2D Skeleton-Based Medical Temporal Segmentation

The effect of limited supervision approaches in 2D skeleton based temporal segmentation of medical procedures

by

G. de Bakker

to obtain the Individual Double Degree of Master of Mechanical Engineering & Robotics at the Delft University of Technology, to be defended publicly on Friday the 31th of Oktober 2025 at 13:00.

Project duration:	March, 2025 - Oktober 2025
Thesis committee:	J.F.P. Kooij
	J.J. van den Dobbelsteen
	B.H.W. Hendriks

TU Delft

Summary

Temporal segmentation of medical procedures holds the potential to improve patient safety, provide decision support to clinicians, and serve as the basis for context-aware robotic assistance systems. However, clinical adoption remains hindered by two key challenges: the scarcity of annotated data and limited generalizability across diverse surgical settings. This thesis therefore explores 2D skeleton-based temporal segmentation as a privacy-preserving and data-efficient alternative to conventional RGB-based methods. Using the CAG-skeleton dataset, which consists of pose sequences extracted from external cardiac angiography (CAG) recordings, the study investigates various model architectures and limited supervision strategies for identifying 14 procedural phases.

A two-stage framework, combining a skeleton-based feature extractor with a temporal model, was adopted. A review and comparison of proven models revealed combinations of PR-GCN or MS-G3D feature extractors with LSTM or TCN temporal models to hold the most promise in the low-data medical domain. After training all combinations on low-data subsets of the CAG-skeleton dataset, it was found that all models outperformed a non-learning baseline model, which always predicts the mean procedure. Between the learning models, clip-wise segmentation accuracy differences held no statistical significance, but LSTM-based models showed a statistically significant superior understanding of sequential order. Considering both sequential metrics and computational efficiency, the PR-GCN + LSTM combination was selected for extensive evaluation, achieving a clip-wise segmentation accuracy of 83.95% when trained on 146 CAG procedures.

To further address the data scarcity challenge, two limited supervision approaches were explored. Transfer learning using the Kinetics-skeleton dataset showed no statistically significant performance gains, suggesting that the knowledge learned from Kinetics-skeleton does not effectively transfer to the surgical domain, and/or that the information transferred is relatively easy for the model to learn from scratch during training. In contrast, pseudo-labeling via class-balanced self-training showed great potential for reducing annotation requirements as it provided consistent improvements to the models' clip-wise segmentation accuracy in the low data regime.

Overall, this study introduces skeleton-based representation as a modality holding large potential for medical temporal segmentation and highlights pseudo-labeling as an effective strategy for reducing annotation requirements.

Contents

1

Introduction

The increasing integration of machine learning into medical settings may transform how we understand, analyze, and support operative workflows.[1] One promising application is the temporal segmentation of medical videos, where every frame of a procedure video is automatically assigned to a specific procedural segment. A procedural segment represents a defined part of the surgical workflow, which can be modeled at different levels of granularity, ranging from coarse phases that describe major stages of the procedure, to intermediate steps and fine-grained actions such as grasping, cutting, or suturing.

To illustrate this concept in a real-world clinical context, Figure 1.1 presents an overview of a cardiac angiography (CAG) procedure recorded with an external camera. The figure demonstrates how a procedure can be divided into distinct phases, each corresponding to a specific stage in the workflow.

If reliable temporal segmentation systems were deployed in clinical practice, they could directly improve patient outcomes and healthcare quality. For example, automatic action recognition could enable real-time detection of complications, ensure adherence to safety protocols, provide immediate decision support to clinicians, and enhance medical training by offering detailed feedback on performance.[1;2;3] Moreover, by providing continuous contextual information about the current procedural phase, temporal segmentation systems can serve as the basis for context-aware robotic assistance systems, enabling robots to anticipate upcoming steps and support surgeons in a timely and task-specific manner.[4;5] In a broader societal context, these systems hold the potential to increase patient safety, shorten operative times, and reduce hospital costs through improved efficiency and workflow optimization.[6;4]

Despite this promise, temporal segmentation methods remain absent in clinical practice. The primary obstacles are twofold: a lack of generalizability across diverse real-world medical environments and the scarcity of annotated medical video data.[7;8] Addressing these limitations is essential if temporal segmentation is to transition from experimental settings to robust, real-world clinical deployment.

1.1. The challenge of generalizability

A core requirement for medical temporal segmentation systems is the ability to generalize across diverse clinical contexts. Models that succeed only in highly specific clinical contexts offer limited practical value. To be clinically viable, generalization must be achieved on multiple levels:

- *Patients:* Models must work for individuals of different body types, ages, genders, and skin tones. A system that only functions on data from fit or light-skinned patients risks introducing inequities in healthcare delivery.
- *Surgeons:* Surgeons may follow different habits when executing the same procedure. For instance, one surgeon may consistently perform a specific movement before transferring to a new phase, while another may not. Models that overfit to these individual patterns risk failing in broader clinical use.
- *Operating room environments:* Hospitals differ in the layout of operating rooms, the positioning of cameras, and the equipment used. Variations in patient attire, surgical instruments, or imaging

systems can further challenge model robustness. Even subtle differences, such as wall colors, lighting conditions, or clothing styles, can degrade the performance of vision-based models.

- *Geographical and regulatory contexts:* Medical practices differ across countries due to cultural or regulatory variations. For instance, adherence to safety checklists may be stricter in some regions than others, requiring models to adapt to localized practices.

Failure to generalize across these levels can compromise both the safety and effectiveness of temporal segmentation systems. Ensuring robust generalization is therefore essential for their reliable and ethical deployment in clinical practice.

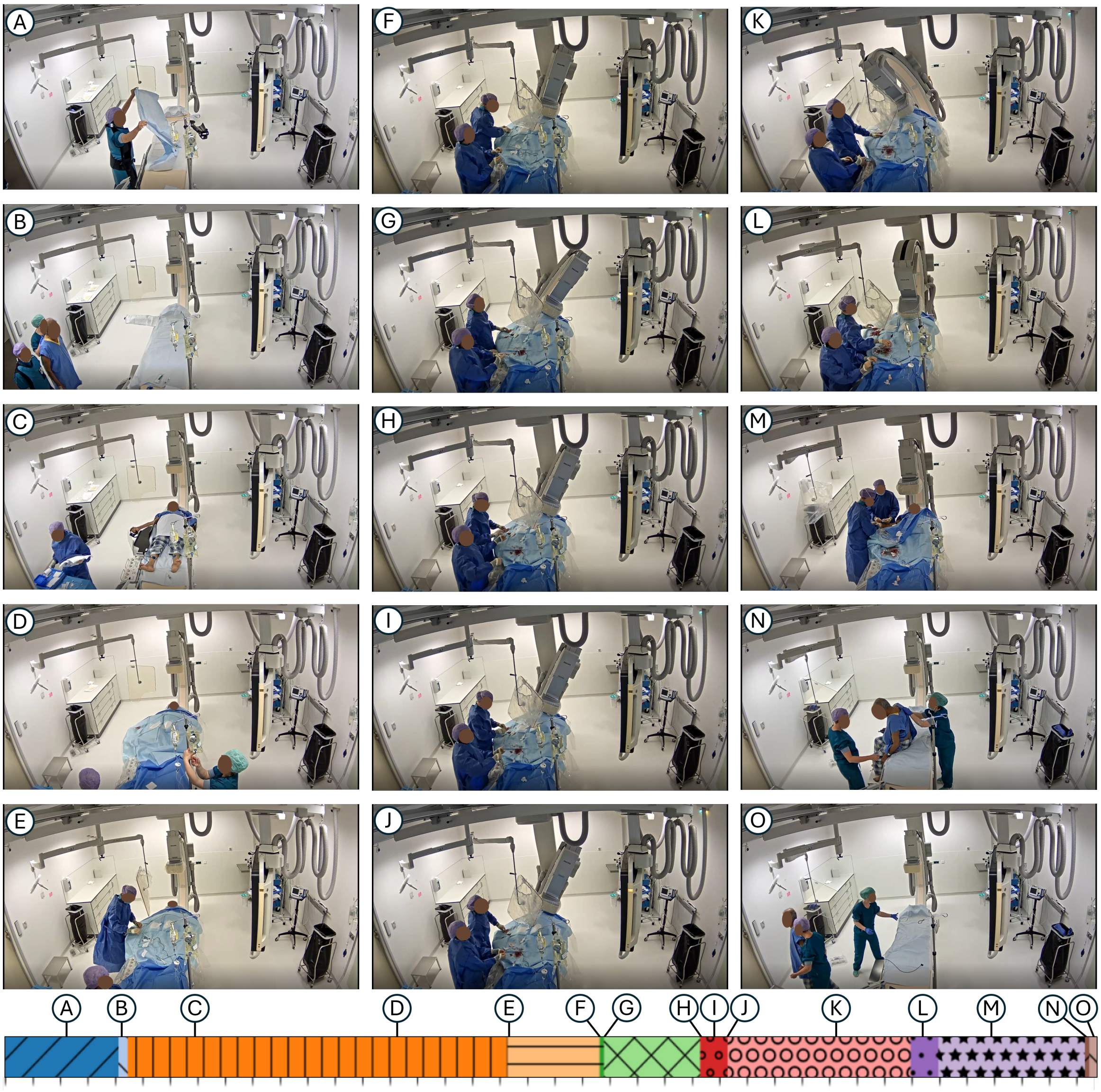


Figure 1.1: Overview of procedural phases in a cardiac angiography procedure recorded with an external camera. Each image is an anonymized frame of the video recording corresponding to a distinct phase in the workflow: A) Preparation before patient entry, B) Patient entry, C–D) Patient on table, E) First contact with cardiologist, F) First catheter insertion, G) First X-ray after catheter insertion, H–I) Catheter switch: H) first catheter removal, I) second catheter insertion, J–K) First (J) and later (K) X-rays after second catheter insertion, L) Second catheter removal, M) Wound closure, N) Patient off table, and O) Patient exit and start of cleaning. The colored bar below illustrates the temporal sequence of the phases and the position of each frame within the overall procedure. The vertical lines beneath the bar indicate one-minute intervals.

1.2. The challenge of limited annotated data

Alongside generalizability, the limited availability of annotated surgical data remains a critical bottleneck. Annotating medical videos requires expert-level clinical knowledge and is both time-intensive and costly.[9] In addition, privacy regulations restrict data sharing, resulting in small datasets collected from only a few hospitals, which limits model performance and hinders robust model training and evaluation.[10] Consequently, temporal segmentation models are prone to overfitting to specific institutions and often fail to generalize in broader clinical settings.

1.3. Data modalities for temporal segmentation

The choice of data modality is a key factor in the generalizability of temporal segmentation systems. Most existing approaches rely on RGB video, which captures rich visual detail but is highly sensitive to variations in background, lighting, viewpoint, and clothing.[11] In low-data environments, models trained on a single operating room often overfit to specific visual features, such as wall colors or surgical gowns, resulting in poor generalization to new clinical settings. Similarly, methods based on object or instrument detection are vulnerable to differences in equipment brands or visual appearances across hospitals.

Other explored modalities include binary instrument usage signals[12], RFID tags and sensors[13], surgical staff or instrument tracking[14;15], eye-gaze tracking[16], and contextual signals such as table inclination and lighting state[17]. While these approaches can provide useful information, their real-world viability is limited, as they often require workflow modifications, additional equipment, or extensive cooperation with industry stakeholders to access machine data.

In contrast, this work explores the skeleton modality, which, despite its advantages, has not yet been explored for medical temporal segmentation. The skeleton modality represents human motion as a sequence of body joint positions. Advances in depth sensors[18] and human pose estimation algorithms[19] have made it feasible to extract skeleton sequence data reliably, even from standard RGB video inputs. Skeleton representations abstract away appearance, background, and lighting conditions, thereby reducing noise and capturing the essence of movement in a way that is both compact and largely invariant to environmental conditions.[20;21] Three-dimensional skeletons are particularly useful as they offer robustness to viewpoint changes, while both 2D and 3D skeletons mitigate privacy concerns since identifiable visual information is discarded. This makes skeleton-based approaches a strong candidate for robust and ethically responsible temporal segmentation of medical procedures.

However, skeleton-based data may not solve all challenges. While skeleton representations abstract away appearance, background, and lighting, which reduces noise and improves generalizability, they also discard information that could be informative for temporal segmentation, potentially limiting model performance. Furthermore, surgeon-specific patterns, hospital room layout, and geographical differences remain sources of variability that cannot be fully abstracted away by motion alone. Ensuring robustness against these factors requires access to sufficiently large and diverse datasets.

Importantly, because skeleton representations do not contain identifiable patient information, they enable the creation of larger, more diverse, and international datasets. This property could improve generalizability across all levels.

1.4. Limited supervision learning

A promising solution to the limited annotated data challenge lies in limited supervision learning, where learning strategies reduce dependence on complete annotations while leveraging unlabeled or coarsely labeled data. Approaches range from weak supervision, where only coarse information, such as action order, is available, to semi-supervised methods that combine small labeled sets with larger unlabeled ones, and self-supervised approaches that learn meaningful representations by solving pretext tasks (e.g., predicting the order of shuffled frames or reconstructing masked inputs). The features learned in this way can then be transferred to temporal segmentation, improving performance even when annotated data is scarce.[22;23]

This work specifically investigates two limited supervision strategies. The first is transfer learning, which leverages large-scale non-medical datasets to compensate for the limited availability of medical data.

The core idea is that abundant non-medical motion data can be used during pretraining to capture the generic spatio-temporal structure of skeleton sequences. Subsequently, the pretrained model is fine-tuned on the surgical phase segmentation task. This approach maximizes the efficient use of scarce surgical annotations while reducing the overall annotation burden.

The second strategy is pseudo-labeling, a semi-supervised technique designed to reduce the number of procedures that require costly frame-level annotations. In this approach, a model is first trained on the small annotated subset and then used to assign provisional labels (pseudo-labels) to the unlabeled procedures. These pseudo-labels are subsequently used for further training, which enables the model to generalize beyond the limited labeled set. However, incorrect pseudo-labels can introduce noise and degrade performance. To address this, confidence thresholds or ensemble methods are typically used to filter unreliable predictions[24;25].

1.5. Scope of this study

This study addresses two key challenges that limit the clinical adoption of temporal segmentation in the medical domain: the scarcity of annotated data and the difficulty of achieving robust generalization across different surgical settings. To tackle these challenges, it investigates two limited supervision techniques that are applied to data based on the human skeleton modality: transfer learning through pretraining on a large-scale non-medical dataset, and pseudo-labeling using unlabeled, domain-specific medical data. Notably, the skeleton modality has not previously been applied in the medical context, allowing this study to provide novel insights into its potential effectiveness and limitations for temporal segmentation of surgical procedures.

In addition to evaluating the effect of limited supervision techniques, this study also compares different learning architectures for skeleton-based temporal segmentation. Specifically, multiple combinations of feature extractor and temporal modeling architectures are explored to assess their respective performance and data efficiency. This comparison may provide insight into which model architecture characteristics are best suited for skeleton-based medical temporal segmentation under limited data conditions.

The central research question guiding this work is:

> *What is the effect of introducing limited supervision techniques, specifically pretraining and pseudo-labeling, on the segmentation accuracy of 2D skeleton-based temporal phase segmentation of medical procedures?*

To answer this question, the study also investigates the following sub-question:

> *Which model architectures, specifically combinations of feature extractors and temporal models, show great promise for high segmentation accuracy in skeleton-based medical temporal segmentation?*

By addressing these questions, this study contributes to the development of temporal segmentation systems that are both robust and capable of generalizing across diverse clinical environments under low-data conditions.

1.6. Report structure

The remainder of this thesis is structured as follows.

Chapter 2 introduces the background and related work. It begins by presenting the 2D human skeleton dataset (CAG-skeleton) used to evaluate the models and learning strategies in this study (Section 2.1). A second dataset is required for pretraining, but since no comparable medical dataset exists, a dataset from the human action recognition domain is selected in Section 2.2.

Next, this chapter reviews prior work on suitable architectural design choices in Section 2.3. Skeleton-based feature extractor models trained on the transfer learning dataset are reviewed and compared in Subsection 2.3.1. Models offering the best trade-off between reported performance and parameter efficiency, and for which pretrained weights are publicly available, are selected for further use. To overcome differences between the human action recognition and medical temporal phase segmentation domains, a second learning model is required that models longer-range temporal relations. These temporal models are reviewed and compared in Subsection 2.3.2, and the most suitable ones, based on ability to capture longer-range temporal relations and data efficiency are selected. Various limited supervision approaches are introduced in Subsection 2.3.3, and the selected transfer learning and pseudo-labeling methods are introduced in more detail. The chapter concludes with an overview of the contributions of this work (Section 2.4).

Chapter 3, methodology, first introduces a non-learning baseline model which will be used as a reference point for evaluating the effectiveness of the proposed learning methods. The proposed transfer learning approach requires alignment between the CAG-skeleton dataset and the Kinetics-skeleton dataset. Preprocessing steps to align the CAG-skeleton dataset in terms of structure, temporal resolution, and joint configuration are described in Section 3.2. The training pipeline, optimized hyperparameters, and learning methods of the feature extractor, temporal model, transfer learning, and pseudo-labeling approaches are presented in Sections 3.3, 3.4, 3.6, and 3.7, respectively.

To reduce computational requirements, the limited supervision methods are evaluated using only the best-performing combination of feature extractor and temporal model, as identified in Section 3.5. The effectiveness of the limited supervision methods is determined using the paired t-test, introduced in Section 3.8. Finally, the performance metrics used to evaluate and compare model performance are presented in Section 3.9.

Chapter 4 presents the experiments and results. To analyze performance under varying amounts of data, 39 distinct training subsets were created, with their division and training, validation, and test splits described in Section 4.1. Quantitative and qualitative findings are reported, and compared against the non-learning baseline model. The effectiveness of the studied model architectures, and limited supervision strategies are analyzed in Sections 4.2 and 4.3 respectively.

Chapter 5 discusses the findings in the context of the challenges identified in the introduction. It reflects on the effect of limited supervision for medical phase segmentation, highlights remaining challenges, and proposes directions for future work. The chapter concludes with the overall conclusions of the study.

2
Background & Related Work

This chapter provides the background and related work that form the foundation for the experiments conducted in this study. Section 2.1 introduces CAG-skeleton, the 2D human skeleton dataset used to evaluate the models and learning strategies in this study. Next, Section 2.2 gives the requirements for the dataset which will be used for the transfer learning limited supervision strategy. It compares various datasets from the human action recognition domain and selects the best one based on size and similarity to the CAG-skeleton dataset.

The learning methods explored in this study are introduced in Section 2.3. Feature extractor models trained on the transfer learning dataset are compared in Subsection 2.3.1. Models offering the best trade-off between performance and parameter efficiency, and for which pretrained weights are publicly available, are selected for further use. To overcome differences between the human action recognition and medical temporal phase segmentation domains, a second learning model is required that models longer-range temporal relations. These temporal models are investigated in Subsection 2.3.2, and the most suitable ones, based on ability to capture longer-range temporal relations and data efficiency are selected. Various limited supervision approaches are introduced in Subsection 2.3.3, and the selected transfer learning and pseudo-labeling methods are presented in more detail. The chapter concludes with an overview of the contributions of this work (Section 2.4).

2.1. CAG-skeleton dataset

To compare the performance of models and learning strategies analysed in this study, this work uses the CAG-skeleton dataset. The CAG-skeleton dataset is introduced by Butler et al.[26], and contains recordings of 290 coronary angiography (CAG) procedures performed at the Reinier de Graaf Gasthuis hospital in Delft, the Netherlands. It provides 2D pose sequences derived from video recordings of actual procedures, and includes frame-level phase annotations for a subset of cases. Data collection was approved by the Medical Ethics Committee Leiden The Hague Delft (protocol number Z19.057, dated 30-10-2019), and informed consent was obtained from all participating patients and clinical staff.

Each procedure was recorded using four Axis M1125 cameras, capturing different viewpoints at a resolution of 1920×1088 pixels and a frame rate of 25 frames per second. A cardiologist, scrub nurse, up to two lab assistants, and the patient were present during each procedure.

2.1.1. Pose estimation

To acquire ground truth pose annotations, ten procedures were selected in collaboration with local clinical experts to represent team diversity and capture rare procedural deviations. For each procedure, 51 frames were uniformly sampled across a 30-second interval and annotated with 2D keypoints across all four viewpoints (see figure 2.1[27]), resulting in 2040 annotated frames in total. Annotations excluded fully occluded individuals and keypoint reflections (e.g., those appearing on monitors), but included visible staff in adjacent rooms, such as the control room or hallway.

Among the four viewpoints, the south (S) wall camera was identified as optimal in terms of viewpoint quality and occlusion levels, and was therefore used for further analysis. Full-length recordings from this viewpoint were processed with PoseBYTE[26], producing 290 skeleton sequences in COCO format (Figure 2.2). Each keypoint is defined by an (x, y) coordinate and a confidence score.

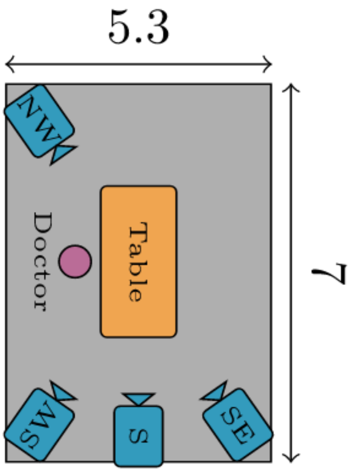


Figure 2.1: Cathlab dimensions (in meters) and camera viewpoints.[27]

2.1.2. Phase segmentation

All 290 procedures were segmented into 14 predefined clinical phases. Annotations were performed by a medical student under the supervision of an expert. Figure 1.1 presents anonymized frames from a representative procedure alongside their corresponding phase labels. Table 2.1 provides an overview of the phase definitions, as well as the mean and interquartile range (IQR) of their durations and the number of procedures in which each phase occurred. It should be noted that the dataset exhibits substantial class imbalance in phase durations. In addition, some phases, such as *Additional catheter change (13)* and *First X-ray acquisition after additional catheter change (14)*, do not occur in every procedure. Furthermore, a subset of procedure recordings start late or end early, resulting in missing phases.

Several phases also feature (nearly) identical activity patterns, complicating classification based solely on observable motions. For example, *First X-ray acquisition (5)* and *First X-ray acquisition after catheter switch (7)* involve (nearly) identical movements.

Improper data storage resulted in temporal misalignment between the phase labels, video recordings, and pose data. Realignment was performed manually by the authors of this paper. For each procedure, the time interval between two visually distinctive phases was used to resynchronize videos and annotations. Pose sequences were separately aligned by matching the interval from the first staff entry to the last staff exit with the corresponding video segment. All realignments were manually verified by overlaying pose data and annotations onto the video recordings.

2.1.3. Final dataset

The resulting CAG-skeleton dataset after realignment contains:

- 189 procedures with both pose data and 14 annotated clinical phases
- 101 procedures with pose data only (no phase labels)
- 2D keypoints with confidence scores in COCO format (see figure 2.2[28])
- Up to five persons per frame
- Frame rate of 25 frames per second

Table 2.1: Clinical phase definitions in the CAG-skeleton dataset with mean duration [min], interquartile range [min], and the number of procedures in which each phase occurs.

Clinical Phase	Description	Mean duration	Interquartile range	Procedures containing phase
0	Start Preparation	10.02	3.88 - 13.29	250
1	Patient entry	0.60	0.3 - 0.55	254
2	Patient transfer to table	10.66	8.17 - 12.62	271
3	First contact cardiologist with patient	6.29	3.33 - 7.27	271
4	Endovascular access (catheter insertion)	0.22	0.07 - 0.23	271
5	First X-ray acquisition	5.10	2.35 - 5.93	290
6	Removal of right catheter & insertion of left catheter	0.99	0.72 - 1.05	290
7	First X-ray acquisition after catheter switch	5.62	3.08 - 6.79	290
8	Removal of second (left) catheter	1.76	1.19 - 2.05	289
9	Wound closure	3.44	2.43 - 4.1	287
10	Patient off table	0.66	0.32 - 0.8	282
11	Patient exit & cleaning	1.47	0.42 - 1.92	278
12	Additional catheter change	6.67	4.49 - 7.55	77
13	First X-ray after additional catheter change	5.21	3.33 - 6.33	78
	All procedures	47.03	34.93 - 55.77	290

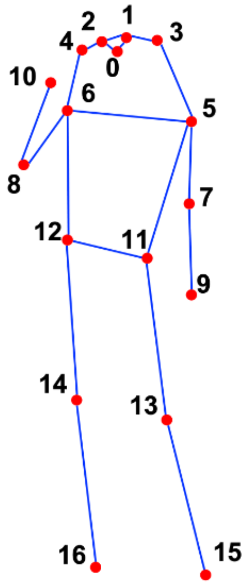


Figure 2.2: COCO skeleton format[28]. Keypoints locations include: nose, left eye, right eye, left ear, right ear, left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle, right ankle.

2.2. Transfer learning dataset

Since one of the limited supervision techniques explored in this work is transfer learning, a second dataset is required for pretraining. Ideally, this dataset should resemble the CAG-skeleton dataset in terms of skeleton representation, activity complexity, and viewpoint characteristics, thereby ensuring that the pretrained model learns features relevant to the downstream task of phase segmentation in CAG procedures.

Moreover, the pretraining dataset should be sufficiently large. A larger dataset enables the model to acquire a more diverse and representative set of motion patterns, which in turn enhances its generalization capability.

As no other skeleton-based medical temporal segmentation datasets are available to the author, a dataset from a different research domain must be used. As research on skeleton sequences has predominantly been explored within the field of human action recognition (HAR), this domain is selected.

Within HAR, numerous datasets have been published, each developed for different application domains. These datasets vary substantially in size, recording method, skeleton structure, and actions analyzed, depending on their intended use cases. For example, Kishore et al.'s *Indian Sign Language dataset*[29] was designed for sign language recognition and includes 18 keypoints per hand to capture fine-grained finger movements. Yun et al.'s *Two-person Interaction Detection dataset*[30], on the other hand, focuses specifically on human–human interactions and excludes single-person activities. Jang et al.'s *ETRI-Activity3D dataset*[31] records daily living activities of elderly individuals to support the development of future human-care robots. Similarly, Martin et al.'s *Drive&Act dataset*[32] targets driver monitoring, with a focus on detecting secondary in-vehicle activities.

A quantitative comparison and summary of the reviewed datasets is presented in Table 2.2. The following subsections discuss data acquisition techniques (Section 2.2.1), skeleton structures (Section 2.2.2), and the final selection of the dataset which will be used for transfer learning (Section 2.2.3).

Table 2.2: Datasets used within the skeleton-based human action recognition domain. A dash ("–") indicates that the corresponding information was either not reported by the dataset creators or could not be found despite an extensive search by the author of this thesis.

