao, Y., Wang, D., Wu, X., Liu, X., Meng, H.: Any-to-many voice conversion with location-relative sequence-to-sequence modeling.

IEEE/ACM Transactions on Audio, Speech, and Language Processing **29**, 1717–1728 (2021)

[65] Liu, Y., Huo, J., Peng, J., Sparks, R., Dasgupta, P., Granados, A., Ourselin, S.: Skit: a fast key information video transformer for online surgical phase recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 21074–21084 (2023)

[66] Luciano, A.A., Luciano, D.E., Gabbert, J., Seshadri-Kreaden, U.: The impact of robotics on the mode of benign hysterectomy and clinical outcomes. The International Journal of Medical Robotics and Computer Assisted Surgery **12**(1), 114–124 (2016)

[67] Luong, M.T., Le, Q.V., Sutskever, I., Vinyals, O., Kaiser, L.: Multi-task sequence to sequence learning. arXiv preprint arXiv:1511.06114 (2015)

[68] Maktabi, M., Neumuth, T.: Online time and resource management based on surgical workflow time series analysis. International journal of computer assisted radiology and surgery **12**, 325–338 (2017)

[69] Melvin, W.S., Needleman, B.J., Krause, K.R., Schneider, C., Ellison, E.C.: Computer-enhanced vs. standard laparoscopic antireflux surgery. Journal of gastrointestinal surgery **6**(1), 11–16 (2002)

[70] Ming, Y., Cheng, Y., Jing, Y., Liangzhe, L., Pengcheng, Y., Guang, Z., Feng, C.: Surgical skills assessment from robot assisted surgery video data. In: 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), pp. 392–396. IEEE (2021)

[71] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. nature **518**(7540), 529–533 (2015)

[72] Nie, H., Han, X., He, B., Sun, L., Chen, B., Zhang, W., Wu, S., Kong, H.: Deep sequence-to-sequence entity matching for heterogeneous entity resolution. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 629–638 (2019)

[73] Nwoye, C.I., Mutter, D., Marescaux, J., Padoy, N.: Weakly supervised convolutional lstm approach for tool tracking in laparoscopic videos. International journal of computer assisted radiology and surgery **14**(6), 1059–1067 (2019)

[74] Padoy, N., Blum, T., Ahmadi, S.A., Feussner, H., Berger, M.O., Navab, N.: Statistical modeling and recognition of surgical workflow. Medical image analysis **16**(3), 632–641 (2012)

[75] Padoy, N., Blum, T., Feussner, H., Berger, M.O., Navab, N.: On-line recognition of surgical activity for monitoring in the operating room. In: AAAI, pp. 1718–1724 (2008)

[76] Pan, K., Zhang, Y., Wang, Y., Wang, Y., Xu, H.: A systematic review and meta-analysis of conventional laparoscopic sacrocolpopexy versus robot-assisted laparoscopic sacrocolpopexy. International Journal of Gynecology Obstetrics **132**(3), 284–291 (2016). DOI https://doi.org/10.1016/j.ijgo.2015.08.008. URL https://www.sciencedirect.com/science/article/pii/S0020729215007158

[77] Psychogyios, D., Colleoni, E., Van Amsterdam, B., Li, C.Y., Huang, S.Y., Li, Y., Jia, F., Zou, B., Wang, G., Liu, Y., et al.: Sar-rarp50: Segmentation of surgical instrumentation and action recognition on robot-assisted radical prostatectomy challenge. arXiv preprint arXiv:2401.00496 (2023)

[78] Ramesh, S., Dall'Alba, D., Gonzalez, C., Yu, T., Mascagni, P., Mutter, D., Marescaux, J., Fiorini, P., Padoy, N.: Weakly supervised temporal

convolutional networks for fine-grained surgical activity recognition. IEEE Transactions on Medical Imaging **42**(9), 2592–2602 (2023)

[79] Richard, A., Gall, J.: Temporal action detection using a statistical language model. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3131–3140 (2016)

[80] Rivoir, D., Bodenstedt, S., Funke, I., von Bechtolsheim, F., Distler, M., Weitz, J., Speidel, S.: Rethinking anticipation tasks: Uncertainty-aware anticipation of sparse surgical instrument usage for context-aware assistance. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 752–762. Springer (2020)

[81] Sahu, M., Strömsdörfer, R., Mukhopadhyay, A., Zachow, S.: Endo-sim2real: Consistency learning-based domain adaptation for instrument segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 784–794. Springer (2020)

[82] Sayed, S.I.: Understanding human actions: Cognitive assessment and action segmentation using human object interaction. The University of Texas at Arlington (2022)

[83] Schmidt, F.: Generalization in generation: A closer look at exposure bias. CoRR **abs/1910.00292** (2019). URL http://arxiv.org/abs/1910.00292

[84] Schoeffmann, K., Taschwer, M., Sarny, S., Münzer, B., Primus, M.J., Putzgruber, D.: Cataract-101: video dataset of 101 cataract surgeries. In: P. César, M. Zink, N. Murray (eds.) Proceedings of the 9th ACM Multimedia Systems Conference, MMSys 2018, Amsterdam, The Netherlands, June 12-15, 2018, pp.

421–425. ACM (2018). DOI 10.1145/3204949.3208137. URL https://doi.org/10.1145/3204949.3208137

[85] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)

[86] Sener, F., Chatterjee, D., Shelepov, D., He, K., Singhania, D., Wang, R., Yao, A.: Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21096–21106 (2022)

[87] Shah, N.A., Sikder, S., Vedula, S.S., Patel, V.M.: Glsformer: Gated-long, short sequence transformer for step recognition in surgical videos. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 386–396. Springer (2023)

[88] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M.: Deterministic policy gradient algorithms. In: International conference on machine learning, pp. 387–395. Pmlr (2014)

[89] Sinyard, R.D., Rentas, C.M., Gunn, E.G., Etheridge, J.C., Robertson, J.M., Gleason, A., Riley, M.S., Yule, S., Smink, D.S.: Managing a team in the operating room: The science of teamwork and non-technical skills for surgeons. Current problems in surgery **59**(7), 101172 (2022)

[90] Stein, S., McKenna, S.J.: Recognising complex activities with histograms of relative tracklets. Computer Vision and Image Understanding **154**, 82–93 (2017)

[91] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. CoRR **abs/1409.3215** (2014). URL http://arxiv.org/abs/1409.3215

[92] Taylor, R.H., Funda, J., Eldridge, B., Gomory, S., Gruben, K., LaRose, D., Talamini, M., Kavoussi, L., Anderson, J.: A telerobotic assistant for laparoscopic surgery. IEEE Engineering in Medicine and Biology Magazine **14**(3), 279–288 (1995)

[93] Trikha, S., Turnbull, A., Morris, R., Anderson, D., Hossain, P.: The journey to femtosecond laser-assisted cataract surgery: new beginnings or a false dawn? Eye **27**(4), 461–473 (2013)

[94] Twinanda, A.P., Shehata, S., Mutter, D., Marescaux, J., De Mathelin, M., Padoy, N.: Endonet: a deep architecture for recognition tasks on laparoscopic videos. IEEE transactions on medical imaging **36**(1), 86–97 (2016)

[95] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems, pp. 5998–6008 (2017)

[96] Vercauteren, T., Unberath, M., Padoy, N., Navab, N.: Cai4cai: the rise of contextual artificial intelligence in computer-assisted interventions. Proceedings of the IEEE **108**(1), 198–214 (2019)

[97] Ward, J.A., Lukowicz, P., Gellersen, H.W.: Performance metrics for activity recognition. ACM Transactions on Intelligent Systems and Technology (TIST) **2**(1), 1–23 (2011)

[98] Weede, O., Dittrich, F., Wörn, H., Jensen, B., Knoll, A., Wilhelm, D., Kranzfelder, M., Schneider, A., Feussner, H.: Workflow analysis and surgical phase recognition in minimally invasive surgery. In: 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1080–1074. IEEE (2012)

[99] Yeung, S., Russakovsky, O., Jin, N., Andriluka, M., Mori, G., Fei-Fei, L.: Every moment counts: Dense detailed labeling of actions in complex videos. International Journal of Computer Vision (2017)

[100] Yi, F., Yang, Y., Jiang, T.: Not end-to-end: Explore multi-stage architecture for online surgical phase recognition. In: Proceedings of the Asian Conference on Computer Vision, pp. 2613–2628 (2022)

[101] Yuan, K., Holden, M., Gao, S., Lee, W.: Anticipation for surgical workflow through instrument interaction and recognized signals. Medical Image Analysis **82**, 102611 (2022)

[102] Yuan, K., Holden, M., Gao, S., Lee, W.S.: Surgical workflow anticipation using instrument interaction. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part IV 24, pp. 615–625. Springer (2021)

[103] Zhang, B., Ghanem, A., Simes, A., Choi, H., Yoo, A., Min, A.: Swnet: Surgical workflow recognition with deep convolutional network. In: Medical imaging with deep learning, pp. 855–869. PMLR (2021)

[104] Zhang, J., Barbarisi, S., Kadkhodamohammadi, A., Stoyanov, D., Luengo, I.: Self-knowledge distillation for surgical phase recognition. International Journal of Computer Assisted Radiology and Surgery pp. 1–8 (2023)

[105] Zia, A., Guo, L., Zhou, L., Essa, I., Jarc, A.: Novel evaluation of surgical activity recognition models using task-based efficiency metrics. International journal of computer assisted radiology and surgery **14**, 2155–2163 (2019)

[106] Zisimopoulos, O., Flouty, E., Luengo, I., Giataganas, P., Nehme, J., Chow, A., Stoyanov, D.: Deepphase: surgical phase recognition in

cataracts videos. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 265–272. Springer (2018)