Dataset	Year	Modality	Dimensionality	# Joints	# Sequences	# Actions	Multi-person
HDM05[33]	2007	MOCAP platform	3D	31	1457	70	✗
DailyActivity3D[34]	2012	MS Kinect V1	3D	20	320	16	✗
HOJ3D[35]	2012	MS Kinect V1	3D	20	200	10	✗
Two-person Interaction Detection[30]	2012	MS Kinect V1	3D	15	300	8	✓
UCFKinect[36]	2013	MS Kinect V1	3D	15	1280	16	✗
MSR Action 3D[37]	2013	MS Kinetics	3D	20	600	20	✗
J-HMDB[38]	2013	RGB camera with Amazon Mechanical Turk	2D	13	928	21	✗
IAS-lab action[39]	2013	NITE middleware	3D	-	540	15	✗
Berkeley MHAD[40]	2013	Impulse MOCAP system	3D	21	660	11	✗
CMU[41]	2014	MOCAP platform	3D	22	44	9	✓
Northwestern UCLA Multiview 3D[42]	2014	MS Kinect V1	3D	21	100	10	✗
UTD-MHAD[43]	2015	MS Kinect V1	3D	20	861	27	✗
MV-TJU[44]	2015	MS Kinect V1	3D	20	7040	22	✗
NTU RGB+D[45]	2016	MS Kinect V2	3D	25	56,880	60	✓
PKU-MMD[46]	2017	MS Kinect V2	3D	25	21,545	51	✓
RGB-D Varying-view[47]	2018	MS Kinect V2	3D	25	25,600	40	✗
Kinetics-skeleton[48;49]	2018	RGB camera with OpenPose[50]	2D+c	18	260,230	400	✓
Indian Sign Language[29]	2018	Vicon MOCAP system	3D	57	2500	500	✗
DHP19[51]	2019	Vicon MOCAP system	3D	13	5610	33	✗
MMAct[52]	2019	-	-	-	36764	37	✗
Drive&Act[32]	2019	Near-infrared cameras with OpenPose[50], OpenFace[53] & triangulation	3D	13	-	83	✗
ETRI-Activity3D[31]	2020	MS Kinect V2	3D	25	112,620	55	✗
EV-Action[54]	2020	MS Kinect V2 & Vicon-T40	3D	39	7000	20	✗
NTU RGB+D 120[55]	2020	MS Kinect V2	3D	25	114,480	120	✓
IKEA ASM[56]	2021	RGB video with 3D VIBE[57]	3D	17	16,764	31	✗

Dataset	Year	Modality	Dim	# J	# S	# A	MP
KLHA3D102[58]	2021	Vicon MOCAP system	3D	39	10,200	102	✗
UAV-Human[59]	2021	Azure Kinect DK with RMPE[60]	2D	17	67,428	155	✓
KLYOGA3D[58]	2021	Vicon MOCAP system	3D	39	2,100	42	✗

2.2.1. Data acquisition

The datasets summarized in Table 2.2 differ in how skeleton data is obtained. Broadly, acquisition relies on RGB-D sensors, motion capture systems, or pose estimation from RGB or near-infrared video.

RGB-D sensors, such as the Microsoft Kinect, capture synchronized RGB and depth images. Combined with toolkits such as OpenNI[36] or the Kinect SDK[43], these devices enable real-time tracking of human skeleton joints and their approximate 3D positions.

Marker-based motion capture systems, such as the Vicon platform, provide more precise 3D skeleton tracking by using optical cameras to record reflective markers placed on the body[54].

In contrast, the CAG-skeleton dataset relies on a monocular RGB setup, where 2D skeletons are estimated directly from video recordings. While several other datasets provide full 3D skeleton representations, this does not make them unsuitable for transfer learning, as 3D sequences can be projected into 2D space.

2.2.2. Skeletons

A critical factor in selecting an suitable dataset for transfer learning is the structural compatibility of the skeleton representations. For transfer learning to be effective, the pretrained model should have already learned meaningful spatial and temporal relationships between keypoints that are transferable to the target data. Such transfer is more likely when the source and target datasets employ comparable skeleton structures.

The datasets reviewed vary considerably in this regard, differing in both the number of keypoints and their anatomical placement. Figure 2.3[58] illustrates several examples of these variations.

Among the analyzed datasets, only three exhibit skeleton structures that closely resemble that of the CAG-skeleton dataset: the IKEA ASM dataset[56], the UAV-Human dataset[59], and the Kinetics-skeleton dataset[48;49]. Of these, the Kinetics-skeleton dataset is structurally almost identical, differing only in the addition of a single neck keypoint.

2.2.3. Final dataset selection

The choice of pretraining dataset was guided by four criteria:

1. Structural similarity of the skeleton representation
2. Viewpoint similarity (for 2D skeletons)
3. Dataset size
4. Ability to handle multi-person scenes

Applying the first criterion narrowed the candidates to three datasets: Kinetics-skeleton, IKEA ASM, and UAV-Human. The UAV-Human dataset was excluded because its 2D drone-based recordings introduce steep top-down viewpoints that differ substantially from those in the CAG-skeleton dataset.

Between the remaining two options, the Kinetics-skeleton dataset offers clear advantages: it is considerably larger than the IKEA ASM dataset and includes multi-person interactions. Consequently, the Kinetics-skeleton dataset was selected as the most suitable source for pretraining.

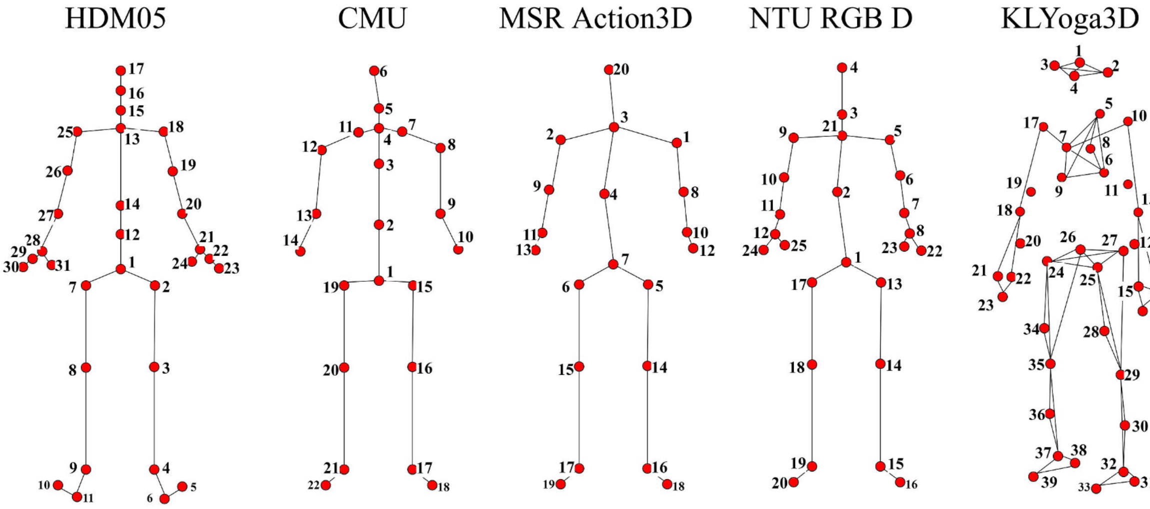


Figure 2.3: Subset of skeletons structures used throughout the datasets.[58]

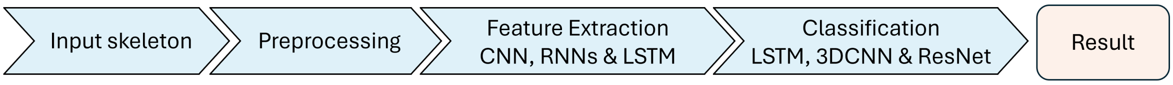

Input skeleton → Preprocessing → Feature Extraction CNN, RNNs & LSTM → Classification LSTM, 3DCNN & ResNet → Result

Figure 2.4: Framework of skeleton-based action recognition approaches. Image adapted from Shin et al. (2024)[21].

2.3. Learning methods

Once skeleton data is collected, the recognition or segmentation process begins. The general pipeline of skeleton-based methods, shown in Figure 2.4, typically consists of preprocessing, feature extraction, and classification[21].

Early approaches relied on handcrafted features such as joint angles, velocities, and relative distances, followed by classical sequence methods such as Hidden Markov Models (HMMs). While interpretable, these approaches lacked the representational power to capture complex spatiotemporal dynamics in real-world data[21].

With the rise of deep learning, Recurrent Neural Networks (RNNs) and their variants such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) became dominant for modeling temporal dependencies in skeleton sequences. Although effective for short-term actions, they suffer from vanishing gradients and difficulties in modeling long-range dependencies[61]. Temporal Convolutional Networks (TCNs) emerged as an alternative, offering stable receptive fields, parallelizability, and improved performance in longer sequences[62].

A major breakthrough came with Graph Convolutional Networks (GCNs), which directly exploit the graph structure of skeletons. The Spatial-Temporal GCN (ST-GCN)[49] extends standard convolution to spatiotemporal skeleton graphs, enabling simultaneous modeling of spatial relationships (joints within a frame) and temporal dynamics (joints across frames). Subsequent research found several limitations of the original ST-GCN framework:

1. The graph structure in ST-GCN is predefined based on the physical structure of the human skeleton. While this captures anatomical relations effectively, it ignores semantic dependencies between joints that are not directly connected. For instance, in actions such as ”touching the face”, the interaction between the hand and the head, although not anatomically linked, is crucial for accurate recognition.[63;20].
2. While manually predefined graph structures are intuitive, they lack flexibility and often require dataset-specific customizations. This increases the manual effort required to adapt the model to different skeleton formats/datasets.

3. As ST-GCN extract spatial features, and models temporal dynamics in two distinct steps, it struggles to capture joint co-occurrences across space and time simultaneously. Identifying actions like ”running”, where the model needs to focus on the swing of the hands and the movements of the legs simultaneously, becomes challenging as the co-occurring movements of the arms and legs parts across both space and time cannot be identified well.[64]
4. ST-GCN architectures tend to be highly complex and over-parameterized, resulting in inefficient model training and inference.[65;66] Recent work by Xie et al.[65] demonstrated that sparse variants of ST-GCN, using up to 95% fewer parameters, can achieve comparable performance compared to their dense counterparts (degradation of less than 1% in top-1 accuracy across four benchmark datasets, including NTU-RGB+D 60/120, Kinetics-400, and FineGYM).

To address these challenges, extensions of ST-GCN have been proposed that incorporate learnable adjacency matrices (ST-GCN++[67], DG-STGCN[68]), reduce model complexity (PR-GCN[69]), and introduce the attention mechanisms. Parallel to these developments, transformer-based architectures have been introduced, leveraging self-attention to model long-range dependencies. For example, Plizzari et al.[70] proposed the Spatio-Temporal Transformer network (ST-TR), which applies spatial self-attention to capture correlations between joints within a frame and temporal self-attention to capture dynamics across frames.

While these models achieve strong performance in the field of human action recognition (HAR), it is important to distinguish HAR from temporal phase segmentation. In HAR, the task is to assign a single action label (e.g., walking, sitting, throwing) to a short skeleton sequence, typically lasting only a few seconds. In contrast, temporal phase segmentation extends this to much longer sequences and requires identifying multiple consecutive actions or phases with frame- or segment-level precision. This introduces additional challenges, including modeling long-range temporal dependencies, detecting action boundaries, and handling large variations in action duration.

Therefore, in this study, a standard HAR model is employed as a feature extractor to capture short-term spatio-temporal dynamics, and is combined with a dedicated temporal model designed to efficiently learn longer-range temporal dependencies.

2.3.1. Feature extractor selection

The choice of feature extractors was guided by two factors: (1) the availability of a pretrained model on the Kinetics-skeleton dataset, and (2) practical considerations of model scale and trainability. Pretraining on Kinetics-skeleton is essential, as it is the only dataset structurally compatible with CAG-skeleton and of sufficient size to support effective transfer learning (see Section 2.2). Training models from scratch and performing extensive hyperparameter searches on the Kinetics-skeleton dataset would be computationally infeasible given the resources and timeframe of this project. Consequently, only models with publicly available pretrained weights on the Kinetics-skeleton dataset were considered.

Although state-of-the-art methods such as ProtoGCN[71] or two-stream architectures (incorporating both joint and bone inputs) achieve strong performance on Kinetics-skeleton, their large parameter counts make them impractical for this study. Larger models not only require substantially more computational resources but also tend to overfit rapidly on the comparatively small datasets common in the medical domain. For this reason, parameter efficiency was treated as an important consideration: models with moderate size are more feasible to train and generally require less target data for training and effective fine-tuning. This consideration also motivated a focus on joint-only configurations, as the bone modality, while offering marginal gains in some benchmarks, approximately doubles model size without introducing fundamentally new information.

After filtering and ranking the candidates identified in an extensive literature review (Table 2.3), two models arose: MS-G3D[72] and PR-GCN[69]. MS-G3D demonstrated stronger performance on Kinetics-skeleton, while PR-GCN offered a highly compact design with just 580 thousand parameters. These complementary characteristics motivated their joint selection as feature extractors for subsequent experiments.

Table 2.3: Overview of evaluated models. ✓ in *Bone* indicates usage of both joint and bone inputs, *Top-1*/*Top-5* are classification accuracies (%). *Params* marked with † are sourced from a alternative studies. *Params* marked with ‡ are manually sourced using pretrained models. ✓ in *Code* and *Pretrained* show public availability and are hyperlinks.

Module	Year	Bone	Top-1	Top-5	Params (M)	Code	Pretrained
ST-GCN[49]	2018		30.7	52.8	3.1†	✓	
AR-GCN[73]	2019		33.5	55.1			
SLnL-rFA[74]	2019		36.6	59.1			
STGR-GCN[75]	2019		33.6	56.1			
AS-GCN[76]	2019		34.8	56.5	6.9†	✓	
DGNN[77]	2019	✓	36.9	59.6	8.2†	✓	
2s-AGCN[78]	2019	✓	36.1	58.7	7.1†	✓	
Ours-Conv[79]	2019		30.8	52.6			
Ours-Conv-Chiral[79]	2019		30.9	53.0			
MS-G3D[72]	2020		35.8	58.6	3.14	✓	✓
	2020	✓	38.0	60.9	6.29	✓	✓
GCN-NAS[80]	2020		35.5	57.9		✓	
	2020	✓	37.1	60.1		✓	
MS-AAGCN[81]	2020		36.0	58.4		✓	
	2020	✓	37.8	61.0	3.8†	✓	
Dynamic GCN[82]	2020	✓	37.9	61.3	3.6		
A-CA-GCN[83]	2020		34.1	56.6	5.38		
PeGCN[84]	2020		34.8	57.2		✓	
SS-GCN[85]	2021		35.2	57.5	6.9		
DualHead-Net[86]	2021		36.6	59.5	3.0	✓	
	2021	✓	38.3	61.1		✓	
PR-GCN[69]	2020		33.6	56.1	0.58	✓	✓
2s-AAGCN+TEM[87]	2021	✓	38.6	61.6			
S-TR[70]	2020		32.4	55.3	3.19‡	✓	✓
	2020	✓	35.4	57.9	12.35‡	✓	✓
T-TR[70]	2020		32.4	55.2	1.76	✓	✓
	2020	✓	33.1	55.9	6.58‡	✓	✓
ST-TR[70]	2020		34.5	57.6	4.95‡	✓	✓
	2020	✓	37.0	59.7	18.93‡	✓	✓
T-TR-agcn[70]	2020		34.4	57.1	2.22‡	✓	
	2020	✓	34.7	56.4	8.06‡	✓	✓
ST-TR-agcn[70]	2020		36.1	58.7	5.41‡	✓	
	2020	✓	38.0	60.5	12.41‡	✓	✓
AAM-GCN[88]	2021	✓	37.5	60.5			
PoseConv3D[89]	2022		46.0		2.0	✓	
	2022	✓	47.7		2.0	✓	
STF[90]	2022		38.2				
	2022	✓	39.9				
Sybio-GNN[91]	2022	✓	37.2	58.1	14.85		
UNFGEF[92]	2022	✓	37.6	60.5			
SKP[93]	2023	✓	43.1				
LKA-GCN[94]	2023		37.8	60.9	3.47		
	2023	✓	37.8	60.9	3.78		
2s-GATCN[95]	2023	✓	36.7	59.8			
HAR-ViT[96]	2023		38.1	60.9			
ProtoGCN[71]	2024	✓	51.9	75.6	25.85‡	✓	✓
DS-GCN[97]	2024	✓	50.6			✓	
Gnet[98]	2025	✓	38.2		7.36		
Tnet[98]	2025	✓	30.7		4.33		
SA-TDGFormer[98]	2025	✓	39.0		7.36		
LMSTGCN[99]	2025	✓	37.7	60.5	5.4		

2.3.2. Temporal model selection

The kinetics-skeleton dataset consists of short segments with a temporal length of ten seconds. Consequently, the pretrained MS-G3D and PR-GCN feature extractors are limited to modeling short-range temporal dependencies. While this may be sufficient for recognizing distinct actions, it is inadequate for medical phase segmentation, where different phases may exhibit (nearly) identical short-term motion patterns (e.g. *First X-ray acquisition (F)* and *First X-ray acquisition after catheter switch (H)*).

Accurately distinguishing between phases requires modeling longer-term contextual information, understanding not only what action is being performed but also where it occurs within the broader temporal structure of the surgical workflow. A common solution used in medical phase segmentation is to combine short-term models with a dedicated temporal module capable of learning long-range temporal dependencies.

In the medical phase segmentation literature, a wide range of temporal modeling approaches have been explored (see Appendix A for details). These include:

- *Hidden Markov Models (HMMs)*: Early works often leveraged HMMs to model temporal dependencies between surgical phases. HMMs capture the sequential nature of procedures using probabilistic state transitions and have the advantage of interpretability and low data requirements.[100;101;102]
- *Recurrent Neural Networks (RNNs)* and *Long Short-Term Memory networks (LSTMs)*: These models extend standard neural networks to sequential data. RNNs can capture short-term dependencies but are prone to vanishing gradients in long sequences, while LSTMs incorporate gating mechanisms to retain relevant long-term information and reduce phase flickering.[103;104;105]
- *Three-Dimensional Convolutional Neural Networks (3D-CNNs)*: By extending 2D convolutions into the temporal dimension, 3D-CNNs learn spatiotemporal motion patterns, such as hand or tool movements. However, they are computationally intensive and typically limited to short-range temporal dependencies.[106;107;108]
- *Temporal Convolutional Networks (TCNs)*: TCNs address the temporal limitations of 3D-CNNs using causal and dilated convolutions to capture both short- and long-range dependencies efficiently. Residual connections allow deep TCNs to be trained without suffering from the vanishing gradient problem.[109;110;111]
- *Transformer-based architectures*: Transformers leverage self-attention to model extremely long-range dependencies. Vision Transformers (ViTs) process frames as sequences of patches to encode spatial features, while Video Transformer Networks (VTNs) extend this mechanism to capture both spatial and temporal dependencies across frames. Although powerful, transformers require substantial annotated data and computational resources, often necessitating pretraining on large video datasets.[112;113;114]

The temporal module selected for this study must capture long-range dependencies while remaining data-efficient, given the limited size of the CAG-skeleton dataset. Hidden Markov Models (HMMs), while lightweight and interpretable, have limited modeling capacity due to their simplified probabilistic structure, which makes it difficult to capture complex temporal patterns and subtle variations in high-dimensional skeleton features. RNNs and 3D-CNNs face practical limitations: RNNs struggle with vanishing gradients over long sequences, while 3D-CNNs require substantial computational resources and large amounts of training data. Transformer-based models, including Vision Transformers and Video Transformer Networks, are capable of modeling extremely long-range dependencies but are generally infeasible in the medical domain due to their high data and pretraining requirements.

To balance temporal modeling range and data efficiency, this study therefore focuses on two widely adopted architectures in medical phase segmentation: LSTMs and TCNs, which can efficiently learn both short- and long-term dependencies.

2.3.3. Limited supervision learning

Over 95% of medical temporal segmentation studies rely on fully supervised learning methods, which require dense frame-level annotations that are costly, time-consuming, and dependent on expert surgical knowledge. Consequently, most available datasets remain relatively small, limiting the ability of supervised models to generalize across hospitals, surgeons, and patients. To address these limitations, various limited supervision strategies have been developed depending on the availability of annotations[22]:

- *Fully supervised TAS*: Each frame of every training video is annotated with an action label.
- *Point-level supervised TAS*: For each action instance in a video, a single frame (a ”point”) within its temporal duration is labeled.
- *Weakly supervised TAS*: Only coarse-grained labels are available for training. These can be an ordered list of occurring labels[115;116], or a set of all possible labels without information about order or occurrence[117].
- *Semi-supervised TAS*: The training set is divided into a small set of fully annotated videos and a (typically larger) set of unlabeled or weakly labeled videos.
- *Self-supervised TAS*: Models are pretrained on a pretext task (such as ordering shuffled frames) such that they learn meaningful features from unlabeled data. The learned features are then leveraged for a downstream task, i.e. supervised, semi-supervised, or unsupervised TAS.
- *Unsupervised TAS*: No labels are available for training.

Appendix B provides a detailed overview of these strategies. This work specifically explores transfer learning and pseudo-labeling.

The transfer learning approach leverages large-scale datasets from the human action recognition (HAR) domain to reduce reliance on medical data. The underlying idea is that abundant HAR data can be used during pretraining to capture the generic spatio-temporal structure of skeleton sequences, without depleting the limited medical annotations. After pretraining, the model is fine-tuned on the surgical phase segmentation task, allowing it to adapt to the domain-specific characteristics. This strategy maximizes the efficiency of the available surgical training data thereby reducing the quantity of medical annotations required.

Pseudo-labeling combines a small set of annotated data with a larger set of unlabeled data. A model trained on the labeled subset assigns provisional labels (pseudo-labels) to the unlabeled videos, which are then used for further training. This iterative process enables generalization beyond the labeled data. However, incorrect pseudo-labels can introduce noise and degrade performance. To address this, confidence thresholds or ensemble methods are typically used to filter unreliable predictions[24;25].

A common issue in pseudo-labeling is bias toward easy-to-classify actions: when applying a fixed confidence threshold, the majority of confident predictions often consist of simple or abundant classes. To mitigate this selection bias, Zou et al. (2018)[118] proposed a class-balanced self-training (CBST) framework, where different confidence scores are used per class for pseudo-label selection. CBST will also be explored in this study.

2.4. Contributions

In summary, this work addresses the challenges of limited data availability and poor generalizability in medical temporal segmentation by investigating the use of skeleton-based representations in combination with limited supervision techniques. Specifically, it evaluates the performance of two feature extractors, MS-G3D and PR-GCN, paired with two temporal models, LSTM and TCN, trained under three supervision regimes: full supervision, transfer learning using the Kinetics-skeleton dataset, and pseudo-labeling via class-balanced self-training on the CAG-skeleton dataset.

The main contributions of this work are as follows:

1. To the best of the authors’ knowledge, this is the first study to investigate skeleton-based medical temporal segmentation, introducing a modality to the medical domain that is well-established in

human action recognition and highly suitable for generalization across unseen clinical environments.

2. A general training framework is proposed for medical temporal segmentation of any human-performed procedure where skeleton sequences can be extracted or recorded.
3. This work explores the use of transfer learning and class-balanced self-training pseudo-labeling for skeleton-based medical phase segmentation, assessing their potential to reduce the need for costly annotated data while maintaining competitive segmentation performance.

3 Methodology

This chapter presents the proposed methodology used in this study. To establish a reference point for evaluating the effectiveness of the proposed methods, a simple non-learning baseline model is proposed in Section 3.1 that solely relies on the average duration of each surgical phase, computed across all procedures in the training set.

The proposed transfer learning approach requires alignment between the CAG-skeleton dataset and the Kinetics-skeleton dataset. Preprocessing steps to align the CAG-skeleton dataset in terms of structure, temporal resolution, and joint configuration are described in Section 3.2. The training procedures for the feature extractor and temporal models are presented in Sections 3.3 and 3.4, respectively. To reduce computational requirements, only the best-performing feature extractor temporal model combination is used for limited supervision analysis. The model combination selection, using statistical tests, is discussed in Section 3.5.

The limited supervision strategies, transfer learning using the Kinetics-skeleton dataset, and pseudo-labeling via class-balanced self-training, are discussed in Sections 3.6 and 3.7. To determine their effectiveness, the paired t-test is used, which is described in Section 3.8. Finally, the chapter concludes with Section 3.9, which presents the performance metrics used to evaluate and compare models across different training configurations and data subsets.

3.1. Baseline model

To establish a reference point for evaluating the effectiveness of the proposed methods, a simple non-learning baseline model was implemented. This baseline relies solely on the average duration of each surgical phase, computed across all procedures in the training set. For phases that occur at multiple points within the temporal sequence (e.g., Additional catheter change and X-ray after additional catheter change), the mean duration of each occurrence was calculated separately. The resulting sequence of average phase durations defines a mean procedure. During evaluation, this mean procedure is compared against all procedures in the test set to determine the baseline model's performance.

3.2. Dataset preprocessing

To enable efficient transfer learning from models pretrained on the Kinetics-skeleton dataset, the CAG-skeleton dataset was adapted to match its structure and characteristics. These modifications ensure that pretrained models can generalize to the CAG-skeleton dataset with minimal fine-tuning, thereby reducing the amount of CAG-specific data required for training.

3.2.1. Frame Rate Adjustment

The kinetics-skeleton dataset was recorded at 30 frames per second (fps), whereas the CAG-skeleton dataset was originally captured at 25 fps. To align the temporal resolutions, the CAG-skeleton dataset was upsampled to 30 fps by inserting an additional frame after every five original frames. For individuals with matching *person_id* values across adjacent frames, joint coordinates in the interpolated frame were estimated via linear interpolation.

3.2.2. Neck joint addition

Unlike the Kinetics-skeleton dataset, the CAG-skeleton dataset does not include a neck joint. To address this, two approaches were evaluated. First, a lightweight neural network with one hidden layer was trained on Kinetics-skeleton data to predict the neck position per frame from the remaining joints. Second, a simple approximation was tested, where the neck joint was placed at the midpoint between the left and right shoulder. The neural network offered insignificant gains over the midpoint method, which was not only computationally more efficient but also provided an approximation sufficiently close to the ground truth (Figure 3.1). Consequently, the midpoint approximation was chosen over the more complex solution.

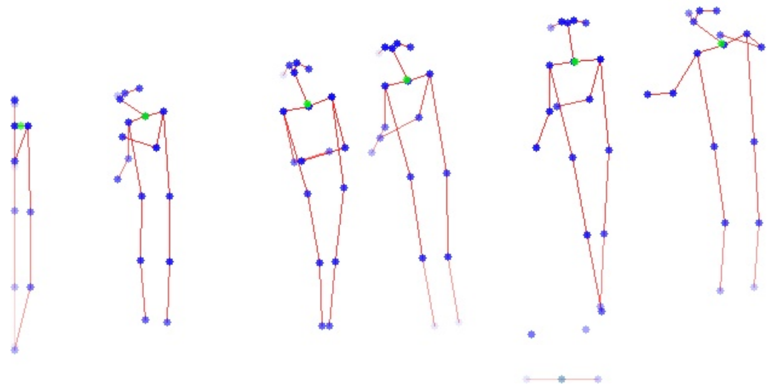


Figure 3.1: Frame 100 of kinetics-skeleton segment *__PYrzYbzKE* with label *jumpstyle dancing*. Green dot indicates the midpoint between the left and right shoulder. Blue dots indicate ground truth joint locations.

3.2.3. Normalize, reorder, filter, and segment

To further align with the Kinetics-skeleton format, the CAG-skeleton data was normalized, keypoints were reordered, and sequences were segmented into 10-second clips. During inspection, it was observed that the pose estimation occasionally produced clearly misplaced keypoints for a single frame, despite high confidence scores. To correct such anomalies, a median filter with a kernel size of 3 was applied independently to each person's pose sequence. This filter replaces each keypoint's coordinates with the median over a short temporal window, effectively smoothing abrupt noise without creating large motion blur. Figure 3.2 illustrates an example of this “keypoint jumping” before and after filtering.

3.2.4. Remaining discrepancies

Despite these preprocessing steps, several discrepancies remain. One issue is keypoint drift, where a joint gradually deviates from its true position over time. Unlike keypoint jumping, keypoint drift moves slowly, making it more difficult to detect and correct. A second limitation arises from the ground truth phase annotations, which are accurate only to within approximately five seconds. As a result, predictions at phase boundaries may be correct, yet appear misaligned with the ground truth annotated labels, as the label still reflects the preceding or subsequent phase. However, because phase boundaries only occur near a small fraction of the total number of clips, their local misalignments are unlikely to meaningfully affect the reported performance.

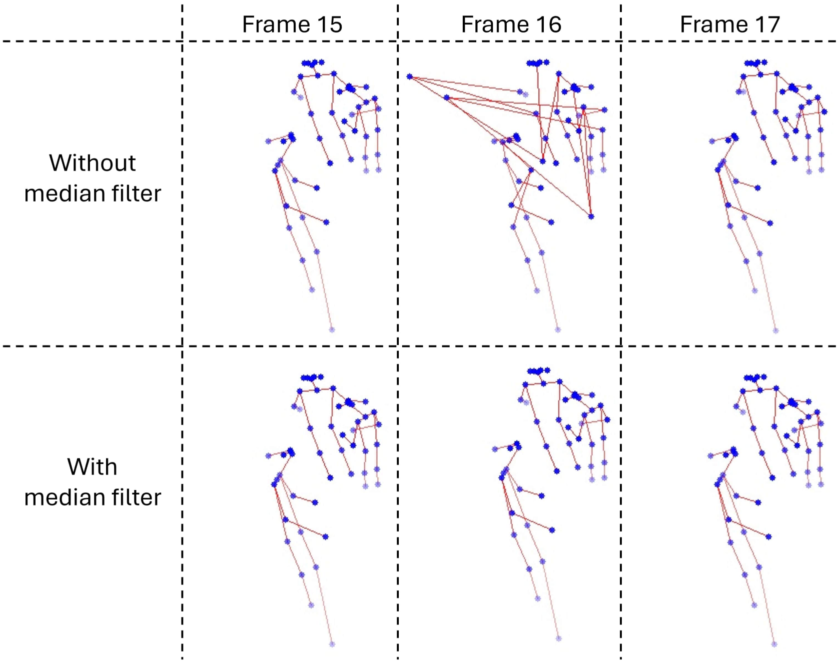


Figure 3.2: Keypoint jumping in frame 88126 of CAG-procedure with procedure_id 100 before and after median filter application.

3.3. Feature extractor training

Feature extractor models are trained on 10-second skeleton sequences (300 frames) to predict the medical phase of the final frame of the sequence. To ensure that the learned representations are maximally discriminative, training employs a cross-entropy loss, weighted inversely to phase occurrence of the training set. This weighting scheme prioritizes separability across all phases, preventing overfitting to common phases and promoting discriminative embeddings, which in turn enables downstream temporal models to more effectively distinguish between phases.

The model architectures are kept identical to their pretrained configurations to enable initialization with weights from the Kinetics-Skeleton dataset. Training was performed with a maximum of 300 epochs, early stopping with a patience of 15 epochs, and learning rate decay with a patience of 5 epochs and a decay factor of 0.5. All other hyperparameters, such as the base learning rate and batch size, were aligned with those used during pretraining on Kinetics-Skeleton.

A graphical overview of the feature extractor training pipeline is provided in Figure 3.3.

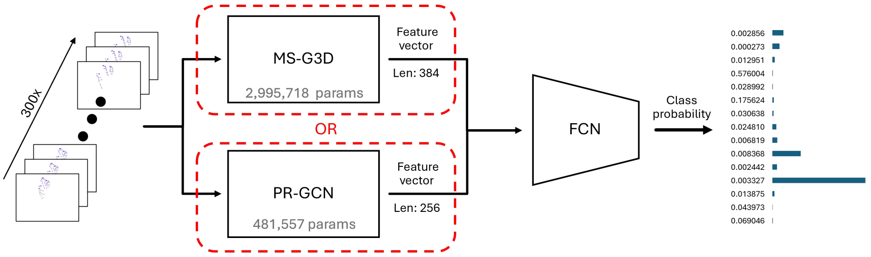


Figure 3.3: Overview of the training pipeline for the feature extractor. A 300-frame-long skeleton sequence is used as input for either the MS-G3D or PR-GCN model. The resulting feature vectors are used as input for a fully connected neural network (FCN) to predict the medical phase of the final frame in the sequence.

3.4. Temporal model training

The temporal models are trained on sequences of feature vectors obtained by passing consecutive 10-second clips through the feature extractor models. Each model is trained to predict the clinical phase of the final frame of the sequence.

Training is performed using non-weighted cross-entropy loss and the Adam optimizer. Similar to the feature extractor models, a learning rate scheduler with a decay factor of 0.5 and a patience of 5 epochs is used, in combination with early stopping using patience 15 and a maximum of 300 epochs. Each model is trained with a maximum temporal lookback of 250 consecutive 10-second clips (41 minutes and 40 seconds). For missing clips at the start of a procedure (e.g., up to 249 clips when predicting the phase of the first recorded clip of a procedure), zero feature vectors are inserted. These vectors were generated by passing an empty clip (without skeleton sequences) through the feature extractor.

A grid search over the hyperparameters listed in Table 3.1 is performed for each dataset and temporal model architecture. The LSTM and TCN training pipelines are graphically presented in figures 3.4 and 3.5, respectively.

Table 3.1: Hyperparameters searched during training of temporal models.

	LSTM	TCN
Learning rate	1e-3, 5e-4, 1e-4	1e-2, 1e-3, 1e-4
Batch size	16, 32, 64	16, 32, 64
Number of layers	1, 2, 3	4, 5, 6, 7, 8
Hidden layer size	64, 128, 256	32, 64, 128
Kernel size	N/A	3
Dropout	0.1, 0.2, 0.3	0.1, 0.2, 0.3

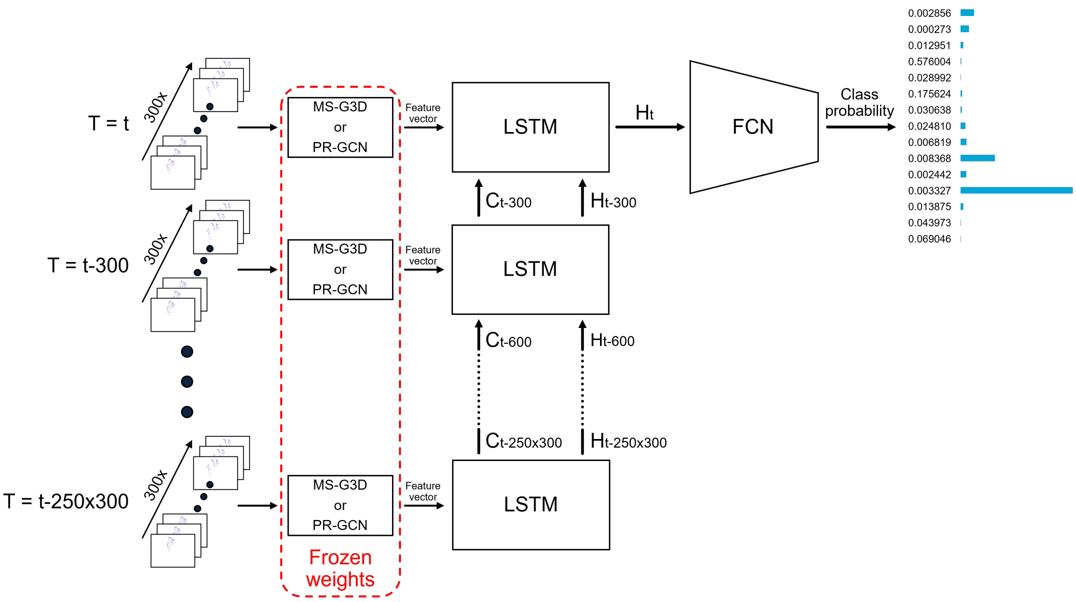


Figure 3.4: Training pipeline of the LSTM model. 250 feature vectors, obtained by passing 10-second clips through the feature extractor models, are used as input to the LSTM model. The final output of the LSTM model is used as input for a fully connected neural network (FCN) to predict the medical phase of the final frame of the sequence.

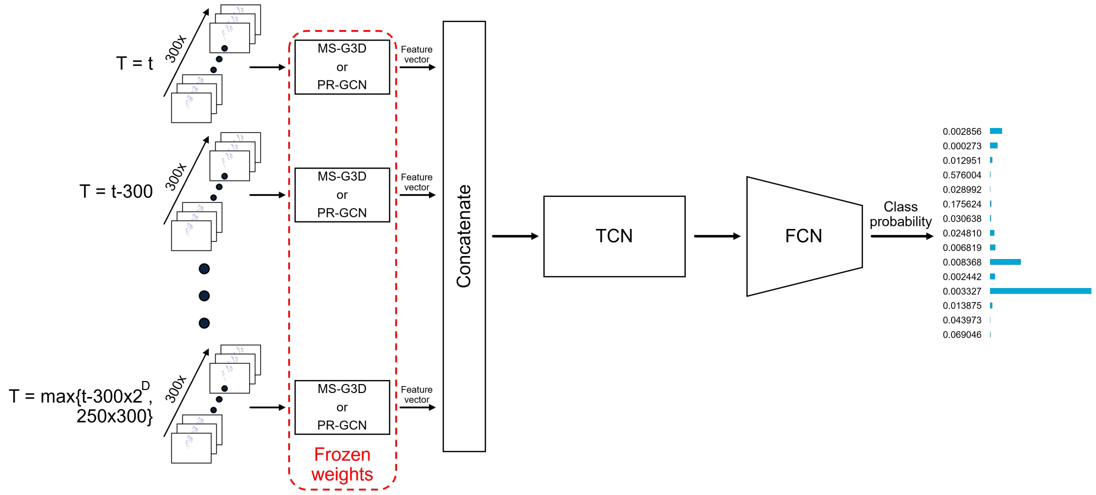


Figure 3.5: Training pipeline of the TCN model. 2^D (with D equal to the number of layers, or depth, of the TCN model) or up to 250 feature vectors, obtained by passing 10-second clips through the feature extractor models, are concatenated and used as input to the TCN model. The output of the TCN is passed to a fully connected neural network (FCN) to predict the medical phase of the final frame of the sequence.

3.5. Determining best model combination

To reduce computational requirements, only the best-performing model combination in the low data regime (up to 40 procedures) is used for limited supervision analysis. Model selection is based on the clip-wise and segmental performance metrics, which will be introduced in Section 3.9.

When comparing multiple machine learning models across different experimental conditions, it is important to determine whether observed performance differences are statistically significant. To do so, this study employs the Friedman test, a non-parametric test that is commonly used to detect differences among several related groups.[119]

In the Friedman test, the algorithms are ranked separately for each dataset, where the best-performing algorithm gets rank 1, the second-best rank 2, etc. The average rank of each algorithm is then computed by $R_j = \frac{1}{N}\sum_i r_i^j$, where r_i^j is the rank of the j-th out of k algorithms on the i-th out of N datasets.

The null hypothesis H_0 is that all models perform equivalently, i.e., they have equal expected ranks R_j. The alternative hypothesis is that at least one model differs significantly from the others.

$$\mathcal{X}_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{3.1}$$

If the Friedman statistic, computed using equation 3.1, is greater than a specified threshold (dependent on the value of k and N), the null hypothesis is rejected. When this is the case, a post-hoc test is required to identify which specific model has a statistically significantly different performance. In this work, the Nemenyi post-hoc test is used, which compares all classifiers pairwise. If the difference in average ranks between two models exceeds the critical difference, their performances are considered significantly different.

The best-performing model combination is selected for the limited supervision analysis. If no model shows statistically better performance, the most computationally efficient model is chosen.

3.6. Transfer learning training

As indicated in Section 2.3.3 and 2.2, the goal is to leverage the large-scale Kinetics-Skeleton dataset from the human action recognition domain to reduce reliance on the costly annotated medical data. Training occurs as described in sections 3.3 and 3.4, with the only change being that the feature extractor models are initialized with weights fully trained on the Kinetics-Skeleton dataset. This way, the feature extractor models transfer the knowledge of the generic spatio-temporal structure of skeleton sequences, learned on the Kinetics-Skeleton dataset, and only require fine-tuning on the limited medical data. As such, the limited medically annotated data is used efficiently, and not depleted early on in the training process to learn the generic spatio-temporal structure of skeleton sequences.

3.7. Pseudo-labeling

This study employs the class-balanced self-training (CBST) framework introduced by Zou et al. (2018)[118] (see Section 2.3.3). The selected model combination first generates predictions on the section of the training data not included in the training set used for training.

In the CBST framework, as described by Zou et al., the predictions are sorted class-wise based on their confidence scores. A parameter k_c is defined for each class c, such that exp $(-k_c)$ equals to the probability ranked at $round\,(p \times N_c)$ for that specific class. Here N_c denotes the number of predictions belonging to class c and p is the proportion of predictions selected as pseudo-labels (starting at 20%). A prediction is retained as a pseudo-label if the ratio between its confidence score and exp$(-k_c)$ for the corresponding class is greater than one.

In practice, this procedure is equivalent to simply selecting the top 20% most confident predictions within each class as pseudo-labels.

After pseudo-label selection, the entire training pipeline is retrained, as described in Sections 3.3 and 3.4, this time using both the original labeled dataset and the generated pseudo-labels.

Following retraining, Zou et al. again make predictions on the portion of the training data excluded from the initial training dataset (including the data previously used for pseudo-label generation). Next, the top 25% most confident predictions are selected as pseudo-labels for a second training iteration. This process is repeated in 5% increments, up to a maximum of 50%, or until the validation loss stops improving.

However, given the limited computational resources available, and the diminishing returns of pseudo-labeling when training with a greater labeled training set (and thus proportionally less unlabeled data available to generate pseudo-labels), this work explores pseudo-labeling up to 40 labeled training procedures and using 20% of unlabeled data as pseudo-labels only.

3.8. Determining limited supervision effectiveness

To evaluate the effectiveness of the limited supervision methods, the paired t-test is used. The pared t-test determines whether the observed differences in performance of two models evaluated on the same datasets are statistically significant. Unlike the Friedman or Nemenyi tests, which are designed for comparisons involving multiple models, the paired t-test is specifically suited for pairwise comparisons.

The test is based on the differences between paired performances, where each pair corresponds to the results of two models (with and without a limited supervision method) trained on the same dataset. The null hypothesis assumes that the mean difference μ_d between the models is zero, indicating no significant performance difference. The test statistic is computed as:

$$t = \frac{\mu_d}{\sigma_d/\sqrt{N}} \tag{3.2}$$

where μ_d is the mean of the performance differences, σ_d is their standard deviation, and N is the number of datasets. The resulting test statistic t is compared to the t-distribution to determine if the observed difference between the models is statistically significant.

3.9. Performance metrics

In this work, two complementary metrics are used to quantify the performance of the models: clip-wise accuracy, which quantifies frame-level correctness, and the normalized Levenshtein edit distance, which assesses the structural similarity between predicted and ground-truth phase sequences.

3.9.1. Clip-wise accuracy

Clip-wise accuracy measures the proportion of correctly classified clips relative to the total number of clips in a sequence. This metric provides a straightforward measure of overall classification performance and is intuitive to interpret. However, because each clip is evaluated independently, this metric does not account for the temporal or sequential structure of the predictions. Consequently, clip-wise accuracy may remain high even when a model fails to capture the correct procedural order or produces temporally inconsistent predictions (see predictions A and B in Figure 3.6).

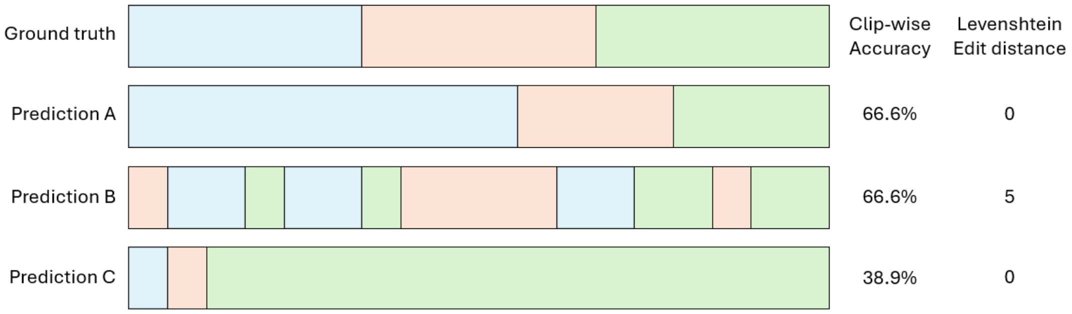


Figure 3.6: Clip-wise accuracy and Levenshtein edit distance score of three fictitious predictions, A, B, and C. The blue, red, and green colors represent three distinct phases.

3.9.2. Normalized Levenshtein edit distance

To account for temporal ordering and segmentation quality, the normalized Levenshtein edit distance is used. This metric evaluates the minimum number of operations (insertions, deletions, and substitutions) required to transform the predicted sequence of phase labels into the ground-truth sequence. The resulting distance is normalized by the length of the ground-truth sequence, enabling fair comparison across procedures of different durations.

Unlike clip-wise accuracy, the edit score measures how well the model has learned the sequential order of the procedure. It penalizes over- and under-segmentation as well as phase ordering errors. Lower values indicate better alignment between the predicted and actual procedural sequences.

Figure 3.6 illustrates the difference of clip-wise accuracy and edit distance using three fictitious predictions: A, B, and C. Although both predictions A and B achieve identical clip-wise accuracy, prediction B exhibits fragmented and temporally inconsistent outputs, leading to a substantially higher (worse) Levenshtein edit distance. Similarly, both predictions A and C achieve a perfect Levenshtein edit distance (zero) while prediction C shows a much lower clip-wise accuracy. This example demonstrates the complementary nature of both metrics: accuracy reflects the model's precision on a local clip-level, while the Levenshtein edit distance captures its performance at a global sequence-level scale.

4
Experiments

The dataset splitting strategy, following a leave-one-user-out (LOUO) protocol to prevent overfitting to individual surgeons, is outlined in Section 4.1. Sections 4.2 and 4.3 discuss the performed experiments, report the achieved performance, and analyze the strengths and weaknesses of the model.

4.1. Dataset splitting

Consistent with prior work[120;121;109], this study adopts a leave-one-user-out (LOUO) protocol, ensuring that no surgeon appears in both the training and test sets. This approach prevents potential overfitting to surgeon-specific behaviors or characteristics from artificially inflating test performance.

Since distinguishing individual surgeons from the video recordings is challenging, given recording quality and the standardized clothing, masks, and caps, a practical approximation was applied by splitting the dataset by surgeon gender. This produced 146 procedures performed by male surgeons for training and 43 procedures performed by female surgeons (22.8% of the labeled data) for testing.

To analyze performance across varying amounts of training data, 39 subsets were generated from the training set, containing 5 (x10 subsets), 10 (×10 subsets), 20 (×7 subsets), 30 (×4 subsets), 40 (×3 subsets), 50, 65, 80, 100, and 146 procedures, respectively. Each subset was further divided into training and validation splits using an 80/20 ratio. For smaller training sizes ($\leq$40 procedures), up to ten subsets (indicated as A, B, C, etc.) were created, while ensuring mutual exclusivity between subsets with the same procedure count (e.g., 5A $\cap$ 5B = $\varnothing$, 5B $\cap$ 5C = $\varnothing$, 5A $\cap$ 5C = $\varnothing$, etc.). Averaging performance across these subsets reduces the influence of outliers, which are especially common in low-data environments.

The construction of subsets A, B, and C followed a nested structure to allow for better comparison between the procedure counts. For example:

- 5A $\subset$ 10A $\subset$ 20A $\subset$ 30A $\subset$ 40A $\subset$ 50 $\subset$ 65 $\subset$ 80 $\subset$ 100 $\subset$ 146
- 5B $\subset$ 10B $\subset$ 20B $\subset$ 30B $\subset$ 40B $\subset$ 80 $\subset$ 100 $\subset$ 146
- 5C $\subset$ 10C $\subset$ 20C $\subset$ 30C $\subset$ 40C $\subset$ 146

A detailed overview of the first three subsets per procedure count, including the specific procedures they contain and their corresponding class occurrence rates, is provided in Appendix C. The training datasets are generally distributed similarly to the test set, while the validation set shows a class distribution that differs more substantially from both the training and test sets. Notably, the B-validation subsets completely lack clinical phases 12 and 13. Furthermore, because procedure lengths vary, datasets with the same number of procedures contain different numbers of clips.

4.2. Supervised learning results & model architecture selection

To identify the feature extractor–temporal model combination that performs best in a low-data setting, such as in medical phase segmentation, the models were trained on the first three datasets (A, B, and C) containing up to 40 procedures, and evaluated on the test set. Table 4.1 reports the average clip-wise segmentation accuracy across procedure counts. For comparison, the table also includes results from: (i) the non-learning baseline model which makes predictions for all test procedures using the average training procedure in the training set, and (ii) using a feature extractor alone without a temporal model. More detailed results, including clip-wise segmentation accuracy per individual dataset and per-phase accuracy, are provided in Appendix D.

Table 4.1: Mean clip-wise segmentation accuracy (%) ± std (%) evaluated on the test set of (i) non-learning baseline model; predicting all test procedures using the average training procedure in the training set, (ii) using a feature extractor alone without a temporal model, and (iii) using a feature extractor–temporal model combinations. All models are trained on all the first three subsets containing up to 40 procedures.

	Number of training procedures:				
Model	**5**	**10**	**20**	**30**	**40**
Training set average	34.02 ± 0.59	33.77 ± 0.94	32.89 ± 1.79	33.35 ± 1.98	33.73 ± 1.20
MS-G3D	24.12 ± 7.16	31.16 ± 4.05	38.20 ± 1.39	37.15 ± 5.43	41.11 ± 0.76
MS-G3D + LSTM	60.86 ± 0.94	66.44 ± 4.13	69.63 ± 3.46	71.76 ± 3.48	73.66 ± 1.10
MS-G3D + TCN	62.89 ± 3.06	66.12 ± 0.55	70.72 ± 0.31	**73.13 ± 1.94**	73.06 ± 0.77
PR-GCN	30.98 ± 4.55	35.61 ± 7.77	36.99 ± 5.06	41.81 ± 2.50	38.77 ± 5.97
PR-GCN + LSTM	61.63 ± 5.11	70.02 ± 2.01	69.85 ± 3.07	72.26 ± 4.18	74.35 ± 3.08
PR-GCN + TCN	**64.21 ± 3.44**	**72.12 ± 1.53**	**71.01 ± 2.58**	72.40 ± 3.10	**74.56 ± 2.48**

Comparing clip-wise segmentation accuracy across varying training set sizes (Table 4.1) shows that feature extractors alone achieve limited clip-wise segmentation accuracy (24–42%). Moreover, when training on small datasets it often performs worse than the non-learning baseline which simply predicts the average procedure sequence (32-34%). By contrast, feature extractor–temporal model combinations substantially improve performance, reaching accuracies above 70% with as few as 20 training procedures, highlighting the importance of temporal context.

Among the temporal models, both LSTM and TCN architectures yield comparable performance improvements, with TCN models showing a marginal advantage over LSTM-based counterparts in terms of clip-wise accuracy. In addition, PR-GCN–based models achieve a slightly higher mean accuracy compared to MS-G3D models, indicating a potential benefit of their small architectures. Nonetheless, a Friedman test, conducted using each individual dataset as a data point, revealed no statistically significant differences in clip-wise accuracy among any of the four evaluated model combinations. (see Appendix E).

To assess how well models learned the sequential order of a procedure, the normalized Levenshtein edit distance was computed (Table 4.2), which measures alignment between predicted and ground truth phase sequences. Here LSTM-based models clearly outperform TCN based achitectures. To ensure this difference is statistically relevant, a Friedman test was conducted.

Table 4.2: Mean normalized Levenshtein edit distance score ± std of feature extractor–temporal model combinations trained on the first three subsets containing up to 40 procedures (a score of zero indicates a perfect match between predicted phase sequence and ground truth phase sequence).

	Number of training procedures:				
Model	**5**	**10**	**20**	**30**	**40**
MS-G3D + LSTM	0.606 ± 0.041	0.556 ± 0.013	0.548 ± 0.046	**0.538 ± 0.026**	0.532 ± 0.008
MS-G3D + TCN	0.679 ± 0.067	0.730 ± 0.047	0.695 ± 0.025	0.672 ± 0.009	0.676 ± 0.055
PR-GCN + LSTM	**0.562 ± 0.043**	**0.533 ± 0.024**	**0.476 ± 0.032**	0.564 ± 0.020	**0.472 ± 0.038**
PR-GCN + TCN	0.692 ± 0.022	0.696 ± 0.034	0.627 ± 0.029	0.616 ± 0.064	0.619 ± 0.009

The Friedman test yielded a statistic of 34.28 ($p \ll 0.001$), indicating statistically significant differences among the evaluated models. A subsequent post-hoc Nemenyi test confirmed that LSTM-based models achieve significantly lower edit distances than TCN-based models. In contrast, no statistically significant differences were observed between MS-G3D– and PR-GCN–based models (see Appendix E).

Figure 4.1 provides a visual overview of the model predictions across two procedures. The models consistently underperform in recognizing uncommon phases (12 and 13) or short phases (1, 4, 6, 10, and 12), with the exception of phase 8 (Removal of second catheter), which is generally well identified. Additionally, all models tend to predict phase 0 at the beginning of each recording, even when the true initial phase occurs later in the procedure (which occurs when recordings begin after the actual start of the intervention). This can likely be attributed to a lack of temporal context.

Furthermore, predictions from TCN-based models contain phase flickering, characterized by rapid transitions between phases, indicating that these models have not learned the normal sequential progression of a procedure. This issue is exemplified by the PR-GCN + TCN predictions on procedure 234 (Figure 4.1), where the model incorrectly predicts phase 0 as the final phase. LSTM-based models show this behavior to a lesser extent. The differences in how well the models capture the inherent sequential structure of procedures directly correlates to the observed variations in Levenshtein edit distances (Table 4.2).

Figure 4.2 presents the confusion matrices for all six configurations trained on dataset 40A. Without explicit temporal modeling, feature extractor models (MS-G3D, PR-GCN) regularly confuse visually similar but temporally distant phases. For example, phase 11 (Patient exit & cleaning) is often misclassified as the visually similar phase 0 (Preparation before patient entry). This confusion is eliminated once temporal models (LSTM or TCN) are added and longer-range temporal context is taken into account, highlighting its importance.

Despite the use of a weighted loss function during training, the stand-alone feature extractor models never predict phases 4 (Endovascular access) or 12 (Additional catheter change). This suggests that these relatively short phases are context-dependent and visually similar to other phases. This problem persists and is extended to other short phases when temporal models are introduced. The usage of an unweighted loss likely contributes to this problem, as the model is incentivized to prioritize frequent, long-duration phases at the expense of rarer or shorter ones.

To reduce computational requirements, only the best-performing model combination in the low data regime is used for limited supervision analysis. Taken together, these results suggest that while the models do not statistically significantly differ in clip-wise segmentation accuracy, LSTM-based models showed superior sequential understanding compared to TCNs. Given their lower computational cost relative to MS-G3D, PR-GCN–based models are particularly attractive. For this reason, the PR-GCN + LSTM configuration was selected for further analysis under limited supervision.

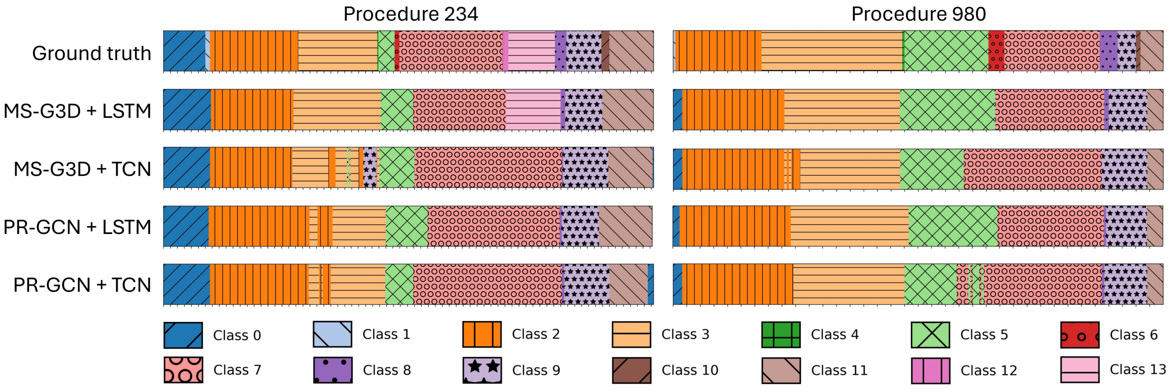


Figure 4.1: Predictions for all feature extractor-temporal model combinations trained on dataset 40A. Each tick on the horizontal axis presents one minute in the procedure (i.e. six clips).

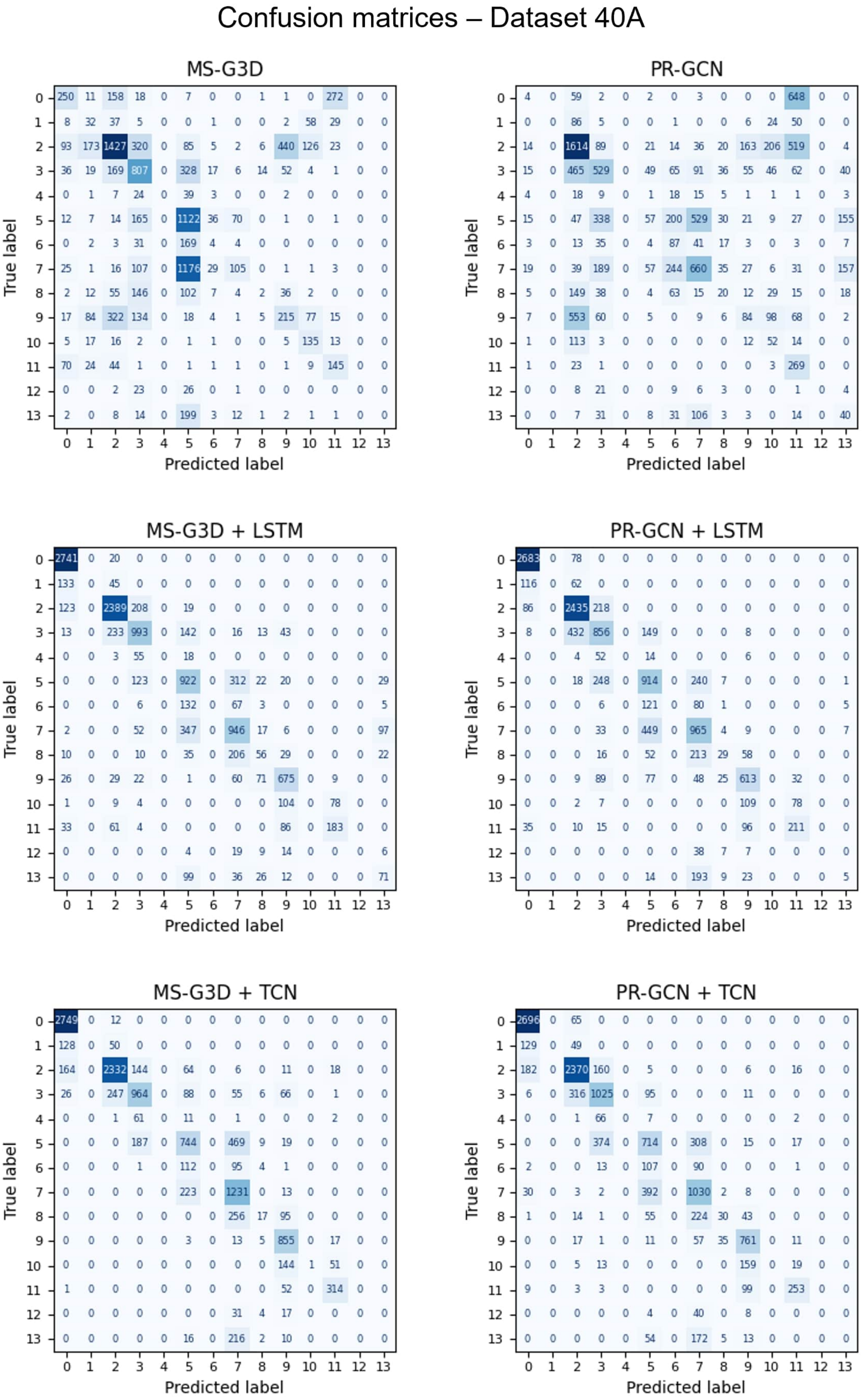


Figure 4.2: Confusion matrices for all six models trained on dataset 40A.

4.3. Extensive training & Limited supervision results

To allow for more confident comparisons, the PR-GCN, PR-GCN + LSTM, pretrained PR-GCN (prePR-GCN), and prePR-GCN + LSTM models were trained using all 39 datasets. As the pseudo-labeling strategy requires adequate amounts of unlabeled data to function well, it was trained on all datasets containing up to 40 procedures. For training on each individual training set, all training data not included in that specific set can be used to generate pseudo-labels.

4.3.1. Extensive training

Training the PR-GCN + LSTM model on all labeled training data resulted in a class-wise accuracy of 83.85%, with a normalized Levenshtein distance of 0.3837, significantly outperforming models trained on fewer data samples (see Tables 4.3 and 4.4).

Table 4.3: Mean clip-wise segmentation accuracy (%) of PR-GCN-based, pretrained PR-GCN-based (prePR-GCN), and PR-GCN + LSTM trained with pseudo-labeling (pseudo(PR-GCN + LSTM)) for all labeled dataset sizes.

	Number of training procedures:									
Model	**5**	**10**	**20**	**30**	**40**	**50**	**65**	**80**	**100**	**146**
PR-GCN	28.65	34.12	36.60	42.64	38.77	41.90	45.08	43.22	42.39	51.32
prePR-GCN	28.17	32.83	33.96	40.71	40.02	46.50	49.78	40.16	51.12	53.37
PR-GCN + LSTM	62.49	65.49	69.69	74.52	74.35	77.35	77.87	**77.68**	77.39	**83.95**
prePR-GCN + LSTM	60.05	67.85	69.18	72.47	74.00	**77.86**	**79.86**	75.95	**83.19**	83.85
pseudo(PR-GCN + LSTM)	**66.14**	**71.78**	**72.49**	**75.89**	**78.29**	N/A	N/A	N/A	N/A	N/A

Table 4.4: Mean normalized Levenshtein distances of PR-GCN + LSTM model trained with and without pretraining on Kinetics-skeleton (prePR-GCN + LSTM) and using pseudo-labeling (pseudo(PR-GCN + LSTM)) for all labeled dataset sizes.

	Number of training procedures:									
Model	**5**	**10**	**20**	**30**	**40**	**50**	**65**	**80**	**100**	**146**
PR-GCN + LSTM	.5616	**.5333**	**.4760**	.5641	**.4716**	**.3992**	.4951	.5201	.4663	**.3837**
prePR-GCN + LSTM	**.5578**	.5364	.4911	.5795	.5085	.4462	**.4907**	**.4379**	**.4085**	.4639
pseudo(PR-GCN + LSTM)	.5802	.5377	.4888	**.5254**	.4823	N/A	N/A	N/A	N/A	N/A

When comparing the confusion matrix of the PR-GCN + LSTM model trained on 40 versus 146 procedures (Figures 4.2 and 4.3), it can be seen that training on more data not only reduces the number of incorrect predictions, but also the severity of the errors is reduced, i.e., misclassifications tend to be temporally closer to the correct phase. Nevertheless, certain errors remain. These become apparent when analyzing the confusion matrices (Figure 4.3), temporally plotted predictions (Figure 4.4), and the original video data.

For example, the confusion matrix indicates that phase 1 (Patient entry) is frequently confused with phase 0 (Preparation before patient entry). The prediction plots suggest that this confusion arises both from small accumulated boundary offsets and a large misclassification in procedure X. Upon reviewing the video footage of procedure X, it becomes clear that the patient entered the room while preparations were still ongoing. The patient subsequently leaves, preparations resume, and the patient enters again three minutes later. In this segment of the procedure, the model's prediction actually reflects the true sequence of events more accurately than the ground truth labels, and is more informative.

Phases 2 (Patient transfer to table) and 3 (First contact of cardiologist with patient) are frequently confused. The prediction plots reveal two main causes: (i) the model occasionally predicts the first contact too early, and (ii) the boundary between these phases is often highly uncertain, leading the model to oscillate between them. Video inspection of the first cause shows that the model sometimes interprets actions performed close to the patient by nurses as the cardiologist's first contact. For instance, in procedure H, approximately five minutes into the recording, a nurse moves an object located below the patient's right wrist, the same location typically used by the cardiologist to administer local anesthetics, causing the model to mistakenly identify this as the start of phase 3.

Further video analysis addressing the second cause revealed inconsistencies within the annotated data itself. In procedure G, for example, the ground truth transition to phase 3 (First contact cardiologist) is incorrectly placed at the second rather than the first contact. As the model is trained on this mislabeled data, its ability to learn accurate phase boundaries is hindered. Additionally, in procedure J, vascular access is established via the patient's left arm. Here, the cardiologist's first contact occurs while standing on the opposite side of the hospital bed, a situation that is uncommon and thus underrepresented in the training data. Finally, several remaining confusions between phases 2 and 3 could not be attributed to the aforementioned causes, and no additional plausible explanations were identified in the other procedures.

Short phases remain particularly challenging. While phase 4 (Catheter insertion) is now occasionally detected, unlike in models trained on Dataset 40A, it is still frequently misclassified. This difficulty arises partly because phase 4 is not always present in the ground truth sequence. When the phase lasts less than 10 seconds, it may fall entirely within the middle of a clip where the start and end frames belong to phases 3 and 5. Furthermore, even slight temporal misalignments in the prediction can cause the model to miss the ground truth phase label altogether. Similarly, phase 6 (Catheter switch) is detected inconsistently. Video analysis indicates that its movements closely resemble other procedural actions, such as reinsertion or extraction of the guiding wire, or moving of the manifold or contrast tubing, making reliable discrimination of this phase difficult.

Phase 8 (Catheter removal), by contrast, is recognized more reliably, with only minor deviations in phase boundary placement. Nonetheless, two recurring types of errors are observed. First, similar to phase 6, it is occasionally confused with other movements that exhibit similar motion patterns. Second, phase 12 (Additional catheter change) is frequently misclassified as phase 8, as both begin with catheter removal and only the subsequent catheter insertion distinguishes both phases.

The confusion matrix reveals clear confusion among the X-ray phases (5, 7, and 13). The prediction plots show that this can be partly attributed to the model's inconsistent recognition of the catheter switch phases (6, 8, and 12). However, the model occasionally predicts phase 5 (X-ray before catheter switch) after the catheter switch has already occurred, indicating that it has not yet fully captured the correct temporal sequence of this segment of the procedure.

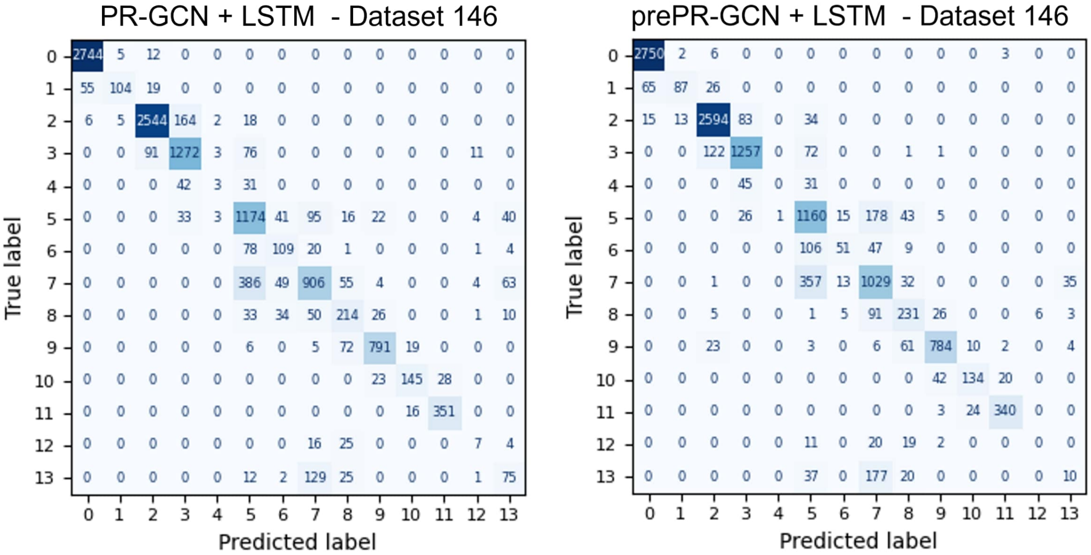


Figure 4.3: Confusion matrix of PR-GCN + LSTM model trained using 146 procedures, both with and without pretraining on the Kinetic-skeletons dataset.

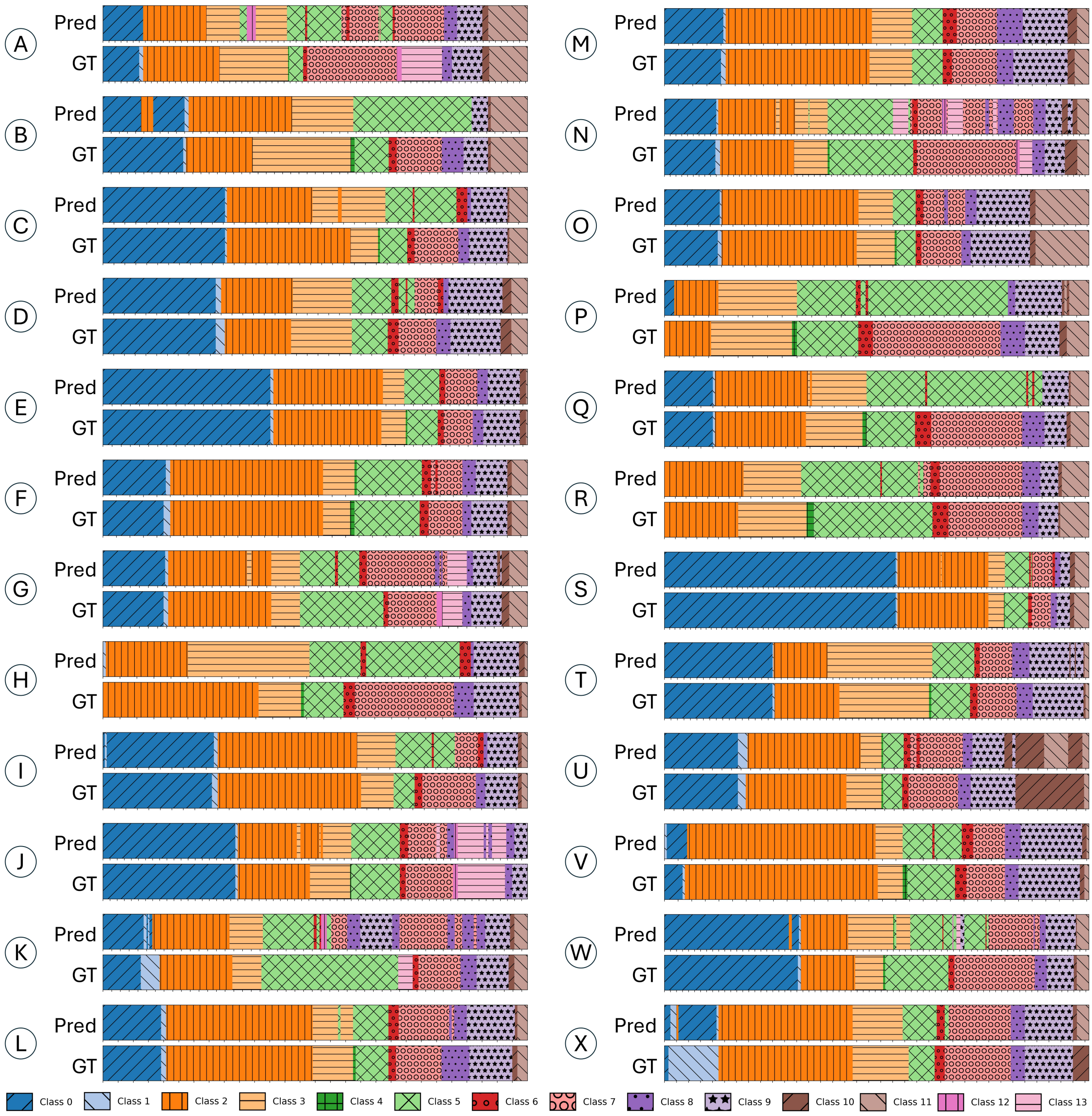


Figure 4.4: First 24 test set predictions (Pred) and ground truth labels (GT) of the PR-GCN + LSTM model trained on 146 procedures.

In contrast, later phases such as Wound closure (9), Patient off table (10), and Patient exit & cleaning (11) are recognized more consistently, and with relatively accurate phase boundaries. One noteworthy example is procedure U, in which the patient is transferred out of the room using a hospital bed. After wound closure, the patient remains briefly on the operating table before walking towards the hospital bed. The ground truth labels transition to phase 11 only when the hospital bed is rolled out of the room, whereas the model briefly predicts phase 11 during the transfer itself.

Other noteworthy findings from the video analysis include:

- Procedure I: Between 0 and 1 minutes, a nurse enters the room and is mistaken by the model for the patient, resulting in a one-clip misclassification as phase 1 (Patient entry).
- Procedure J: Near the end of the recording, the cardiologist walks away toward the screen to point something out to the patient or a nurse. This movement is misclassified as phase 8 (Catheter removal), likely because the cardiologist often performs a similar motion immediately after catheter

removal to allow the nurses to proceed with wound closure.

- Procedure K: The cardiologist first removes and reinserts a guiding wire, predicted by the model as phases 6 and 12, respectively, before deciding to switch vascular access from the right wrist to the groin. The model identifies the subsequent catheter removal as phase 8 (Catheter removal). The nurses then close the wound at the right wrist, which the model also identifies, although both steps are not included in the ground truth labels. Finally, the catheter insertion at the groin area is again predicted as phase 8 by the model.
- Procedure N: At 39 minutes, the model incorrectly predicts phase 13. However, additional catheter changes occurring at 48 and 52 minutes are visible in the video and model predictions, but absent from the ground truth annotations. The procedure concludes with the nurses assisting the patient off the table, noticing residual blood on the patient's wrist, helping the patient lie back down, cleaning the area, and then assisting the patient off the table once more. This sequence of actions is captured quite accurately by the model, yet it is not reflected in the annotated ground truth labels.

Overall, the predictions, confusion matrices, and video analyses demonstrate that the model has not yet fully learned the sequential order of the complete procedure. Temporally short phases that require near-perfect temporal alignment, as well as visually similar phases that differ only by subtle motion cues (e.g., catheter changes and X-ray phases), remain the primary sources of error. Nonetheless, the model exhibits notable robustness in handling rare or atypical procedural scenarios, often producing contextually reasonable predictions even when these deviate from the annotated ground truth.

The analysis also underscores the limitations of the ground truth labels themselves. Several annotations contain temporal inaccuracies, skip relevant actions, or impose a rigid phase sequence that does not always reflect real procedural variability. These inconsistencies partly explain some of the remaining misclassifications and indicate that model performance is not only restricted by the model quality but also by imperfect annotation quality. In this regard, the model's predictions occasionally provide a more faithful representation of the true procedural flow than the available ground truth labels.

4.3.2. Transfer learning

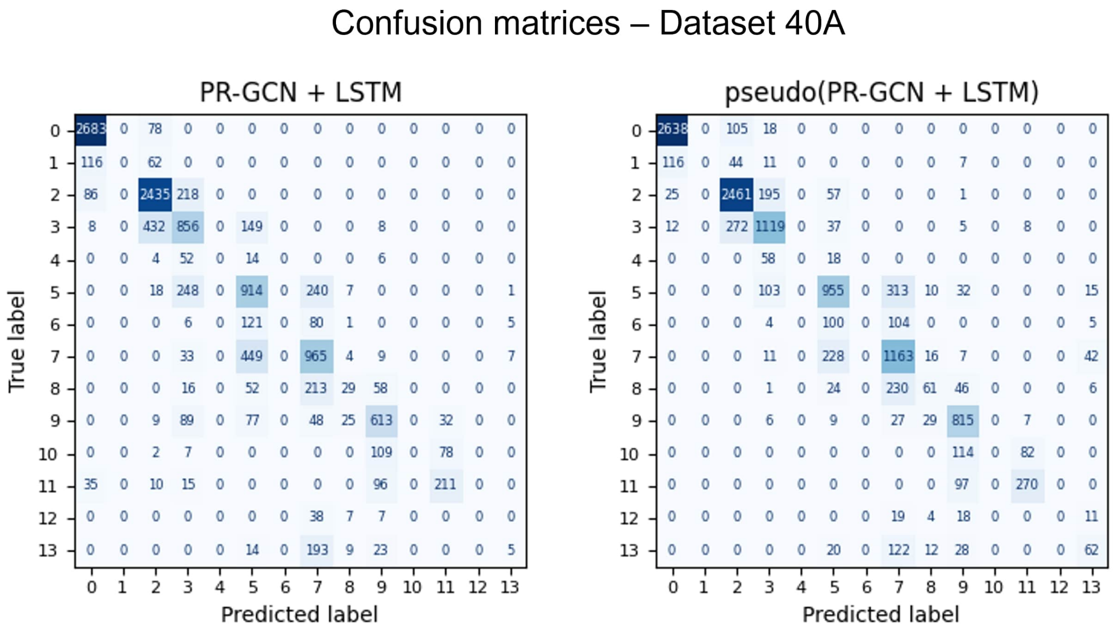
The performance of the PR-GCN models initialized with weights pretrained on the Kinetics-Skeleton dataset (pre-PR-GCN) is reported in Tables 4.3 and 4.4. Both the clip-wise accuracies and normalized Levenshtein distances show that pretrained initialization has no clear advantage over random initialization. This observation is supported statistically by a paired t-test, which yielded p-values of 0.936 for accuracy and 0.272 for Levenshtein distance, indicating that the observed performance differences are not significant. Moreover, a comparison of the confusion matrices for the randomly initialized and pretrained models (Figure 4.3) reveals no phase-specific performance improvements, implying that pretraining does not enhance recognition of specific procedural phases.

4.3.3. Pseudo-labeling

For each training dataset, the corresponding trained PR-GCN + LSTM model is used to generate predictions on the training samples that were not part of the dataset. The class-balanced self-training (CBST) framework, described in Section 3.7, is then applied to select per class the most confident 20% of these predictions as pseudo-labels. The clip-wise accuracies of both the generated predictions and the selected pseudo-labels are presented in Table 4.5. On average, the selected pseudo-labels are approximately 15% more accurate than the overall prediction set, confirming a strong positive correlation between model confidence and prediction accuracy.

Table 4.5: Mean prediction accuracy of the unlabeled data and pseudo-label accuracy per labeled training set size.

	Number of training procedures:				
Model	**5**	**10**	**20**	**30**	**40**
Unlabeled train data	54.75	63.46	64.83	66.36	69.00
Pseudo-labels	69.06	78.31	79.30	81.18	82.76



PR-GCN + LSTM

True label \ Predicted label	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	2683	0	78	0	0	0	0	0	0	0	0	0	0	0
1	116	0	62	0	0	0	0	0	0	0	0	0	0	0
2	86	0	2435	218	0	0	0	0	0	0	0	0	0	0
3	8	0	432	856	0	149	0	0	0	8	0	0	0	0
4	0	0	4	52	0	14	0	0	0	6	0	0	0	0
5	0	0	18	248	0	914	0	240	7	0	0	0	0	1
6	0	0	0	6	0	121	0	80	1	0	0	0	0	5
7	0	0	0	33	0	449	0	965	4	9	0	0	0	7
8	0	0	0	16	0	52	0	213	29	58	0	0	0	0
9	0	0	9	89	0	77	0	48	25	613	0	32	0	0
10	0	0	2	7	0	0	0	0	0	109	0	78	0	0
11	35	0	10	15	0	0	0	0	0	96	0	211	0	0
12	0	0	0	0	0	0	0	38	7	7	0	0	0	0
13	0	0	0	0	0	14	0	193	9	23	0	0	0	5

pseudo(PR-GCN + LSTM)

True label \ Predicted label	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	2638	0	105	18	0	0	0	0	0	0	0	0	0	0
1	116	0	44	11	0	0	0	0	0	7	0	0	0	0
2	25	0	2461	195	0	57	0	0	0	1	0	0	0	0
3	12	0	272	1119	0	37	0	0	0	5	0	8	0	0
4	0	0	0	58	0	18	0	0	0	0	0	0	0	0
5	0	0	0	103	0	955	0	313	10	32	0	0	0	15
6	0	0	0	4	0	100	0	104	0	0	0	0	0	5
7	0	0	0	11	0	228	0	1163	16	7	0	0	0	42
8	0	0	0	1	0	24	0	230	61	46	0	0	0	6
9	0	0	0	6	0	9	0	27	29	815	0	7	0	0
10	0	0	0	0	0	0	0	0	0	114	0	82	0	0
11	0	0	0	0	0	0	0	0	0	97	0	270	0	0
12	0	0	0	0	0	0	0	19	4	18	0	0	0	11
13	0	0	0	0	0	20	0	122	12	28	0	0	0	62

Figure 4.5: Confusion matrix of PR-GCN + LSTM model trained using dataset 40A, both with and without pseudolabeling.

The models were then retrained using both the original labeled dataset and the generated pseudo-labels. For all datasets, the validation scores of the retrained models improved compared to those of the original models. The clip-wise accuracies and normalized Levenshtein distances of models trained with and without pseudo-labeling are reported in Tables 4.3 and 4.4. While a clear improvement in clip-wise segmentation accuracy is observed, the normalized Levenshtein distances show no clear improvements. This observation is supported by a paired t-test, which yielded a p-value $\ll 0.001$ for accuracy and 0.237 for Levenshtein distance.

Comparing the confusion matrices of models trained with and without pseudo-labeling (Figure 4.5), along with video analysis of major errors, reveals that pseudo-labeling does not eliminate any specific type of error. Instead, it reduces the frequency of all error types.

5 Discussion & Conclusion

This thesis aimed to tackle two key barriers to the clinical adoption of temporal segmentation in the medical domain: (1) the scarcity of annotated data and (2) the difficulty of achieving robust generalization across different surgical settings. In order to achieve this, it explored 2D skeleton-based temporal phase segmentation for cardiac angiography (CAG) procedures using the CAG-skeleton dataset, with the goal of identifying 14 procedural phases based solely on human skeleton sequences extracted from external video.

5.1. Promising model architecture

Consistent with prior work in temporal segmentation, a two-stage architecture was adopted, consisting of (i) a feature extractor that encodes short-term spatio-temporal patterns in the skeleton data, and (ii) a temporal model that captures long-range dependencies. A variety of skeleton-based feature extractors, used in the neighboring human action recognition domain, and temporal models, commonly used in medical temporal segmentation, were reviewed and compared.

Among these architectures, PR-GCN and MS-G3D feature extractors, and LSTM and TCN temporal models showed the greatest promise in terms of data efficiency and potential performance. Combinations of all four models were experimentally evaluation on a series of small subsets of the CAG-skeleton dataset, which revealed no statistically significant difference in frame-level classification accuracy. However, architectures incorporating LSTM models demonstrated a superior understanding of procedural order and temporal continuity, likely owing to the LSTM's internal memory mechanism.

Considering both architecture performance and computational efficiency, the PR-GCN + LSTM combination was selected for further investigation. The resulting model, consisting of a PR-GCN feature extractor, an LSTM-based temporal model, and a fully connected classification head, was trained on 146 CAG procedures and achieved a clip-wise segmentation accuracy of 83.95%, highlighting the potential of the underexplored skeleton modality for medical workflow analysis.

Despite these promising results, several challenges remain. The sequential order of the entire procedure is still not perfectly learned. Especially temporally short phases, requiring near-perfect alignment, and phases where movements closely resemble other procedural actions, continue to cause misclassifications. Moreover, detailed inspection revealed limitations in the ground truth annotations, which sometimes exhibit temporal inaccuracies, skip relevant actions, or enforce a rigid procedural order that does not always align with real-world surgical variability. Nonetheless, the model exhibits notable robustness, often generating plausible predictions in atypical or noisy scenarios that deviated from the annotated ground truth.

The persistent bottleneck of limited annotated data motivated the subsequent exploration of limited supervision strategies, aiming to further enhance model generalization and clinical applicability.

5.2. Limited supervision

To address the scarcity of annotated training data, this study investigated two strategies for leveraging additional information: transfer learning and pseudo-labeling.

5.2.1. Transfer learning

Pretraining the PR-GCN feature extractor on the large-scale Kinetics-skeleton dataset (i.e. transfer learning) yielded no statistically significant improvement in segmentation accuracy or normalized Levenshtein distance. This outcome suggests that either the knowledge learned from Kinetics-skeleton does not effectively transfer to the surgical domain, or that the information transferred is relatively easy for the model to learn from scratch during training. A combination of both explanations is most plausible.

The domain gap between the Kinetics-skeleton and surgical datasets is substantial. Most actions in Kinetics-skeleton involve large-scale, full-body human activities such as sports or daily motions, whereas surgical workflows are characterized by fine-grained hand and arm movements captured from a fixed and relatively distant camera viewpoint. Consequently, the knowledge transferred from pretraining may be limited to low-level spatiotemporal relationships between joints, such as basic motion continuity, which can be rapidly relearned during task-specific training. In contrast, the higher-level temporal dependencies and contextual relations required to differentiate surgical phases are largely absent from the source dataset, reducing the effectiveness of the pretrained initialization.

Consequently, the pretrained weights may transfer only marginally useful knowledge to the surgical domain, providing no meaningful advantage over random initialization. Utilizing more domain-specific pretraining or self-supervised representation learning on unlabeled surgical videos could be used to obtain spatiotemporal features that are both relevant and transferable to medical temporal segmentation tasks.

5.2.2. Pseudo-labeling

Pseudo-labeling, in contrast, showed clear potential for exploiting unlabeled data. Even with a single pseudo-labeling iteration, it improved segmentation accuracy by approximately 15 percentage points in low-data regimes. This indicates that incorporating confidently predicted labels into the training process can substantially enhance data efficiency and generalization. Since this work applied only one pseudo-labeling cycle, following the complete learning schedule proposed by Zou et al.[25] may further boost performance.

5.3. Recommendations and future work

Despite achieving a strong performance of 83.95% clip-wise accuracy, several systematic errors remain. The next subsections examine these errors and potential remedies, while the final subsection highlights future research needed to better understand the proposed approach and to advance its clinical readiness for adoption.

5.3.1. Short phases

Rare or short phases, particularly catheter insertion, remained under-recognized. This can likely be attributed to models prioritizing longer and more frequent phases. Potential solutions include the use of weighted loss functions, which encourage the model to pay more attention to underrepresented classes. Weights should be carefully selected to avoid overemphasizing short phases, which could reduce clinical relevance of the achieved predictions. Alternatively, integrating object detections (e.g. of the catheters, guiding wires or patient) could provide highly informative cues for rare events.

5.3.2. Sequential understanding

Although the LSTM-based models showed statistically significantly better sequential understanding compared to TCN-based models, errors remained. These errors were most prominent in visually similar phases, such as catheter switches and X-ray acquisitions, but also showed a limited understanding of overall procedural order. For example, the model occasionally predicted phase 5 (X-ray before catheter switch) after phase 6 (catheter switch), which is clinically impossible.

To enhance temporal modeling capabilities and reduce such errors, several strategies are recommended:

1. A penalty term could be introduced to the loss function that increases with the temporal distance between the predicted and true phase. For instance, if the ground truth is phase 5, misclassifying it as phase 7 should result in a stronger penalty than misclassifying it as phase 6.
2. Post-hoc sequence filtering, where models with inherent sequential constraints, such as Hidden Markov Models, are used after the LSTM or fully connected network, could eliminate impossible phase transitions.
3. Sequence masking, where phases that have already occurred cannot reappear after a defined threshold, may reduce sequential order errors, as it enforces procedural order.

In addition, more advanced sequence models such as transformers may offer further improvements by modeling even longer-range temporal dependencies. However, these architectures typically require large datasets for training, thus their advantages are unlikely to be fully realized within the low-data setting of medical phase segmentation.

5.3.3. Annotation

Video analysis revealed that several procedural annotations contain temporal inaccuracies, omit relevant actions, or impose a rigid phase sequence that does not always reflect real procedural variability. These annotation errors could be mitigated by employing a second annotator to label all videos, with discrepancies resolved through discussion and mutual agreement. However, this approach would effectively double the labeling cost. The same effort might be more efficiently spent on expanding the overall size of the annotated dataset.

Uncommon occurrences during procedures are inevitable in real-world clinical settings. For example, a patient entering the procedural room during preparation or the need to repeat wound closure after the patient begins to move off the table are uncommon scenarios but still occur in procedures X and U (Figure 4.4), respectively. The current phase labels may impose a sequence that is too rigid to capture such real-world variability. Since each procedural phase consists of multiple distinct actions (e.g., the X-ray phases involve actions such as manipulating and rotating the catheter, injecting contrast fluids, etc.), action segmentation could offer a more flexible, fine-grained, and procedurally accurate alternative. Therefore, future work could investigate the segmentation of individual actions rather than rigid phases. If required, these recognized actions could then serve as inputs for phase segmentation or be used directly in robotic assistance systems, providing more actionable insights than phase labels alone.

5.3.4. Model insights and clinical readiness

To better understand the proposed approach and advance its clinical readiness for adoption, future work should explore, in addition to the aforementioned aspects, the following areas:

- Ablation of feature extractor: While adding a temporal model to the PR-GCN feature extractor substantially improved clip-wise segmentation accuracy, the independent contribution of the feature extractor remains unexplored. Ablation studies using skeleton data directly with a temporal model (e.g., LSTM or TCN) could clarify its relative importance.
- Video-modality comparison: This study relied exclusively on skeleton data. Future work could use the raw video data to assess whether skeletons provide a significant performance gain or if video-based features offer additional benefits. Such studies would clarify the trade-off between low-dimensional, highly informative skeleton representations and richer, but noisy visual modalities.
- Generalizability across clinical settings: Although the model generalized well across patients and surgeons within the dataset, its robustness to variations in hospital layout, procedural protocols, equipment, camera setups, or skeleton detection algorithms remains unexplored. Evaluating performance across different geographical locations and institutions is essential before clinical adoption.
- 3D skeleton usage: Employing 3D skeletons could improve clinical adaptability by reducing sensitivity to camera viewpoint and orientation. This would make the approach more practical to use across diverse hospital environments.

5.4. Conclusion

In conclusion, this thesis demonstrates the feasibility and possibilities of skeleton-based medical temporal segmentation using cardiac angiography procedures. It reviewed several model architectures, and the proposed PR-GCN + LSTM model showed strong potential in low-data settings. Transfer learning using the Kinetics-skeleton dataset showed no statistically significant performance gains, suggesting that the knowledge learned from Kinetics-skeleton does not effectively transfer to the surgical domain, and/or that the information transferred is relatively easy for the model to learn from scratch during training. In contrast, pseudo-labeling via class-balanced self-training showed great potential for reducing annotation requirements as it provided consistent improvements to the models' clip-wise segmentation accuracy in the low data regime. Nevertheless, challenges remain in accurately modeling rare or short phases, improving sequential understanding, and ensuring generalizability across medical contexts. Addressing these limitations through loss function engineering, 3D skeleton usage, post-hoc filtering and more advanced sequence modeling will be crucial steps toward reliable, scalable, and clinically useful workflow analysis systems.

Declaration of AI Assistance

OpenAI's Chat-GPT has been used for this paper to provide new perspectives and improve the report writing. Although Chat-GPT provides substantial assistance, there is a large possibility of encountering inaccuracies and incorrect information. Therefore, individual research must always be conducted to validate Chat-GPT's answers. In the end, the author's own perspectives were broadened with those provided by Chat-GPT.

Bibliography

[1] Daniel A Hashimoto, Guy Rosman, Daniela Rus, and Ozanan R Meireles. Artificial intelligence in surgery: promises and perils. *Annals of surgery*, 268(1):70–76, 2018.

[2] Martin Wagner, Beat-Peter Müller-Stich, Anna Kisilenko, Duc Tran, Patrick Heger, Lars Mündermann, David M Lubotsky, Benjamin Müller, Tornike Davitashvili, Manuela Capek, et al. Comparative validation of machine learning algorithms for surgical workflow and skill analysis with the heichole benchmark. *Medical image analysis*, 86:102770, 2023.

[3] Isabel Funke, Sören Torge Mees, Jürgen Weitz, and Stefanie Speidel. Video-based surgical skill assessment using 3d convolutional neural networks. *International journal of computer assisted radiology and surgery*, 14(7):1217–1225, 2019.

[4] Isabel Funke, Dominik Rivoir, Stefanie Krell, and Stefanie Speidel. Tunes: A temporal u-net with self-attention for video-based surgical phase recognition. *IEEE Transactions on Biomedical Engineering*, 2025.

[5] C.-B. Chng, W. Lin, Y. Hu, Y. Hu, J. Liu, and C.-K. Chui. Automatic step recognition with video and kinematic data for intelligent operating room and beyond. In *ACM Int. Conf. Proc. Ser.*, pages 599–606. Association for Computing Machinery. ISBN 979-840070891-6 (ISBN). doi: 10.1145/3628797.3628999. URL `https://www.scopus.com/inward/record.uri?eid=2-s2.0-85180550841&doi=10.1145%2f3628797.3628999&partnerID=40&md5=7f4a3ec7d5c839cb3af862629a92515e`. Journal Abbreviation: ACM Int. Conf. Proc. Ser.

[6] Alaa Merghani Abdelrazig Merghani, Abdullah Khaled Ahmed Esmail, Ahmed Mohamed Elamin Mubarak Osman, Nihal Ahmed Abdelfrag Mohamed, Safwa Mustafa Mohamed Ali Shentour, and Shaima Merghani Abdelrazig Merghani. The role of machine learning in management of operating room: A systematic review. 17(2):e79400, 2023. ISSN 2168-8184. doi: 10.7759/cureus.79400. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11929973/`.

[7] Zhili Yuan, Jialin Lin, and Dandan Zhang. Hierarchical semi-supervised learning framework for surgical gesture segmentation and recognition based on multi-modality data. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7659–7666. IEEE, 2023.

[8] Simon C Williams, Jinfan Zhou, William R Muirhead, Danyal Z Khan, Chan Hee Koh, Razna Ahmed, Jonathan P Funnell, John G Hanrahan, Alshaymaa Mortada Ali, Shankhaneel Ghosh, et al. Artificial intelligence assisted surgical scene recognition: A comparative study amongst healthcare professionals. *Annals of Surgery*, pages 10–1097, 2024.

[9] Thomas M Ward, Danyal M Fer, Yutong Ban, Guy Rosman, Ozanan R Meireles, and Daniel A Hashimoto. Challenges in surgical video annotation. *Computer Assisted Surgery*, 26(1):58–68, 2021.

[10] Abdolrahim Kadkhodamohammadi, Nachappa Sivanesan Uthraraj, Petros Giataganas, Gauthier Gras, Karen Kerr, Imanol Luengo, Sam Oussedik, and Danail Stoyanov. Towards video-based surgical workflow understanding in open orthopaedic surgery. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 9(3):286–293, 2021.

[11] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. A survey of depth and inertial sensor fusion for human action recognition. *Multimedia Tools and Applications*, 76(3):4405–4425, 2017.

[12] Nicolas Padoy, Tobias Blum, Seyed-Ahmad Ahmadi, Hubertus Feussner, Marie-Odile Berger, and Nassir Navab. Statistical modeling and recognition of surgical workflow. *Medical image analysis*, 16(3):632–641, 2012.

[13] Jakob E Bardram, Afsaneh Doryab, Rune M Jensen, Poul M Lange, Kristian LG Nielsen, and Søren T Petersen. Phase recognition during surgical procedures using embedded and body-worn sensors. In *2011 IEEE international conference on pervasive computing and communications (PerCom)*, pages 45–53. IEEE, 2011.

[14] Atsushi Nara, Kiyoshi Izumi, Hiroshi Iseki, Takashi Suzuki, Kyojiro Nambu, and Yasuo Sakurai. Surgical workflow monitoring based on trajectory data mining. In *JSAI International Symposium on Artificial Intelligence*, pages 283–291. Springer, 2010.

[15] Matthew Stephen Holden, Tamas Ungi, Derek Sargent, Robert C McGraw, Elvis CS Chen, Sugantha Ganapathy, Terry M Peters, and Gabor Fichtinger. Feasibility of real-time workflow segmentation for tracked needle interventions. *IEEE Transactions on Biomedical Engineering*, 61(6):1720–1728, 2014.

[16] Adam James, D Vieira, Benny Lo, Ara Darzi, and G-Z Yang. Eye-gaze driven surgical workflow segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 110–117. Springer, 2007.

[17] R Stauder, A Okur, and N Navab. Detecting and analyzing the surgical workflow to aid human and robotic scrub nurses. In *7th Hamlyn Symposium on Medical Robotics. London*, 2014.

[18] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1–10, 2017.

[19] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4929–4937, 2016.

[20] Liqi Feng, Yaqin Zhao, Wenxuan Zhao, and Jiaxi Tang. A comparative review of graph convolutional networks for human skeleton-based action recognition. *Artificial Intelligence Review*, 55(5):4275–4305, 2022.

[21] J Shin, N Hassan, A Miah, and S Nishimura. A comprehensive methodological survey of human activity recognition across diverse data modalities. arxiv 2024. *arXiv preprint arXiv:2409.09678*, 2024.

[22] Elahe Vahdani and Yingli Tian. Deep learning-based action detection in untrimmed videos: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4302–4320, 2022.

[23] Guodong Ding, Fadime Sener, and Angela Yao. Temporal action segmentation: An analysis of modern techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2):1011–1030, 2023.

[24] Zhi-Hui You, Jia-Xin Wang, Si-Bao Chen, Jin Tang, and Bin Luo. Fmwdct: Foreground mixup into weighted dual-network cross training for semisupervised remote sensing road extraction. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15: 5570–5579, 2022.

[25] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. PseudoSeg: Designing pseudo labels for semantic segmentation, . URL `http://arxiv.org/abs/2010.09713`.

[26] Rick M Butler, Teddy S Vijfvinkel, Emanuele Frassini, Sjors van Riel, Chavdar Bachvarov, Jan Constandse, Maarten van der Elst, John J van den Dobbelsteen, and Benno HW Hendriks. 2d human pose tracking in the cardiac catheterisation laboratory with byte. *Medical Engineering & Physics*, 135:104270, 2025.

[27] Rick M Butler, Emanuele Frassini, Teddy S Vijfvinkel, Sjors van Riel, Chavdar Bachvarov, Jan Constandse, Maarten van der Elst, John J van den Dobbelsteen, and Benno HW Hendriks. Benchmarking 2d human pose estimators and trackers for workflow analysis in the cardiac catheterization laboratory. *Medical Engineering & Physics*, 136:104289, 2025.

[28] MMPOse. 2d human keypoints dataset. url: https://mmpose.readthedocs.io/zh-cn/latest/dataset_zoo/2d_body_keypoint.html, 2020. [Accessed 13-07-2025].

[29] PVV Kishore, D Anil Kumar, AS Chandra Sekhara Sastry, and E Kiran Kumar. Motionlets matching with adaptive kernels for 3-d indian sign language recognition. *IEEE Sensors Journal*, 18(8):3327–3337, 2018.

[30] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L Berg, and Dimitris Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 28–35. IEEE, 2012.

[31] Jinhyeok Jang, Dohyung Kim, Cheonshu Park, Minsu Jang, Jaeyeon Lee, and Jaehong Kim. Etri-activity3d: A large-scale rgb-d dataset for robots to recognize daily activities of the elderly. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10990–10997. IEEE, 2020.

[32] Manuel Martin, Alina Roitberg, Monica Haurilet, Matthias Horne, Simon Reiß, Michael Voit, and Rainer Stiefelhagen. Drive&act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2801–2810, 2019.

[33] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. Mocap database hdm05. *Institut für Informatik II, Universität Bonn*, 2(7), 2007.

[34] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1290–1297. IEEE, 2012.

[35] Lu Xia, Chia-Chih Chen, and Jake K Aggarwal. View invariant human action recognition using histograms of 3d joints. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 20–27. IEEE, 2012.

[36] Chris Ellis, Syed Zain Masood, Marshall F Tappen, Joseph J Laviola Jr, and Rahul Sukthankar. Exploring the trade-off between accuracy and observational latency in action recognition. *International Journal of Computer Vision*, 101(3):420–436, 2013.

[37] Omar Oreifej and Zicheng Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 716–723, 2013.

[38] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3192–3199, 2013.

[39] Matteo Munaro, Gioia Ballin, Stefano Michieletto, and Emanuele Menegatti. 3d flow estimation for human action recognition from colored point clouds. *Biologically Inspired Cognitive Architectures*, 5:42–51, 2013.

[40] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *2013 IEEE workshop on applications of computer vision (WACV)*, pages 53–60. IEEE, 2013.

[41] Mathieu Barnachon, Saïda Bouakaz, Boubakeur Boufama, and Erwan Guillou. Ongoing human action recognition with motion capture. *Pattern Recognition*, 47(1):238–247, 2014.

[42] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning, and recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, . doi: 10.1109/cvpr.2014.339. URL `https://ieeexplore.ieee.org/document/6909735`.

[43] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 168–172, . doi: 10.1109/ICIP.2015.7350781. URL `https://ieeexplore.ieee.org/document/7350781/`.

[44] An-An Liu, Yu-Ting Su, Ping-Ping Jia, Zan Gao, Tong Hao, and Zhao-Xuan Yang. Multiple/single-view human action recognition via part-induced multitask structural learning. *IEEE transactions on cybernetics*, 45(6):1194–1208, 2014.

[45] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+d: A large scale dataset for 3d human activity analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. doi: 10.1109/cvpr.2016.115. URL `http://ieeexplore.ieee.org/document/7780484/`.

[46] Chunhui Liu, Yueyu Hu, Yanghao Li, Sijie Song, and Jiaying Liu. Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding, 2017.

[47] Yanli Ji, Feixiang Xu, Yang Yang, Fumin Shen, Heng Tao Shen, and Wei-Shi Zheng. A large-scale RGB-d database for arbitrary-view human action recognition. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1510–1518. ACM. doi: 10.1145/3240508.3240675. URL `https://dl.acm.org/doi/10.1145/3240508.3240675`.

[48] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset, 2017.

[49] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[50] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: Realtime multi-person 2d pose estimation using part affinity fields. 43(1):172–186. ISSN 1939-3539. doi: 10.1109/TPAMI.2019.2929257. URL https://ieeexplore.ieee.org/abstract/document/8765346.

[51] Enrico Calabrese, Gemma Taverni, Christopher Awai Easthope, Sophie Skriabine, Federico Corradi, Luca Longinotti, Kynan Eng, and Tobi Delbruck. DHP19: Dynamic vision sensor 3d human pose dataset. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1695–1704. IEEE. doi: 10.1109/cvprw.2019.00217. URL https://ieeexplore.ieee.org/document/9025364/.

[52] Quan Kong, Ziming Wu, Ziwei Deng, Martin Klinkigt, Bin Tong, and Tomokazu Murakami. MMAct: A large-scale dataset for cross modal human action understanding. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8657–8666. IEEE. doi: 10.1109/iccv.2019.00875. URL https://ieeexplore.ieee.org/document/9009579/.

[53] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. OpenFace 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 59–66. IEEE. doi: 10.1109/fg.2018.00019. URL https://ieeexplore.ieee.org/document/8373812/.

[54] Lichen Wang, Bin Sun, Joseph Robinson, Taotao Jing, and Yun Fu. EV-action: Electromyography-vision multi-modal action dataset. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 160–167, . doi: 10.1109/FG47880.2020.00018. URL https://ieeexplore.ieee.org/document/9320160/.

[55] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701, 2019.

[56] Yizhak Ben-Shabat, Xin Yu, Fatemeh Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The IKEA ASM dataset: Understanding people assembling furniture through actions, objects and pose. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 846–858. IEEE. doi: 10.1109/wacv48630.2021.00089. URL https://ieeexplore.ieee.org/document/9423070/.

[57] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5253–5263, 2020.

[58] M Teja Kiran Kumar, PVV Kishore, Boddapati Taraka Phani Madhav, D Anil Kumar, N Sasi Kala, K Praveen Kumar Rao, and B Prasad. Can skeletal joint positional ordering influence action recognition on spectrally graded cnns: A perspective on achieving joint order independent learning. *IEEE Access*, 9:139611–139626, 2021.

[59] Tianjiao Li, Jun Liu, Wei Zhang, Yun Ni, Wenqian Wang, and Zhiheng Li. UAV-human: A large benchmark for human behavior understanding with unmanned aerial vehicles. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16261–16270. IEEE, . doi: 10.1109/cvpr46437.2021.01600. URL https://ieeexplore.ieee.org/document/9578530/.

[60] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2353–2362. IEEE, 2017.

[61] Jingyu Zhao, Feiqing Huang, Jia Lv, Yanjie Duan, Zhen Qin, Guodong Li, and Guangjian Tian. Do rnn and lstm have long memory? In *International Conference on Machine Learning*, pages 11365–11375. PMLR, 2020.

[62] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3575–3584, 2019.

[63] Jongmin Yu, Yongsang Yoon, and Moongu Jeon. Predictively encoded graph convolutional network for noise-robust skeleton-based action recognition, . URL http://arxiv.org/abs/2003.07514.

[64] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1227–1236, 2019.

[65] Jianyang Xie, Yitian Zhao, Yanda Meng, He Zhao, Anh Nguyen, and Yalin Zheng. Are spatial-temporal graph convolution networks for human action recognition over-parameterized? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24309–24319, 2025.

[66] Jinze Huo. *Skeleton-based action recognition by deep learning*. PhD thesis, Loughborough University, 2024.

[67] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. PYSKL: Towards good practices for skeleton action recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, pages 7351–7354. Association for Computing Machinery, . ISBN 978-1-4503-9203-7. doi: 10.1145/3503161.3548546. URL https://dl.acm.org/doi/10.1145/3503161.3548546.

[68] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. DG-STGCN: Dynamic spatial-temporal modeling for skeleton-based action recognition, . URL http://arxiv.org/abs/2210.05895.

[69] Shijie Li, Jinhui Yi, Yazan Abu Farha, and Juergen Gall. Pose refinement graph convolutional network for skeleton-based action recognition. 6(2):1028–1035, . ISSN 2377-3766. doi: 10.1109/LRA.2021.3056361. URL https://ieeexplore.ieee.org/document/9345415/.

[70] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Skeleton-based action recognition via spatial and temporal transformer networks. 208-209:103219. ISSN 1077-3142. doi: 10.1016/j.cviu.2021.103219. URL https://linkinghub.elsevier.com/retrieve/pii/S1077314221000631. Publisher: Elsevier BV.

[71] Hongda Liu, Yunfan Liu, Min Ren, Hao Wang, Yunlong Wang, and Zhenan Sun. Revealing key details to see differences: A novel prototypical perspective for skeleton-based action recognition. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29248–29257, 2025.

[72] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 140–149. IEEE, . ISBN 978-1-7281-7168-5. doi: 10.1109/CVPR42600.2020.00022. URL `https://ieeexplore.ieee.org/document/9156556/`.

[73] Xiaolu Ding, Kai Yang, and Wai Chen. An attention-enhanced recurrent graph convolutional network for skeleton-based action recognition. In *Proceedings of the 2019 2nd International Conference on Signal Processing and Machine Learning*, pages 79–84. ACM, . doi: 10.1145/3372806.3372814. URL `https://dl.acm.org/doi/10.1145/3372806.3372814`.

[74] Guyue Hu, Bo Cui, and Shan Yu. Skeleton-based action recognition with synchronous local and non-local spatio-temporal learning and frequency attention. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1216–1221. doi: 10.1109/ICME.2019.00212. URL `https://ieeexplore.ieee.org/abstract/document/8784951`. ISSN: 1945-788X.

[75] Bin Li, Xi Li, Zhongfei Zhang, and Fei Wu. Spatio-temporal graph routing for skeleton-based action recognition. 33(1):8561–8568, . ISSN 2374-3468. doi: 10.1609/aaai.v33i01.33018561. URL `https://ojs.aaai.org/index.php/AAAI/article/view/4875`. Number: 01.

[76] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3595–3603, 2019.

[77] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7904–7913. IEEE, . doi: 10.1109/cvpr.2019.00810. URL `https://ieeexplore.ieee.org/document/8954160/`.

[78] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12026–12035, 2019.

[79] Raymond Yeh, Yuan-Ting Hu, and Alexander Schwing. Chirality nets for human pose regression. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., . URL `https://proceedings.neurips.cc/paper/2019/hash/1f88c7c5d7d94ae08bd752aa3d82108b-Abstract.html`.

[80] Wei Peng, Xiaopeng Hong, Haoyu Chen, and Guoying Zhao. Learning graph convolutional network for skeleton-based human action recognition by neural searching. 34(3):2669–2676. ISSN 2374-3468. doi: 10.1609/aaai.v34i03.5652. URL `https://ojs.aaai.org/index.php/AAAI/article/view/5652`. Number: 03.

[81] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. 29:9532–9545, . ISSN 1941-0042. doi: 10.1109/TIP.2020.3028207. URL `https://ieeexplore.ieee.org/abstract/document/9219176`.

[82] Fanfan Ye, Shiliang Pu, Qiaoyong Zhong, Chao Li, Di Xie, and Huiming Tang. Dynamic GCN: Context-enriched topology learning for skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 55–63. ACM. doi: 10.1145/3394171.3413941. URL `https://dl.acm.org/doi/10.1145/3394171.3413941`.

[83] Xikun Zhang, Chang Xu, and Dacheng Tao. Context aware graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14333–14342, 2020.

[84] Yongsang Yoon, Jongmin Yu, and Moongu Jeon. Predictively encoded graph convolutional network for noise-robust skeleton-based action recognition. 52(3):2317–2331. ISSN 1573-7497. doi: 10.1007/s10489-021-02487-z. URL `https://doi.org/10.1007/s10489-021-02487-z`.

[85] Hua-Bin Chen, Zhen Li, Pan Fu, Zhen-Liang Ni, and Gui-Bin Bian. Spatio-temporal causal transformer for multi-grained surgical phase recognition. In *2022 44th annual international conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1663–1666. IEEE, 2022.

[86] Tailin Chen, Desen Zhou, Jian Wang, Shidong Wang, Yu Guan, Xuming He, and Errui Ding. Learning multi-granular spatio-temporal graph network for skeleton-based action recognition. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, pages 4334–4342. Association for Computing Machinery, . ISBN 978-1-4503-8651-7. doi: 10.1145/3474085.3475574. URL `https://dl.acm.org/doi/10.1145/3474085.3475574`.

[87] Yuya Obinata and Takuma Yamamoto. Temporal extension module for skeleton-based action recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 534–540. doi: 10.1109/ICPR48806.2021.9412113. URL `https://ieeexplore.ieee.org/abstract/document/9412113`. ISSN: 1051-4651.

[88] Jun Xie, Qiguang Miao, Ruyi Liu, Wentian Xin, Lei Tang, Sheng Zhong, and Xuesong Gao. Attention adjacency matrix based graph convolutional networks for skeleton-based action recognition. 440:230–239, . ISSN 0925-2312. doi: 10.1016/j.neucom.2021.02.001. URL `https://linkinghub.elsevier.com/retrieve/pii/S0925231221002101`. Publisher: Elsevier BV.

[89] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2969–2978, 2022.

[90] Lipeng Ke, Kuan-Chuan Peng, and Siwei Lyu. Towards to-a-t spatio-temporal focus for skeleton-based action recognition. 36(1):1131–1139. ISSN 2374-3468. doi: 10.1609/aaai.v36i1.19998. URL `https://ojs.aaai.org/index.php/AAAI/article/view/19998`. Number: 1.

[91] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction. 44(6):3316–3333, . ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3053765. URL `https://ieeexplore.ieee.org/abstract/document/9334430`.

[92] Dong Yang, Monica Mengqi Li, Hong Fu, Jicong Fan, Zhao Zhang, and Howard Leung. Unifying graph embedding features with graph convolutional networks for skeleton-based action recognition. URL `http://arxiv.org/abs/2003.03007`.

[93] Ryo Hachiuma, Fumiaki Sato, and Taiki Sekii. Unified keypoint-based action recognition framework via structured keypoint pooling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22962–22971, 2023.

[94] Yanan Liu, Hao Zhang, Yanqiu Li, Kangjian He, and Dan Xu. Skeleton-based human action recognition via large-kernel attention graph convolutional network. 29(5):2575–2585, . ISSN 1941-0506. doi: 10.1109/TVCG.2023.3247075. URL `https://ieeexplore.ieee.org/abstract/document/10049725`.

[95] Shu-Bo Zhou, Ran-Ran Chen, Xue-Qin Jiang, and Feng Pan. 2s-GATCN: Two-stream graph attentional convolutional networks for skeleton-based action recognition. 12(7):1711. ISSN 2079-9292. doi: 10.3390/electronics12071711. URL https://www.mdpi.com/2079-9292/12/7/1711. Publisher: MDPI AG.

[96] Huiyan Han, Hongwei Zeng, Liqun Kuang, Xie Han, and Hongxin Xue. A human activity recognition method based on vision transformer. 14:15310. ISSN 2045-2322. doi: 10.1038/s41598-024-65850-3. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11222487/.

[97] Jianyang Xie, Yanda Meng, Yitian Zhao, Anh Nguyen, Xiaoyun Yang, and Yalin Zheng. Dynamic semantic-based spatial graph convolution network for skeleton-based human action recognition. 38(6):6225–6233, . ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v38i6.28440. URL https://ojs.aaai.org/index.php/AAAI/article/view/28440. Publisher: Association for the Advancement of Artificial Intelligence (AAAI).

[98] Dong Chen, Mingdong Chen, Peisong Wu, Mengtao Wu, Tao Zhang, and Chuanqi Li. Two-stream spatio-temporal GCN-transformer networks for skeleton-based action recognition. 15(1), . ISSN 2045-2322. doi: 10.1038/s41598-025-87752-8. URL https://www.nature.com/articles/s41598-025-87752-8. Publisher: Springer Science and Business Media LLC.

[99] Zhiyun Zheng, Qilong Yuan, Huaizhu Zhang, Yizhou Wang, and Junfeng Wang. Lightweight multiscale spatio-temporal graph convolutional network for skeleton-based action recognition. 8(2):310–325. ISSN 2097-406X. doi: 10.26599/BDMA.2024.9020095. URL https://ieeexplore.ieee.org/document/10856896/.

[100] Andru P. Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel de Mathelin, and Nicolas Padoy. EndoNet: A deep architecture for recognition tasks on laparoscopic videos. 36(1):86–97. ISSN 1558-254X. doi: 10.1109/TMI.2016.2593957.

[101] D.T. Tran, R. Sakurai, and J.-H. Lee. Integration of a topic probability distribution into surgical phase estimation with a hidden markov model. In *Annu. Conf. IEEE Industrial Electron. Soc., IECON*, pages 4766–4771. Institute of Electrical and Electronics Engineers Inc. ISBN 978-147991762-4 (ISBN). doi: 10.1109/IECON.2015.7392845. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-84973129414&doi=10.1109%2fIECON.2015.7392845&partnerID=40&md5=cf08df302927f9797bc5216cbdb8f113. Journal Abbreviation: Annu. Conf. IEEE Industrial Electron. Soc., IECON.

[102] Olga Dergachyova, David Bouget, Arnaud Huaulmé, Xavier Morandi, and Pierre Jannin. Automatic data-driven real-time segmentation and recognition of surgical workflow. 11(6):1081–1089. ISSN 1861-6429. doi: 10.1007/s11548-016-1371-x.

[103] Danyal Z Khan, Imanol Luengo, Santiago Barbarisi, Carole Addis, Lucy Culshaw, Neil L Dorward, Pinja Haikka, Abhiney Jain, Karen Kerr, Chan Hee Koh, et al. Automated operative workflow analysis of endoscopic pituitary surgery using machine learning: development and preclinical evaluation (ideal stage 0). *Journal of Neurosurgery*, 137(1):51–58, 2021.

[104] D.A. Hashimoto, G. Rosman, E.R. Witkowski, C. Stafford, A.J. Navarette-Welton, D.W. Rattner, K.D. Lillemoe, D.L. Rus, and O.R. Meireles. Computer vision analysis of intraoperative video: Automated recognition of operative steps in laparoscopic sleeve gastrectomy. *Annals of Surgery*, 270(3):414–421, 2019. ISSN 00034932 (ISSN). doi: 10.1097/SLA.0000000000003460. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85070961171&doi=10.1097%2fSLA.0000000000003460&partnerID=40&md5=87f7923bc4cb0780c788478f25e477ff. Publisher: Lippincott Williams and Wilkins.

[105] Hirenkumar Nakawala, Roberto Bianchi, Laura Erica Pescatori, Ottavio De Cobelli, Giancarlo Ferrigno, and Elena De Momi. “deep-onto” network for surgical workflow and context recognition. *International journal of computer assisted radiology and surgery*, 14(4):685–696, 2019.

[106] L. Bastian, T. Czempiel, C. Heiliger, K. Karcz, U. Eck, B. Busam, and N. Navab. Know your sensors — a modality study for surgical action classification. 11(4):1113–1121. ISSN 21681163 (ISSN). doi: 10.1080/21681163.2022.2152377. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85144280383&doi=10.1080%2f21681163.2022.2152377&partnerID=40&md5=cc994809b3ac908c2622eacbf47a957b. Publisher: Taylor and Francis Ltd.

[107] Y. Ding, J. Fan, K. Pang, H. Li, T. Fu, H. Song, L. Chen, and J. Yang. Surgical workflow recognition using two-stream mixed convolution network. In *Proc. - Int. Conf. Adv. Electron. Mater., Comput. Softw. Eng., AEMCSE*, pages 264–269. Institute of Electrical and Electronics Engineers Inc., . ISBN 978-172818143-1 (ISBN). doi: 10.1109/AEMCSE50948.2020.00064. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85088642870&doi=10.1109%2fAEMCSE50948.2020.00064&partnerID=40&md5=cea98dc0e4266d32952864a980ea64c1. Journal Abbreviation: Proc. - Int. Conf. Adv. Electron. Mater., Comput. Softw. Eng., AEMCSE.

[108] Bokai Zhang, Amer Ghanem, Alexander Simes, Henry Choi, and Andrew Yoo. Surgical workflow recognition with 3dcnn for sleeve gastrectomy. *International Journal of Computer Assisted Radiology and Surgery*, 16(11):2029–2036, 2021.

[109] G. De Rossi, S. Roin, F. Setti, and R. Muradore. A multi-modal learning system for on-line surgical action segmentation. In *Int. Symp. Med. Robot., ISMR*, pages 132–138. Institute of Electrical and Electronics Engineers Inc. ISBN 978-172815488-6 (ISBN). doi: 10.1109/ISMR48331.2020.9312950. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85100271447&doi=10.1109%2fISMR48331.2020.9312950&partnerID=40&md5=d5babb56f54189a82404909aad2c9754. Journal Abbreviation: Int. Symp. Med. Robot., ISMR.

[110] Tomer Golany, Amit Aides, Daniel Freedman, Nadav Rabani, Yun Liu, Ehud Rivlin, Greg S Corrado, Yossi Matias, Wisam Khoury, Hanoch Kashtan, et al. Artificial intelligence for phase recognition in complex laparoscopic cholecystectomy. *Surgical Endoscopy*, 36(12):9215–9223, 2022.

[111] Masashi Takeuchi, Hirofumi Kawakubo, Kosuke Saito, Yusuke Maeda, Satoru Matsuda, Kazumasa Fukuda, Rieko Nakamura, and Yuko Kitagawa. Automated surgical-phase recognition for robot-assisted minimally invasive esophagectomy using artificial intelligence. *Annals of Surgical Oncology*, 29(11):6847–6855, 2022.

[112] Yuhao Zhai, Zhen Chen, Zhi Zheng, Xi Wang, Xiaosheng Yan, Xiaoye Liu, Jie Yin, Jinqiao Wang, and Jun Zhang. Artificial intelligence for automatic surgical phase recognition of laparoscopic gastrectomy in gastric cancer. *International Journal of Computer Assisted Radiology and Surgery*, 19(2):345–353, 2024.

[113] Ekamjit S Deol, Matthew K Tollefson, Alenka Antolin, Maya Zohar, Omri Bar, Danielle Ben-Ayoun, Lance A Mynderse, Derek J Lomas, Ross A Avant, Adam R Miller, et al. Automated surgical step recognition in transurethral bladder tumor resection using artificial intelligence: transfer learning across surgical modalities. *Frontiers in Artificial Intelligence*, 7:1375482, 2024.

[114] Y. Tian, S. Paheding, E. Azimi, and E.-J. Lee. Exploring action recognition in endoscopy video datasets. In Kehtarnavaz N. and Shirvaikar M.V., editors, *Proc SPIE Int Soc Opt Eng*, volume 13034. SPIE. ISBN 0277786X (ISSN); 978-151067386-1 (ISBN). doi: 10.1117/12.3014345. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85197472731&doi=10.1117%2f12.3014345&partnerID=40&md5=74163d16295118a83b9fd495366d55f5. Journal Abbreviation: Proc SPIE Int Soc Opt Eng.

[115] Hilde Kuehne, Alexander Richard, and Juergen Gall. A hybrid RNN-HMM approach for weakly supervised temporal action segmentation. 42(4):765–779. ISSN 1939-3539. doi: 10.1109/TPAMI.2018.2884469. URL https://ieeexplore.ieee.org/document/8585084/.

[116] Sanat Ramesh, Diego Dall'Alba, Cristians Gonzalez, Tong Yu, Pietro Mascagni, Didier Mutter, Jacques Marescaux, Paolo Fiorini, and Nicolas Padoy. Weakly supervised temporal convolutional networks for fine-grained surgical activity recognition. 42(9):2592–2602, . ISSN 1558-254X. doi: 10.1109/TMI.2023.3262847.

[117] Annetje CP Guédon, Senna EP Meij, Karim NMMH Osman, Helena A Kloosterman, Karlijn J van Stralen, Matthijs CM Grimbergen, Quirijn AJ Eijsbouts, John J van den Dobbelsteen, and Andru P Twinanda. Deep learning for surgical phase recognition using endoscopic videos. *Surgical endoscopy*, 35(11):6150–6157, 2021.

[118] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, volume 11207, pages 297–313. Springer International Publishing, . ISBN 978-3-030-01218-2 978-3-030-01219-9. doi: 10.1007/978-3-030-01219-9_18. URL https://link.springer.com/10.1007/978-3-030-01219-9_18. Series Title: Lecture Notes in Computer Science.

[119] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

[120] Irene Rivas-Blanco, Carmen López-Casado, Juan M. Herrera-López, José Cabrera-Villa, and Carlos J. Pérez-del Pulgar. Instrument detection and descriptive gesture segmentation on a robotic surgical maneuvers dataset. 14(9):3701. ISSN 2076-3417. doi: 10.3390/app14093701. URL https://www.mdpi.com/2076-3417/14/9/3701. Publisher: Multidisciplinary Digital Publishing Institute.

[121] Jinglu Zhang, Yinyu Nie, Yao Lyu, Hailin Li, Jian Chang, Xiaosong Yang, and Jian Jun Zhang. Symmetric dilated convolution for surgical gesture recognition. In Anne L. Martel, Purang Abolmaesumi, Danail Stoyanov, Diana Mateus, Maria A. Zuluaga, S. Kevin Zhou, Daniel Racoceanu, and Leo Joskowicz, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, pages 409–418. Springer International Publishing, . ISBN 978-3-030-59716-0. doi: 10.1007/978-3-030-59716-0_39.

[122] M Takeuchi, T Collins, A Ndagijimana, H Kawakubo, Y Kitagawa, J Marescaux, D Mutter, S Perretta, A Hostettler, and B Dallemagne. Automatic surgical phase recognition in laparoscopic inguinal hernia repair with artificial intelligence. *Hernia*, 26(6):1669–1678, 2022.

[123] Monica Ortenzi, Judith Rapoport Ferman, Alenka Antolin, Omri Bar, Maya Zohar, Ori Perry, Dotan Asselmann, and Tamir Wolf. A novel high accuracy model for automatic surgical workflow recognition using artificial intelligence in laparoscopic totally extraperitoneal inguinal hernia repair (tep). *Surgical Endoscopy*, 37(11):8818–8828, 2023.

[124] Ke Cheng, Jiaying You, Shangdi Wu, Zixin Chen, Zijian Zhou, Jingye Guan, Bing Peng, and Xin Wang. Artificial intelligence-based automated laparoscopic cholecystectomy surgical phase recognition and analysis. *Surgical endoscopy*, 36(5):3160–3168, 2022.

[125] Felix Yu, Gianluca Silva Croso, Tae Soo Kim, Ziang Song, Felix Parker, Gregory D. Hager, Austin Reiter, S. Swaroop Vedula, Haider Ali, and Shameema Sikder. Assessment of automated identification of phases in videos of cataract surgery using machine learning and deep learning techniques. 2(4):e191860, . ISSN 2574-3805. doi: 10.1001/jamanetworkopen.2019.1860.

[126] Thomas M. Ward, Daniel A. Hashimoto, Yutong Ban, David W. Rattner, Haruhiro Inoue, Keith D. Lillemoe, Daniela L. Rus, Guy Rosman, and Ozanan R. Meireles. Automated operative phase identification in peroral endoscopic myotomy. 35(7):4008–4015. ISSN 1432-2218. doi: 10.1007/s00464-020-07833-9.

[127] B. Namazi, G. Sankaranarayanan, and V. Devarajan. Automatic detection of surgical phases in laparoscopic videos. In Arabnia H.R., de la Fuente D., Kozerenko E.B., Olivas J.A., and Tinetti F.G., editors, *World Congr. Comput. Sci., Comput. Eng. Appl. Comput., CSCE - Proc. Int. Conf. Artif. Intell., ICAI*, pages 124–130. CSREA Press. ISBN 1601324804 (ISBN); 978-160132480-1 (ISBN). URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85068405831&partnerID=40&md5=aa3383c85ffa034ce3deed52c9ac9a2b. Journal Abbreviation: World Congr. Comput. Sci., Comput. Eng. Appl. Comput., CSCE - Proc. Int. Conf. Artif. Intell., ICAI.

[128] P.S. Vadali, B. Jayam, P. Deepika, M. Rao, and V. Vazhayil. Computer assisted phase recognition of micro-neurosurgical intraoperative videos. In Ochoa-Ruiz G., Grisan E., Ali S., Sicilia R., Santamaria L.P., Kane B., Daul C., Ante G.S., and Gonzalez A.R., editors, *Proc. IEEE Symp. Comput.-Based Med. Syst.*, pages 456–460. Institute of Electrical and Electronics Engineers Inc. ISBN 10637125 (ISSN); 979-835038472-7 (ISBN). doi: 10.1109/CBMS61543.2024.00081. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85200505655&doi=10.1109%2fCBMS61543.2024.00081&partnerID=40&md5=71eda43b54cf4583177bef112ae1290e. Journal Abbreviation: Proc. IEEE Symp. Comput.-Based Med. Syst.

[129] K. Kawamura, R. Ebata, R. Nakamura, and N. Otori. Improving situation recognition using endoscopic videos and navigation information for endoscopic sinus surgery. 18(1):9–16. ISSN 18616410 (ISSN). doi: 10.1007/s11548-022-02754-5. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85138742308&doi=10.1007%2fs11548-022-02754-5&partnerID=40&md5=1c166695eaf0b4a4b9956c2f133eb448. Publisher: Springer Science and Business Media Deutschland GmbH.

[130] Aneeq Zia, Liheng Guo, Linlin Zhou, Irfan Essa, and Anthony Jarc. Novel evaluation of surgical activity recognition models using task-based efficiency metrics. 14(12):2155–2163. ISSN 1861-6429. doi: 10.1007/s11548-019-02025-w.

[131] H.-H. Yeh, A.M. Jain, O. Fox, and S.Y. Wang. Phacotrainer: A multicenter study of deep learning for activity recognition in cataract surgical videos. 10(13), . ISSN 21642591 (ISSN). doi: 10.1167/TVST.10.13.23. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85121055474&doi=10.1167%2fTVST.10.13.23&partnerID=40&md5=9d03a83ace78c39ea004a8b44b9fbf5a. Publisher: Association for Research in Vision and Ophthalmology Inc.

[132] Xueying Shi, Yueming Jin, Qi Dou, and Pheng-Ann Heng. Semi-supervised learning with progressive unlabeled data excavation for label-efficient surgical workflow recognition. *Medical Image Analysis*, 73:102158, 2021.

[133] Y. Chen, Q.L. Sun, and K. Zhong. Semi-supervised spatio-temporal CNN for recognition of surgical workflow. 2018(1), . ISSN 16875176 (ISSN). doi: 10.1186/s13640-018-0316-4. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85052201089&doi=10.1186%2fs13640-018-0316-4&partnerID=40&md5=e038959b5a7cb52521a5e184fc1ecac5. Publisher: Springer International Publishing.

[134] Y. Li, Y. Li, W. He, W. Shi, T. Wang, and Y. Li. SE-OHFM: A surgical phase recognition network with SE attention module. In *Int. Conf. Electron. Inf. Eng. Comput. Sci., EIECS*, pages 608–611. Institute of Electrical and Electronics Engineers Inc., . ISBN 978-166541674-0 (ISBN). doi: 10.1109/EIECS53707.2021.9587961. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85119417457&doi=10.1109%2fEIECS53707.2021.9587961&partnerID=40&md5=8d5710223ec4a8569a97df00b095ea86. Journal Abbreviation: Int. Conf. Electron. Inf. Eng. Comput. Sci., EIECS.

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

[136] Lixin Fang, Lei Mou, Yuanyuan Gu, Yan Hu, Bang Chen, Xu Chen, Yang Wang, Jiang Liu, and Yitian Zhao. Global–local multi-stage temporal convolutional network for cataract surgery phase recognition. *BioMedical Engineering OnLine*, 21(1):82, 2022.

[137] B. Zhang, A. Ghanem, A. Simes, H. Choi, A. Yoo, and A. Min. SWNet: Surgical workflow recognition with deep convolutional network. In Heinrich M., Dou Q., de Bruijne M., de Bruijne M., Lellmann J., Schlaefer A., and Ernst F., editors, *Proc. Mach. Learn. Res.*, volume 143, pages 838–852. ML Research Press, . ISBN 26403498 (ISSN). URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85162889528&partnerID=40&md5=9b8323ea523ff9c718a98e87d998de19. Journal Abbreviation: Proc. Mach. Learn. Res.

[138] Sanat Ramesh, Diego Dall'Alba, Cristians Gonzalez, Tong Yu, Pietro Mascagni, Didier Mutter, Jacques Marescaux, Paolo Fiorini, and Nicolas Padoy. Multi-task temporal convolutional networks for joint recognition of surgical phases and steps in gastric bypass procedures. 16(7):1111–1119, . ISSN 1861-6429. doi: 10.1007/s11548-021-02388-z.

[139] Wenxi Yue, Hongen Liao, Yong Xia, Vincent Lam, Jiebo Luo, and Zhiyong Wang. Cascade multi-level transformer network for surgical workflow analysis. *IEEE transactions on medical imaging*, 42(10):2817–2831, 2023.

[140] B. Zhang, J. Meng, B. Cheng, D. Biskup, S. Petculescu, and A. Chapman. Friends across time: Multi-scale action segmentation transformer for surgical phase recognition. In *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*. Institute of Electrical and Electronics Engineers Inc., . ISBN 1557170X (ISSN); 979-835037149-9 (ISBN). doi: 10.1109/EMBC53108.2024.10782887. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85215013843&doi=10.1109%2fEMBC53108.2024.10782887&partnerID=40&md5=21d82026979d4a5b180ac1b9ea74fb15. Journal Abbreviation: Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS.

[141] Yang Liu, Maxence Boels, Luis C Garcia-Peraza-Herrera, Tom Vercauteren, Prokar Dasgupta, Alejandro Granados, and Sebastien Ourselin. Lovit: Long video transformer for surgical phase recognition. *Medical Image Analysis*, 99:103366, 2025.

[142] Z. Chen, Y. Zhai, J. Zhang, and J. Wang. Temporal action-aware network with sequence regularization for phase recognition. In Jiang X., Wang H., Alhajj R., Hu X., Engel F., Mahmud M., Pisanti N., Cui X., and Song H., editors, *Proc. - IEEE Int. Conf. Bioinform. Biomed., BIBM*, pages 1836–1841. Institute of Electrical and Electronics Engineers Inc., . ISBN 979-835033748-8 (ISBN). doi: 10.1109/BIBM58861.2023.10385308. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85184882371&doi=10.1109%2fBIBM58861.2023.10385308&partnerID=40&md5=ec36a3ada284de4c53dd120562f2b1f5. Journal Abbreviation: Proc. - IEEE Int. Conf. Bioinform. Biomed., BIBM.

[143] Rong Tao, Xiaoyang Zou, and Guoyan Zheng. Last: Latent space-constrained transformers for automatic surgical phase recognition and tool presence detection. *IEEE Transactions on Medical Imaging*, 42(11):3256–3268, 2023.

[144] Y. Liu, J. Huo, J. Peng, R. Sparks, P. Dasgupta, A. Granados, and S. Ourselin. SKiT: a fast key information video transformer for online surgical phase recognition. In *Proc IEEE Int Conf Comput Vision*, pages 21017–21027. Institute of Electrical and Electronics Engineers Inc., . ISBN 15505499 (ISSN); 979-835030718-4 (ISBN). doi: 10.1109/ICCV51070.2023.01927. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85185867869&doi=10.1109%2fICCV51070.2023.01927&partnerID=40&md5=3adca666dbe342871f40fa4f599a344c. Journal Abbreviation: Proc IEEE Int Conf Comput Vision.

[145] J. Long, J. Hong, Z. Wang, T. Chen, Y. Chen, and L. Yang. SPHASE: Multi-modal and multi-branch surgical phase segmentation framework based on temporal convolutional network. In Jiang X., Wang H., Alhajj R., Hu X., Engel F., Mahmud M., Pisanti N., Cui X., and Song H., editors, *Proc. - IEEE Int. Conf. Bioinform. Biomed., BIBM*, pages 586–593. Institute of Electrical and Electronics Engineers Inc. ISBN 979-835033748-8 (ISBN). doi: 10.1109/BIBM58861.2023.10385579. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85184892501&doi=10.1109%2fBIBM58861.2023.10385579&partnerID=40&md5=d07fdbdb8616e5651a39bdf36b7239f8. Journal Abbreviation: Proc. - IEEE Int. Conf. Bioinform. Biomed., BIBM.

[146] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly, 2019.

[147] B. Zhang, A. Fung, M. Torabi, J. Barker, G. Foley, R. Abukhalil, M.L. Gaddis, and S. Petculescu. C-ECT: Online surgical phase recognition with cross-enhancement causal transformer. In *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn.*, volume 2023-April. IEEE Computer Society, . ISBN 19457928 (ISSN); 978-166547358-3 (ISBN). doi: 10.1109/ISBI53787.2023.10230841. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85172153992&doi=10.1109%2fISBI53787.2023.10230841&partnerID=40&md5=15284038a776cdd429ae44e0134e99c2. Journal Abbreviation: IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn.

[148] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3156–3165. IEEE. ISBN 978-1-6654-0191-3. doi: 10.1109/ICCVW54120.2021.00355. URL https://ieeexplore.ieee.org/document/9607406/.

[149] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, volume 9908, pages 137–153. Springer International Publishing, 2016. ISBN 978-3-319-46492-3 978-3-319-46493-0. doi: 10.1007/978-3-319-46493-0_9. URL http://link.springer.com/10.1007/978-3-319-46493-0_9. Series Title: Lecture Notes in Computer Science.

[150] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 563–579, 2018.

[151] Zihao Jiang and Yidong Li. Weakly supervised temporal action localization through segment contrastive learning. In Biao Luo, Long Cheng, Zheng-Guang Wu, Hongyi Li, and Chaojie Li, editors, *Neural Information Processing*, pages 228–243. Springer Nature, 2024. ISBN 978-981-99-8141-0. doi: 10.1007/978-981-99-8141-0_18.

[152] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Comput. Vis. Image Underst.*, 163:78–89, 2017. ISSN 1077-3142. doi: 10.1016/j.cviu.2017.06.004. URL https://doi.org/10.1016/j.cviu.2017.06.004.

[153] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 628–643. Springer International Publishing, 2014. ISBN 978-3-319-10602-1. doi: 10.1007/978-3-319-10602-1_41.

[154] Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5987–5996. IEEE, 2018. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00627. URL `https://ieeexplore.ieee.org/document/8578725/`.

[155] Nicolas Aziere and Sinisa Todorovic. Multistage temporal convolution transformer for action segmentation. *Image and Vision Computing*, 128:104567, 2022. ISSN 02628856. doi: 10.1016/j.imavis.2022.104567. URL `https://linkinghub.elsevier.com/retrieve/pii/S0262885622001962`.

[156] Zijia Lu and Ehsan Elhamifar. Weakly-supervised action segmentation and alignment via transcript-aware union-of-subspaces learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8065–8075. IEEE, 2021. ISBN 978-1-6654-2812-5. doi: 10.1109/ICCV48922.2021.00798. URL `https://ieeexplore.ieee.org/document/9710443/`.

[157] Xiaobin Chang, Frederick Tung, and Greg Mori. Learning discriminative prototypes with dynamic time warping. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8391–8400. IEEE, 2021. ISBN 978-1-6654-4509-2. doi: 10.1109/CVPR46437.2021.00829. URL `https://ieeexplore.ieee.org/document/9577531/`.

[158] Zijia Lu and Ehsan Elhamifar. Set-supervised action learning in procedural task videos via pairwise order consistency. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19871–19881. IEEE, 2022. ISBN 978-1-6654-6946-3. doi: 10.1109/CVPR52688.2022.01928. URL `https://ieeexplore.ieee.org/document/9879753/`.

[159] Hassan Al Hajj, Mathieu Lamard, Pierre-Henri Conze, Soumali Roychowdhury, Xiaowei Hu, Gabija Maršalkaitė, Odysseas Zisimopoulos, Muneer Ahmad Dedmari, Fenqiang Zhao, Jonas Prellberg, Manish Sahu, Adrian Galdran, Teresa Araújo, Duc My Vo, Chandan Panda, Navdeep Dahiya, Satoshi Kondo, Zhengbing Bian, Arash Vahdat, Jonas Bialopetravičius, Evangello Flouty, Chenhui Qiu, Sabrina Dill, Anirban Mukhopadhyay, Pedro Costa, Guilherme Aresta, Senthil Ramamurthy, Sang-Woong Lee, Aurélio Campilho, Stefan Zachow, Shunren Xia, Sailesh Conjeti, Danail Stoyanov, Jogundas Armaitis, Pheng-Ann Heng, William G. Macready, Béatrice Cochener, and Gwenolé Quellec. CATARACTS: Challenge on automatic tool annotation for cataRACT surgery. *Medical Image Analysis*, 52:24–41, 2019. ISSN 1361-8415. doi: 10.1016/j.media.2018.11.008. URL `https://www.sciencedirect.com/science/article/pii/S136184151830865X`.

[160] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Changxin Gao, and Nong Sang. Self-supervised learning for semi-supervised temporal action proposal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1905–1914, 2021.

[161] Unaiza Ahsan, Rishi Madhok, and Irfan Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 179–189. IEEE, 2019.

[162] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-supervised spatiotemporal feature learning via video rotation prediction, 2019. URL `http://arxiv.org/abs/1811.11387`.

[163] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9922–9931, 2020.

[164] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9640–9649, 2021.

[165] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. 35:10078–10093. URL `https://proceedings.neurips.cc/paper_files/paper/2022/hash/416f9cb3276121c42eebb86352a4354a-Abstract-Conference.html`.

[166] Yuliya Vybornova, Maxim Aleshin, Svetlana Illarionova, Ilya Novikov, Dmitrii Shadrin, Artem Nikonorov, and Evgeny Burnaev. Self-supervised learning for temporal action segmentation in industrial and manufacturing videos. *IEEE Access*, 13:39650–39665, 2025. ISSN 2169-3536. doi: 10.1109/ACCESS.2025.3545768. URL `https://ieeexplore.ieee.org/document/10906499/`.

[167] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023.

[168] Fida Mohammad Thoker, Hazel Doughty, Piyush Bagad, and Cees G. M. Snoek. How severe is benchmark-sensitivity in video self-supervised learning? In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 632–652. Springer Nature Switzerland. ISBN 978-3-031-19830-4. doi: 10.1007/978-3-031-19830-4_36.

[169] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8368–8376, 2018.

[170] Gaurav Yengera, Didier Mutter, Jacques Marescaux, and Nicolas Padoy. Less is more: Surgical phase recognition with less annotations through self-supervised pre-training of CNN-LSTM networks, 2018. URL `http://arxiv.org/abs/1805.08569`.

[171] Xingjian Gu, Supeng Yu, Fen Huang, Shougang Ren, and Chengcheng Fan. Consistency self-training semi-supervised method for road extraction from remote sensing images. *Remote Sensing*, 16(21):3945, 2024. ISSN 2072-4292. doi: 10.3390/rs16213945. URL `https://www.mdpi.com/2072-4292/16/21/3945`. Number: 21 Publisher: Multidisciplinary Digital Publishing Institute.

[172] Wei Chih Hung, Yi Hsuan Tsai, Yan Ting Liou, Yen Yu Lin, and Ming Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. In *29th British Machine Vision Conference, BMVC 2018*, 2019.

[173] Shasvat Desai and Debasmita Ghose. Active learning for improved semi-supervised semantic segmentation in satellite images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 553–563, 2022.

[174] Xinpeng Ding, Nannan Wang, Xinbo Gao, Jie Li, Xiaoyu Wang, and Tongliang Liu. KFC: An efficient framework for semi-supervised temporal action localization. *IEEE Transactions on Image Processing*, 30:6869–6878, 2021. ISSN 1941-0042. doi: 10.1109/TIP.2021.3099407. URL `https://ieeexplore.ieee.org/document/9500051/`.

[175] Jingwei Ji, Kaidi Cao, and Juan Carlos Niebles. Learning temporal action proposals with fewer labels. *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[176] Weining Wang, Tianwei Lin, Dongliang He, Fu Li, Shilei Wen, Liang Wang, and Jing Liu. Semi-supervised temporal action proposal generation via exploiting 2-d proposal map. *IEEE Transactions on Multimedia*, 24:3624–3635, 2022. ISSN 1941-0077. doi: 10.1109/TMM.2021.3104398. URL `https://ieeexplore.ieee.org/document/9513598/`.

[177] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in neural information processing systems*, 34:18408–18419, 2021.

A
Temporal Models

Within the field of video based temporal segmentation of medical procedures, various temporal models have been explored, which are listed below and will be discussed in the following sections.

- Hidden Markov Models[100;101;122;102]
- Recurrent Neural Networks[103]
- Long Short-Term Memory networks[123;124;125;126;127;128;104;105;129;130;131;132;133;134]
- Three-Dimensional Convolutional Neural Networks[106;107;108]
- Temporal Convolutional Networks[109;110;111;5;135;136;137;138]
- Transformer based[112;139;140;141;85;142]
  - Vision Transformer[114;143;144;145]
  - Video Transformer Network[123;113]

Much of the information presented in this chapter is based on *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* by Aurélien Géron[146].

A.1. Hidden Markov Model

Hidden Markov Models (HMMs) are among the earliest temporal models applied to surgical phase segmentation. In an HMM, the observed features (either handcrafted or learned) are assumed to be generated by an underlying sequence of unobserved states, which in this context are the surgical phases. HMMs are based on the Markov property: the probability of being in a particular state (phase) at time t depends only on the state at time $t-1$, not on any earlier states.

An HMM consists of two key components:

1. Transition probabilities, which describe the likelihood of moving from one hidden state (phase) to another. These ensure the final sequence follows a logical order.
2. Emission probabilities, which describe the likelihood of observing a particular feature vector given the current hidden state.

When training a HMM, it learns both the transition and emission probabilities.

HMMs have a simple and lightweight probabilistic structure, that explicitly incorporates the sequential phase order. However, they struggle to capture long-range temporal patterns due to the Markov assumption, which limits temporal dependencies to only the immediately preceding state (phase).

A.2. Recurrent Neural Network

Recurrent Neural Networks (RNNs) are a special type of neural network developed to handle sequential data. At each time step, an RNN takes an input vector and combines it with a hidden state from the previous step. This allows the network to learn short-term temporal dependencies without significantly increasing model complexity or increasing input size (Figure A.1).

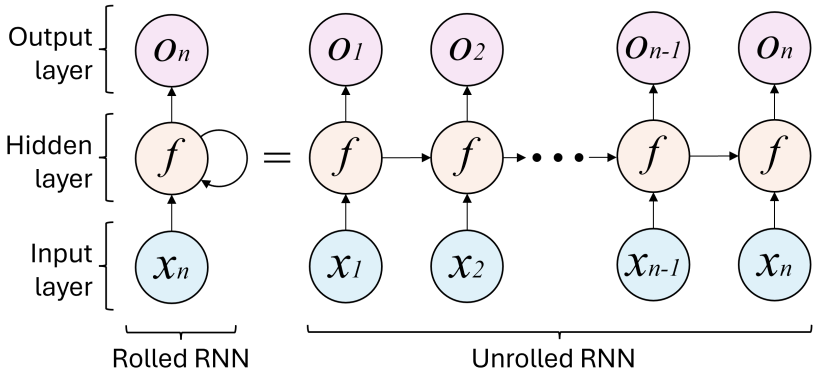


Figure A.1: Graphical representation of a simple recurrent neural network with one hidden layer. x_t represents an input feature at time t, f represents an activation function, O_t represents an output at time t. Due to recursion, the model retains information from previous frames.

Although standard RNNs can handle short-term dependencies while keeping the parameter count manageable, thus reducing the need for excessive training data, they struggle with long-term dependencies due to the vanishing gradient problem. During backpropagation, gradients either shrink too much (vanishing gradients) or explode, making it difficult to learn relationships between frames that are temporally separated far apart.

A.3. Long Short-Term Memory networks

Long Short-Term Memory networks (LSTM) are a type of RNN designed to address the vanishing gradient problem and effectively model long-term dependencies.

Each LSTM unit consists of several key components that regulate information flow over time (see figure A.2). The forget gate (f_t) determines which information is retained, and which information is discarded from the previous memory cell. The input gate (i_t) controls how much new information is stored in the memory cell, allowing the model to update its internal memory based on the incoming data. The cell state (C_t) acts as long-term memory. It maintains long-term dependencies by retaining relevant information across frames. Finally, the output gate (o_t) determines which information from the current memory cell is used for the final hidden state. The final hidden state serves both as the output of the model, and as short-term memory.

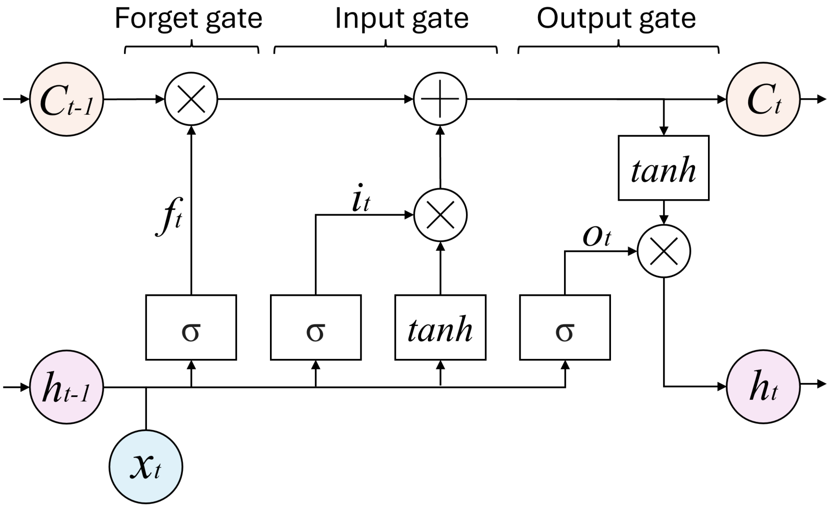


Figure A.2: Graphical representation of a long short-term memory network. C_t represents the cell state (or long term memory) at time t, h_t the output of the model, and the sort term memory at time t, f_t the forget gate, i_t the input gate, o_t the output gate, σ the sigmoid activation function, and $tanh$ the tanh activation function.

At each time step t, the LSTM updates its memory and hidden state by selectively forgetting, updating, and outputting information. This enables LSTMs to retain important historical information while discarding irrelevant past details, learn both short-term and long-term temporal dependencies between surgical phases, and reduce phase flickering by considering both the past and current context. Despite this, LSTMs still struggle with very long-range temporal dependencies.[61]

A.4. Three-Dimensional Convolutional Neural Network

Three-Dimensional Convolutional Neural Networks (3D CNNs) extend traditional 2D CNNs by adding a temporal dimension to the kernel. This enables them to perform spatiotemporal convolutions, allowing the model to learn motion patterns. Although 3D CNNs are able to understand for example hand or tool movements, the kernel expansion to the temporal dimension results in a significant increase in model parameters. Thus both computational cost, and required training data is significantly increased compared to their 2D CNNs. Furthermore, 3D CNNs by themselves can only model short-range temporal dependencies, still requiring models such as LSTMs to model long-range dependencies.

A.5. Temporal Convolutional Network

Temporal Convolutional Networks (TCNs) address the short temporal range limitation of 3D CNNs. They apply 1D convolutions over time to model temporal dependencies, using dilated causal convolutions to capture both short- and long-range relationships. The use of dilation allows the model to incorporate previous information of a much longer time span without requiring a proportional increase in the number of layers (see Figure A.3).

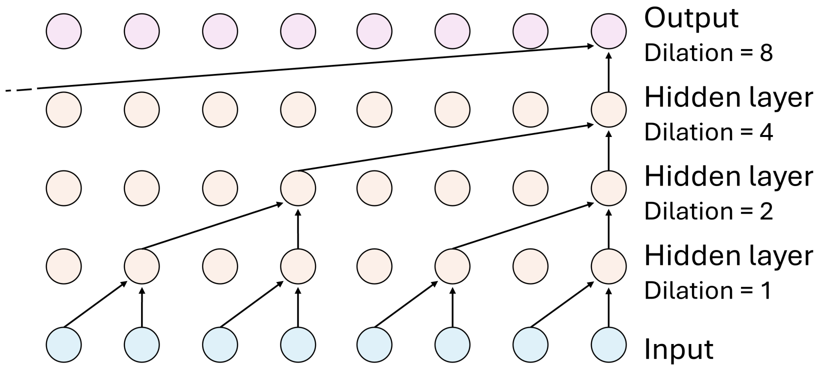


Figure A.3: Graphical representation of dilated causal convolutions. Temporal receptive field increases exponentially with increased network depth.

To prevent vanishing gradients in deep networks, TCNs incorporate residual connections, similar to ResNets. This allows them to be trained even with many layers. However, despite their advantages, TCNs still have limitations. The maximum receptive field is determined by the number of layers and dilation rates. If sequences are extremely long, TCNs may still struggle to capture dependencies between far away time steps.

TCNs can be designed with varying numbers of stages. While Ramesh et al. (2021)[138] found that a multi-stage TCN did not provide a significant improvement over the single-stage variant for both step and phase recognition, the majority of TCNs in the included literature still contain multiple stages[109;147] [135;136;110;111].

A.6. Transformer based

Recent advances in deep learning have introduced transformers, originally developed for natural language processing (the T in GPT stands for Transformer), into the field of TAS. Transformers utilize the self-attention mechanism to process the sequential data and capture long-range dependencies.

The self-attention mechanism determines the relevance of different input tokens when producing an output. Transformers specifically utilize the multi-head attention mechanism, which allows the model to process information through multiple self-attention layers simultaneously. This gives the model the ability to focus on different aspects of the input, capturing a richer representation of it. By dividing the attention process into multiple heads, the mechanism can attend to different parts of the input sequence independently. Multi-head attention improves the expressiveness of attention layers without significantly increasing the number of parameters. It achieves this by running several attention computations in parallel and then merging their results.

In contrast to other methods, transformers are able to model extremely long-range dependencies. The self-attention mechanism computes the relevance of all parts of the input sequence simultaneously. However, given that the self-attention mechanism scales quadratically with sequence length, it is computationally expensive for long videos. Furthermore, transformers require large amounts of annotated training data. For this reason, transformers are often pretrained on large video datasets before being fine-tuned on surgical videos. For this reason, transformers are often pretrained on large video datasets before being fine-tuned on surgical videos.

Two noteworthy transformer architectures used in TAS are Vision Transformers and Video Transformer Networks.

A.6.1. Vision Transformer

The Vision Transformer (ViT) treats each video frame as a sequence of small patches. These patches are projected into feature vectors, and a positional embedding is added to retain spatial information. The resulting embeddings are then used as tokens in the self-attention mechanism, modeling spatial dependencies. In TAS, ViTs often process frames independently and primarily serve as feature extractors. They encode spatial information that can then be passed to temporal models for action segmentation.

A.6.2. Video Transformer Network

Video Transformer Networks (VTNs)[148] extend ViTs by integrating temporal self-attention across frames. Instead of treating each frame individually, VTNs model relationships between frames, capturing both spatial and temporal dependencies.

A typical VTN consists of a feature extractor (often a 2D CNN or ViT) that extracts spatial features from each frame. This is followed by a transformer encoder that applies self-attention across frames, allowing the model to capture long-range dependencies. Finally, the sequence is passed through a classifier which predict the phase of the current frame.

While transformers have proven to be effective in TAS, their high computational cost, and training data requirements remains a challenge.

B

Limited Supervision Learning

Within the broader domain of action segmentation or recognition, several levels of supervision are used depending on annotation availability:[22]

- *Fully supervised*: Each frame of every training video is annotated with an action label that is available for training.
- *Point-level supervised*: For each action instance in a video, a single frame (a ”point”) within its temporal duration is labeled.
- *Weakly supervised*: Only coarse-grained labels are available for training. These can be an ordered list of occurring labels[115;116], or a set of all possible labels without information about order or occurrence[117].
- *Semi-supervised*: The training set is divided into a small set of fully annotated videos and a (typically larger) set of unlabeled or weakly labeled videos.
- *Self-supervised*: Models are pretrained on a pretext task (such as ordering shuffled frames) such that they learn meaningful features from unlabeled data. The learned features are then leveraged for a downstream task, i.e. supervised, semi-supervised, or unsupervised TAS.
- *Unsupervised*: No labels are available for training.

Although the temporal segmentation of medical videos literature uses a large variety of models, most ($\sim$97%) are fully supervised learning methods, training only on fully annotated data. Obtaining large amounts of fully annotated surgical data is time-consuming and requires specific domain expertise. Therefore, limited supervised learning approaches (weakly supervised, semi-supervised, self-supervised and unsupervised learning) provide promising alternatives. By leveraging both (coarse-grained) labeled and unlabeled data, they can achieve on-par performance with supervised alternatives whilst reducing the amount of time-intensive and costly labels used.

In the following sections, various limited supervision methods, and their implementation in TAS are discussed.

B.1. Weakly supervised methods

Weakly supervised techniques aim to reduce the annotation burden by minimizing the reliance on dense frame-level supervision. Among the studied forms of weak supervision in temporal action segmentation, action transcript-based and action set-based methods are popular, each providing different levels of supervision.[23]

B.1.1. Action transcript-based methods

Action transcript based methods rely on a sequential list of actions that occur in a video. They do not require frame-level annotations or temporal boundary information, thereby reducing the annotation burden. Action transcript based methods can broadly be categorized into iterative two-stage and single-stage methods.[149]

Iterative two-stage solutions begin with an initial estimate of frame-wise labels based on the provided transcript label and progressively improve the previous predictions and re-estimate the model parameters iteratively.[150;151] Supporters of single-stage solutions argue, however, that the two-stage solutions are initialization-sensitive and may not always converge.[23]

Early works by Kuehne et al.[152] provide an example of a two-stage solution using the video modality on medical data. They model each action class using a Hidden Markov Model (HMM). The models are first initialized by an initial segmentation, generated by uniformly distributing all actions across the video timeline. The HMM parameters are optimized to maximize the likelihood that the observed video sequence is generated by the HMMs, given the action order from the transcript. Afterward, new action boundaries are inferred based on the updated model, and these boundaries are used to re-estimate the HMM parameters. Each HMM state's observation probabilities are modeled with Gaussian Mixture Models (GMMs). These steps are repeated until model convergence.

B.1.2. Action set-based methods

Action sets are a unique unordered set of the actions that may occur in a video. They are a weaker form of supervision than action transcripts, as they lack the action ordering and frequency (how often an action occurs).[153]

Unlike transcript-based annotation, which still requires the annotator to watch the entire procedure to determine the sequence and catch rare or anomalous actions, action sets can be annotated rapidly. Since action sets do not guarantee that every listed action occurs in the video (i.e., actions can have zero occurrence),[154], and the order of actions does not need to be correct, a single universal set of possible and anomalous actions can be reused across all videos within the same domain, such as surgery. As a result, the annotation cost is extremely low and remains constant regardless of dataset size.

A 2023 study by Ding et al.[23] compared fully supervised, transcript-based, and action set-based methods on the Breakfast dataset. Fully supervised models achieved an average Mean over Frames (MoF), defined as the proportion of correctly classified frames across the video, of approximately 67%, with the top model reaching 77.5%.[155] In contrast, the best iterative two-stage transcript-based methods reached 49.9%[156], single-stage transcript methods reached 50.8%[157], and action set-based methods achieved 42.4%[158]. These results shows that the reduction in annotation burden comes at a large cost to model performance.

B.1.3. Weak supervision in surgical activity recognition

Ramesh et al.[116] present the only study within video based medical phase segmentation research that applies weak supervision. Their approach tries to segment the surgical videos into fine-grained surgical steps using coarse phase annotations as weak supervision. The proposed model, trained on the Bypass40[138] and CATARACTS[159] datasets, utilizes a coarse-to-fine method in which a subset of videos is annotated with detailed surgical steps, while the remainder are annotated only with higher granularity surgical phases (11 phases vs 44 steps for Bypass40[138] and 5 phases vs 19 steps for CATARACTS[159]).

To exploit both annotation types, they use a Single-Stage Temporal Convolutional Network (SS-TCN) with a ResNet-50 backbone. When step-level labels are available, the model is supervised via cross-entropy loss. For phase-only videos, they use a step-phase mapping matrix that maps step predictions to phases and apply a step-phase dependency loss. This allows the network to learn from both coarse and fine annotations without retraining. Their results show a 10–13% increase in accuracy, precision, recall, and F1 score when using both 3 step-labeled videos and 21 phase-labeled videos compared to only 3 step-labeled videos on the Bypass40 dataset. This increased to 13-22% when training on the CATARACTS dataset with 3 step-labeled videos and 22 phase-labeled videos. The study demonstrates that introducing weak supervision through coarse annotations can substantially enhance model performance in endoscopic settings.

B.2. Self-supervised methods

In the context of temporal action segmentation (TAS), self-supervised methods aim to learn temporally and semantically meaningful video representations from unlabeled videos. These learned representations can then be transferred to downstream tasks such as surgical phase or step recognition using limited annotated data. This approach reduces reliance on expensive manual annotations, as the model no longer needs supervision to learn the structure and meaning of video sequences, but only to perform a downstream task.

Unlike semi-supervised learning, which relies on a mix of labeled and unlabeled data during task-specific training, self-supervised learning involves a two-stage process: (1) pretraining on a pretext task defined on unlabeled data, and (2) fine-tuning on the downstream TAS task. These pretext tasks are designed to encourage the model to learn temporal structure and semantic meaning in the videos, without access to ground-truth labels.

The core idea is to use automatically generated pseudo-labels in the pretext task to learn temporal and semantic features without requiring human annotation.[22] Several pretext tasks have been proposed for self-supervised TAS, which include but are not limited to:

- *Clip/frame order prediction*: The model is trained to identify the correct temporal order of shuffled frames or clips. By learning to sort sequences, the model leverages the chronological order of video frames/clips to learn discriminative temporal representations.[160]
- *Video Jigsaw*: Multiple video frames are divided into grids of patches. The model is trained to solve jigsaw puzzles on these patches from multiple frames. This trains the network to correctly identify the position of a patch within a video frame as well as the position of a patch over time.[161]
- *Rotation prediction*: A set of rotations are applied to all videos, and a pretext task is defined as prediction of these rotations.[162]
- *Video speed*: The model is trained to detect is a video is sped up, or playing at normal rate, thus learning a space-time representation.[163]
- *Contrastive learning*: Contrastive methods learn feature embeddings by bringing similar samples closer together and pushing dissimilar samples apart in latent space. In videos, this often means grouping pairs from different augmentations of the same clip or temporally nearby frames, while pushing frames from other videos further apart.[164]
- *Masked Autoencoders (MAE)*: Random portions of the data are masked and the model is trained to reconstruct the missing parts using the visible information. This helps the model to learn the context and structure of the data, including sequence dynamics and time-dependent events.[165;160] IIn the TAS domain, masked modeling has shown advantages over contrastive learning, particularly in efficiency and robustness. Unlike contrastive learning, MAE does not require carefully curated pairs or augmentations, reducing computational cost in many scenarios.[166]
- *Joint-Embedding Predictive Architectures (JEPA)*: JEPA extends on MAE by predicts a latent representation of the missing content instead of reconstructing pixels. This not only promotes learning of context, but also forces learning of high-level abstractions in latent space rather than detailed pixel reconstructions.[167;166]

Once the pretraining phase is complete, the learned representations are fine-tuned using a small set of annotated examples. Depending on the task and available supervision, this fine-tuning may follow fully supervised, weakly supervised, or semi-supervised methods.

A 2022 benchmark study by Thoker et al.[168] evaluated nine video-based self-supervised models on multiple datasets. They observed that while many self-supervised learning models perform well on standard datasets, performance often varies significantly depending on the downstream task and domain. Therefore, careful consideration must be given to method selection.

B.3. Unsupervised methods

Unsupervised learning eliminates the need for any human-annotated labels during training, making it the most annotation-efficient learning method. Instead of relying on external supervision, these methods exploit intrinsic patterns in the data, such as visual similarity, motion dynamics, and temporal regularities, to learn meaningful segmentations.

In the context of Temporal Action Segmentation (TAS), unsupervised approaches generally fall into three categories: (1) two-step iterative methods, which alternate between representation learning and frame-wise clustering, (2) joint methods, which perform representation learning and clustering simultaneously, and (3) boundary detection methods, which identify action transitions within videos.[23]

The first method to perform temporal action segmentation using solely visual inputs was introduced by Sener and Yao[169]. They used a two-stage approach to segment complex activities into sub-activities that alternate between a discriminative appearance model and a generative temporal model. In each iteration, the model learns a visual representation of sub-activities (discriminative model) and their temporal ordering across videos (generative model).

The appearance model maps frame-level features into a low-dimensional embedding space using a linear transformation, optimized such that visual features from the same sub-activity are pushed together, while different sub-activities are pulled apart. This enables the model to discover visual groupings that correspond to sub-activities. To capture the structure and variation in the temporal ordering of sub-activities, they employ a Generalized Mallows Model (GMM), a probabilistic distribution over permutations. This model allows for variability in action sequence orderings across different videos and can handle missing steps.

Although unsupervised learning has a clear annotation cost advantage, it remains largely absent from surgical phase segmentation literature.[170] Several factors may contribute to this. First, unsupervised models may produce clusters that lack clinical meaning. Second, even fully supervised models struggle with the fine-grained nature of surgical activity recognition due to the high visual similarity between steps and the subtlety of transitions. Therefore, it is unlikely that unsupervised methods will produce sufficient results for clinical applications. Finally, weakly, self-, and semi-supervised methods offer a more practical compromise, requiring less annotation than full supervision while still providing clinically relevant clusters. Nevertheless, these factors remain speculative, as only a handful of studies have tested the potential of unsupervised learning in surgical phase segmentation.

B.4. Semi-supervised methods

Semi-supervised methods only use a small number of videos that are fully annotated and many videos that are either unlabeled or include only weak labels. The techniques used in semi-supervised learning on videos can broadly be categorized into four categories discussed below.[171]

B.4.1. Generative methods

Generative methods aim to model the underlying structure of video data by learning to generate new, realistic-looking frames or features. The core idea is that if a model can recreate what videos look like, it must have learned meaningful representations. A common approach involves Generative Adversarial Networks (GANs), where a generator network produces synthetic frames or features, and a discriminator network tries to distinguish them from real ones, encouraging the generator to improve.[172;173]

In the context of TAS, generative methods remain uncommon, due to the difficulty of producing synthetic images that are sufficiently realistic.[173] However, with recent advancements in image and video generation models, generative approaches may warrant renewed exploration in the TAS domain.

B.4.2. Consistency regularization methods

Models are trained to maintain stable predictions under various perturbations, such as spatial or temporal augmentation[174], noise[174;175], time warping[175;176], time masking[175;176], etc. The perturbation method is crucial to the success of consistency regularization. A small perturbation would be insufficient to learn a robust model, while a lager perturbation may destroy the semantic information of original data.[174]

B.4.3. Pseudo-labeling or self-training methods

These methods assign provisional labels (pseudo-labels) for unlabeled videos using a model trained on the labeled subset. These pseudo-labels act as supervision for further training, enabling the model to generalize beyond the labeled subset. This approach assumes that the model can generalize well enough to assign reasonably accurate labels. However, incorrect pseudo-labels can introduce noise, and degrade performance. Therefore, confidence thresholds or ensemble models are often employed to filter out low-quality pseudo-labels and mitigate error propagation.[24;25]

B.4.4. Hybrid methods

Hybrid methods combine elements from multiple semi-supervised strategies, such as integrating consistency regularization with pseudo-labeling[177], to achieve optimal outcomes.

B.4.5. Semi-supervised learning in video-based medical phase segmentation

Shi et al.[132] is the only study within the video based medical phase segmentation domain that uses semi-supervised learning. Their novel SurgSSL framework is a hybrid approach combining what they call a self-supervised Visual and Temporal Dynamic Consistency (VTDC) method with pseudo-labeling.

Given a video clip, VTDC creates sub-clips by downsampling the clip in the time dimension using flexible stride and using conventual visual-level data augmentation such as flipping, rotation and mirroring, on every frame. Given two subsequences from the same video clip, they train the model to give consistent predictions, similar to consistency regularization. The second stage of SurgSSL performs pseudo-labeling on the VTDC-subclips.

SurgSSL[132] is tested on the Cholec80[100] and M2CAI16[100] datasets, and shows 4-8% accuracy increase for various labeled/unlabeled data ratios compared to fully supervised models trained only on the labeled data. The authors also show that using only 50% of labeled videos yields results nearly equivalent to full-supervision on the Cholec80[100] dataset, demonstrating the method's potential for reducing annotation burden in surgical phase segmentation.

B.5. Conclusion

The choice of learning method for surgical phase segmentation represents a fundamental trade-off between annotation cost and model performance. Each level of supervision offers advantages and limitations that influence its applicability (to the clinical domain).

Fully supervised methods offer the highest accuracy, as they learn directly from frame-level annotations. However, this accuracy comes at an very high annotation cost, requiring surgical domain experts to annotate each frame, which is both time consuming and impractical for large datasets.

Point-level supervision offers a slight reduction in annotation burden by requiring only a single labeled frame per action instance, rather than all frames including temporal boundaries. While this reduces labeling time, it still remains relatively annotation-intensive, and requires domain expert.

Weakly supervised methods significantly reduce the annotation effort. Transcript-based approaches rely on an ordered list of actions occurring in the video, which removes the need for frame-level annotations, but still requires a domain expert to watch the entire video. In contrast, action set-based methods offer a more scalable solution by only requiring an unordered set of possible actions. These sets can be reused across videos, drastically reducing annotation burden. However, this simplicity comes at the cost of reduced model performance.

Self-supervised learning eliminates the need for any labels during pretraining by using pretext tasks to learn temporal and semantic video representations. The pretrained models are then fine-tuned with a small amount of labeled. Although computationally intensive, self-supervised learning can drastically reduce annotation requirements, especially when combined with limited supervision in downstream tasks.

Unsupervised learning, while theoretically the most annotation-efficient, remains largely unexplored in surgical phase segmentation. Its limited adoption is likely due to possible challenges in producing clinically meaningful segmentations in the absence of any supervision.

Semi-supervised learning offers a compelling middle ground, combining a small amount of labeled data with a larger pool of unlabeled or weakly labeled videos. Techniques such as pseudo-labeling, consistency regularization, and hybrid training strategies have demonstrated significant performance gains in surgical datasets like Cholec80[100], in some cases approaching the performance of fully supervised models using only half the annotations.

In conclusion, while fully supervised methods remain the gold standard in surgical phase segmentation performance, they are rarely feasible in practice due to the cost of annotation. Among the alternative methods, combining self-supervised representation learning with hybrid semi-supervised learning, holds the most potential for surgical phase segmentation. It effectively reduces annotation burden while maintaining clinical relevance and segmentation quality.

C

Data subset details

The distribution of procedures across datasets, and the phase occurence rates in each dataset are presented in tables C.1 and C.2 respectively.

Table C.1: Distribution of procedures across datasets. ‘ID’ refers to the internally used procedure identifier. ‘T’ indicates the procedure is used for training, ‘V’ for validation, and ‘E’ for testing.

ID	5A	5B	5C	10A	10B	10C	20A	20B	20C	30A	30B	30C	40A	40B	40C	50	65	80	100	146	Test
100										T			T			T	T	T	T	T	
101													T			T	T	T	T	T	
102									T			T			T	T	T	T	T	T	
105									T			T			T			T	T	T	
112																					E
119												V			V				V	V	
121																				V	
122									T			T			T		T	T	T	T	
127																					E
134	T			T			T			T			T			T	T	T	T	T	
140																				T	
142											V			V				V	V	V	
143													T			T	T	T	T	T	
148																				T	
167																					E
168																				V	
182																					E
183												T			T					T	
184																				T	
189																				T	
195																					E
197													T			T	T	T	T	T	
199															T		T	T	T	T	
221														V			V	V	V	V	
228																					E
234																					E
240																					E
243												T			T					T	
246																				T	
252		T			T			T			T			T					T	T	
253																				T	
256			T			T			T			T			T				T	T	
259							T			T			T			T	T	T	T	T	
266																					E
273										T			T			T	T	T	T	T	
282								T			T			T				T	T	T	
290															V					V	
292																					E
302								T			T			T						T	
307																					E
310																					E
312															T					T	
314											T			T					T	T	
316											T			T						T	
320								T			T			T					T	T	
324															T	T	T	T	T	T	
327									V			V			V					V	
334		V			V			V			V			V				V	V	V	

ID	5A	5B	5C	10A	10B	10C	20A	20B	20C	30A	30B	30C	40A	40B	40C	50	65	80	100	146	Test
335																				T	
346																					E
350					T			T			T			T			T	T	T	T	
360																				V	
362											T			T		T	T	T	T	T	
364									T			T			T				T	T	
365										T			T			T	T	T	T	T	
369							T			T			T			T	T	T	T	T	
371								T			T			T					T	T	
378																					E
383							V			V			V			V	V	V	V	V	
388																				T	
395																					E
399									T			T			T			T	T	T	
400															T					T	
404									T			T			T					T	
411				T			T			T			T			T	T	T	T	T	
416			V			V			V			V			V				V	V	
421		T			T			T			T			T					T	T	
422								V			V			V			V	V	V	V	
435	V			V			V			V			V			V	V	V	V	V	
438							T			T			T			T	T	T	T	T	
446																					E
449									T			T			T		T	T	T	T	
453							T			T			T			T	T	T	T	T	
458						T			T			T			T					T	
461																					E
462																					E
467											T			T				T	T	T	
474																					E
476										T			T			T	T	T	T	T	
478																					E
485										V			V			V	V	V	V	V	
488																					E
491										V			V			V	V	V	V	V	
495																					E
501	T			T			T			T			T			T	T	T	T	T	
502	T			T			T			T			T			T	T	T	T	T	
503	T			T			T			T			T			T	T	T	T	T	
508																				T	
509																					E
510														T					T	T	
515																				T	
526			T			T			T			T			T			T	T	T	
537																					E
540																					E
542						T			T			T			T					T	
545													T			T	T	T	T	T	
548												T			T			T	T	T	
550											T			T						T	
555														T		T	T	T	T	T	
569							T			T			T			T	T	T	T	T	
574							T			T			T			T	T	T	T	T	
576																				T	
579				T			T			T			T			T	T	T	T	T	
581						V			V			V			V				V	V	
585																				T	
586					T			T			T			T						T	
591															T					T	
592									V			V			V				V	V	
595																					E
597																					E
600							V			V			V			V	V	V	V	V	
605															T		T	T	T	T	
620																					E
624																				T	
625																					E
626																				T	
627										T			T			T	T	T	T	T	
638														T					T	T	
643																					E
645						T			T			T			T		T	T	T	T	
646								V			V			V		V	V	V	V	V	
648								T			T			T						T	
650																					E
652												T			T			T	T	T	
661																				T	
662					V			V			V			V				V	V	V	
666														T					T	T	
670																					E
678																					E
679								T			T			T					T	T	

ID	5A	5B	5C	10A	10B	10C	20A	20B	20C	30A	30B	30C	40A	40B	40C	50	65	80	100	146	Test
691					T			T			T			T		T	T	T	T	T	
693													T			T	T	T	T	T	
695												T			T		T	T	T	T	
712																				T	
732		T			T			T			T			T						T	
736																					E
738				T			T			T			T			T	T	T	T	T	
742											V			V		V	V	V	V	V	
745											T			T					T	T	
751			T			T			T			T			T		T	T	T	T	
752																				V	
753														T				T	T	T	
756																				T	
757													V			V	V	V	V	V	
760																					E
767														T				T	T	T	
768																					E
771												V			V					V	
772			T			T			T			T			T	T	T	T	T	T	
781																				V	
783													T			T	T	T	T	T	
791																				T	
792						T			T			T			T		T	T	T	T	
805												T			T	T	T	T	T	T	
810															T	T	T	T	T	T	
812		T			T			T			T			T				T	T	T	
814							T			T			T			T	T	T	T	T	
815								T			T			T			T	T	T	T	
816																					E
824													T			T	T	T	T	T	
827																					E
829													V			V	V	V	V	V	
836																				T	
839														T				T	T	T	
845																					E
854															V					V	
856														V			V	V	V	V	
858										T			T			T	T	T	T	T	
869														T						T	
871								T			T			T						T	
879												T			T			T	T	T	
881										T			T			T	T	T	T	T	
889					T			T			T			T					T	T	
892																					E
914									T			T			T				T	T	
917																				T	
928																					E
933																				T	
936							T			T			T			T	T	T	T	T	
942												T			T				T	T	
944				V			V			V			V			V	V	V	V	V	
946													T			T	T	T	T	T	
966											T			T					T	T	
968				T			T			T			T			T	T	T	T	T	
973															T		T	T	T	T	
980																					E
986										T			T			T	T	T	T	T	
993											T			T			T	T	T	T	
999																					E

Table C.2: Class occurence rates (%) and number of segments per train, test, and validation dataset

Training sets & testing set																					
Clinical Phase	**5A**	**5B**	**5C**	**10A**	**10B**	**10C**	**20A**	**20B**	**20C**	**30A**	**30B**	**30C**	**40A**	**40B**	**40C**	**50**	**65**	**80**	**100**	**146**	**Test**
0	9.1	19.0	14.2	6.1	17.6	15.9	6.6	25.3	14.1	9.5	23.9	14.1	14.5	20.8	13.3	13.7	13.6	16.0	16.4	15.9	22.1
1	1.0	1.1	0.7	0.7	1.6	1.0	1.0	1.2	1.1	1.0	1.4	1.2	0.9	1.3	1.1	0.9	0.9	0.9	1.0	1.1	1.4
2	20.6	25.3	21.9	21.4	20.7	24.3	20.2	20.3	25.6	21.0	21.2	24.6	20.0	22.2	25.2	20.5	21.4	21.3	21.6	22.4	21.9
3	16.8	10.2	11.5	15.2	11.0	11.2	15.7	8.7	11.5	14.7	8.7	12.2	14.5	10.8	11.5	14.5	14.0	13.2	13.2	13.2	11.6
4	1.0	0.8	1.8	0.8	0.7	1.3	0.6	0.5	0.9	0.7	0.9	0.9	0.8	0.9	0.8	0.7	0.7	0.8	0.7	0.8	0.6
5	12.0	12.6	9.3	16.2	14.3	10.4	16.2	14.8	9.2	15.1	13.6	9.1	12.9	12.6	10.9	12.0	11.6	10.8	11.2	11.2	11.4
6	1.8	2.2	1.8	2.4	2.0	1.8	2.2	1.6	2.0	2.1	1.8	2.1	2.0	1.9	2.1	2.0	2.0	2.0	2.0	2.0	1.7
7	12.4	11.3	11.2	10.4	10.3	11.3	10.9	8.4	11.8	10.7	8.7	11.3	10.6	9.1	10.7	11.1	11.2	10.9	10.8	10.8	11.7
8	3.1	4.8	4.4	3.4	4.7	4.4	3.3	4.2	4.8	3.7	3.9	4.3	3.6	4.0	3.8	3.8	4.1	4.1	3.9	3.9	2.9
9	7.5	7.9	8.2	7.7	7.0	7.6	9.2	6.6	7.1	8.9	6.9	7.8	8.2	7.1	7.9	8.2	8.1	7.9	7.8	7.7	7.1
10	1.3	1.1	1.3	1.3	1.8	1.3	1.5	1.5	1.9	1.5	1.4	2.2	1.4	1.3	1.9	1.5	1.7	1.6	1.5	1.5	1.6
11	3.3	3.7	5.2	3.1	4.4	4.2	5.5	4.0	4.1	4.9	4.5	5.7	4.9	4.8	5.1	5.2	5.1	4.9	4.9	5.0	3.6
12	1.8	0.0	5.7	1.9	0.8	3.2	1.3	0.6	2.4	1.0	0.6	1.7	1.0	0.6	1.5	1.0	0.9	1.2	1.1	1.0	0.4
13	8.3	0.0	2.8	9.5	2.9	2.0	5.8	2.3	3.4	5.2	2.5	2.7	4.7	2.6	4.3	4.7	4.7	4.3	3.9	3.5	1.9
n-segments	1148	980	1236	2094	2214	2346	4540	5057	4140	7094	7068	6267	9443	9162	8730	11303	14582	17426	22510	33537	12514

Validation sets																				
Clinical Phase	**5A**	**5B**	**5C**	**10A**	**10B**	**10C**	**20A**	**20B**	**20C**	**30A**	**30B**	**30C**	**40A**	**40B**	**40C**	**50**	**65**	**80**	**100**	**146**
0.0	15.2	53.5	9.9	11.7	49.7	14.0	8.5	39.4	16.2	15.9	31.9	21.0	13.3	33.5	19.1	15.7	18.8	23.5	22.8	22.7
1.0	0.8	1.3	1.4	0.8	1.0	1.0	0.7	1.6	1.5	0.9	1.7	1.2	0.9	1.9	1.0	1.0	1.4	1.4	1.3	1.2
2.0	29.6	9.6	28.2	27.9	13.2	28.0	19.6	17.2	29.5	24.5	20.5	26.9	24.8	20.2	23.7	23.9	23.3	22.5	23.5	22.6
3.0	7.8	7.5	12.0	10.5	7.9	16.2	9.4	7.6	12.6	10.0	11.6	11.3	11.5	10.9	9.8	13.0	12.2	11.2	11.1	11.1
4.0	0.8	0.3	0.7	0.6	0.4	0.7	0.9	0.8	0.7	0.7	0.8	0.6	0.8	0.7	0.6	0.8	0.7	0.7	0.7	0.7
5.0	9.7	3.1	14.1	8.4	3.8	6.7	6.8	4.8	7.4	7.2	4.6	7.3	7.9	6.6	9.4	7.3	8.0	7.2	7.3	8.4
6.0	1.6	1.8	3.5	2.5	1.7	2.4	2.1	1.8	2.2	1.9	1.7	2.1	1.8	1.7	1.8	1.7	1.8	1.8	1.8	1.9
7.0	18.7	6.8	12.7	13.8	6.7	7.6	26.0	6.9	9.5	18.1	7.1	8.1	17.2	8.0	14.0	15.1	14.0	12.5	11.9	12.6
8.0	3.5	1.3	7.7	5.2	1.4	4.5	4.8	2.1	4.6	3.9	2.7	3.7	3.6	2.6	3.5	3.6	3.3	3.1	3.2	3.2
9.0	10.9	5.7	9.2	11.7	5.3	7.4	8.8	5.8	8.8	6.1	7.8	7.7	6.5	6.9	7.5	7.6	6.9	6.7	6.9	6.5
10.0	1.2	1.8	0.7	1.0	1.4	1.0	1.2	1.2	1.2	0.9	1.3	1.2	1.1	1.1	1.1	1.1	1.0	1.1	1.1	1.1
11.0	0.4	7.3	0.0	5.7	7.6	1.0	9.1	10.9	0.9	8.5	8.1	1.0	8.0	6.1	1.2	7.0	7.1	7.0	6.0	4.9
12.0	0.0	0.0	0.0	0.0	0.0	1.2	0.5	0.0	0.6	0.3	0.0	1.4	0.5	0.0	1.1	0.4	0.3	0.3	0.4	0.7
13.0	0.0	0.0	0.0	0.0	0.0	8.6	1.7	0.0	4.2	1.0	0.0	6.6	2.1	0.0	6.4	1.7	1.3	1.0	2.0	2.4
n-segments	257	385	142	477	720	421	1070	1174	851	1727	1748	1463	2361	2433	2171	2987	3868	4794	5835	8424

D

Results

A summary of class-wise segmentation accuracies and Levenshtein edit distance scores of all analyzed models can be found in Table D.1 and D.2, respectively. Detailed per-dataset and per-phase results, including visual representations of predictions made on the test set, can be found using the following link:

> https://tud365-my.sharepoint.com/:f:/r/personal/gdebakker_tudelft_nl/Documents/Master%20thesis%20-%20G.%20de%20Bakker?csf=1&web=1&e=eNpcAw

Access can be requested by sending an email to: g.debakker@student.tudelft.nl

Table D.1: Class-wise segmentation accuracy (% ± std) of analysed models.

	Number of procedures									
Phase	**5**	**10**	**20**	**30**	**40**	**50**	**65**	**80**	**100**	**146**
MS-G3D	24.12 ± 7.16	31.16 ± 4.05	38.20 ± 1.39	37.15 ± 5.43	41.11 ± 0.79	N/A	N/A	N/A	N/A	N/A
MS-G3D + LSTM	60.86 ± 0.94	66.44 ± 4.13	69.63 ± 3.46	71.76 ± 3.48	73.66 ± 1.10	N/A	N/A	N/A	N/A	N/A
MS-G3D + TCN	62.89 ± 3.06	66.12 ± 0.55	70.72 ± 0.31	73.13 ± 1.94	73.06 ± 0.77	N/A	N/A	N/A	N/A	N/A
PR-GCN	28.65 ± 4.23	34.12 ± 5.23	36.60 ± 4.67	42.64 ± 2.59	38.77 ± 5.97	41.90	45.08	43.22	43.39	51.32
PR-GCN + LSTM	61.49 ± 3.89	65.49 ± 5.31	69.69 ± 3.80	74.52 ± 5.33	74.35 ± 3.08	77.35	77.87	77.68	77.39	83.95
PR-GCN + TCN	64.21 ± 3.44	72.12 ± 1.53	71.01 ± 2.58	72.40 ± 3.10	74.56 ± 2.48	N/A	N/A	N/A	N/A	N/A
prePR-GCN	28.17 ± 6.27	32.83 ± 4.55	33.96 ± 8.39	40.71 ± 3.32	40.02 ± 1.66	46.50	49.78	40.16	51.12	53.37
prePR-GCN + LSTM	60.05 ± 8.31	67.85 ± 4.57	69.18 ± 7.39	72.47 ± 4.15	74.00 ± 1.96	77.86	79.86	75.95	83.19	83.85
pseudo(PR-GCN + LSTM)	66.14 ± 3.39	71.78 ± 4.53	72.49 ± 4.79	75.89 ± 3.40	78.29 ± 1.36	N/A	N/A	N/A	N/A	N/A

Table D.2: Levenshtein edit distance (% ± std) of analysed models.

	Number of procedures									
Phase	**5**	**10**	**20**	**30**	**40**	**50**	**65**	**80**	**100**	**146**
MS-G3D + LSTM	.6057 ± .0411	.5558 ± .0133	.5476 ± .0462	.5380 ± .0262	.5321 ± .0077	N/A	N/A	N/A	N/A	N/A
MS-G3D + TCN	.6788 ± .0672	.7299 ± .0468	.6952 ± .0252	.6716 ± .0094	.6763 ± .0547	N/A	N/A	N/A	N/A	N/A
PR-GCN + LSTM	.5616 ± .0432	.5333 ± .0239	.4760 ± .0320	56.41 ± .0196	.4716 ± .0379	.3992	.4951	.5201	.4663	.3837
PR-GCN + TCN	.6921 ± .0216	.6959 ± .0337	.6267 ± .0293	.6161 ± .0635	.6192 ± .0091	N/A	N/A	N/A	N/A	N/A
prePR-GCN + LSTM	.5578 ± .0533	.5364 ± .0237	.4911 ± .0374	.5795 ± .0555	.5085 ± .0199	.4462	.4907	.4379	.4085	.4639
pseudo(PR-GCN + LSTM)	.5802 ± .0069	.5364 ± .0237	.4888 ± .0281	.5254 ± .0669	.4823 ± .0328	N/A	N/A	N/A	N/A	N/A

E

Statistical comparison of models using the Friedman test

E.1. Methodology

When comparing multiple machine learning models across different experimental conditions, it is important to determine whether observed performance differences are statistically significant. This study employs the Friedman test, a non-parametric test that is commonly used to detect differences among several related groups.[119]

In the Friedman test, the algorithms are ranked seperately for each data, where the best performing algorithm gets rank 1, the second best rank 2, etc. The average ranks of each algorithm is then computed by $R_j = \frac{1}{N}\sum_i r_i^j$, where r_i^j is the rank of the j-th out of k algoriths on the i-th out of N data sets.

The null hypothesis H_0 is that all models perform equivalently, i.e., they have equal expected ranks R_j. The alternative hypothesis is that at least one model differs significantly from the others. The Friedman statistic is computed as:

$$\mathcal{X}_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{E.1}$$

If the null hypothesis is rejected, post-hoc tests are required to identify which specific model have statistically different performance. In this work, the Nemenyi test is used, which compares all classifiers pairwise. If the difference in average ranks between two models exceeds the critical difference, their performances are considered significantly different.

E.2. Implementation

In this study, each feature extractor temporal model combination was initially trained 15 times, three times per training set size (5, 10, 20, 30, 40). Accuracy-based rankings are presented in table E.1, where columns represent the different models, and rows represent datasets. The average ranks for the models are as follows:

- MS-G3D + LSTM: 2.73
- MS-G3D + TCN: 2.73
- PR-GCN + LSTM: 2.47
- PR-GCN + TCN: 2.07

Using equations E.1 gives a Friedman statistic $\mathcal{X}_F^2$ equal to 2.68, corresponding to a p-value of 0.44. This indicates that the obtained results do not provide sufficient statistical evidence to conclude with

Table E.1: Clip-wise segmentation accuracy-based model rankings per training set.

Dataset	MS-G3D + LSTM	MS-G3D + TCN	PR-GCN + LSTM	PR-GCN + TCN
5A	1	3	4	2
5B	4	3	1	2
5C	3	2	4	1
10A	1	4	2	3
10B	4	3	2	1
10C	4	3	2	1
20A	3	4	1	2
20B	4	1	3	2
20C	1	3	4	2
30A	2	1	3	4
30B	2	3	1	4
30C	4	2	3	1
40A	2	1	4	3
40B	3	4	1	2
40C	3	4	2	1
Average rank	2.73	2.73	2.47	2.07

Table E.2: Levenshtein edit distance score-based model rankings per training set.

Dataset	MS-G3D + LSTM	MS-G3D + TCN	PR-GCN + LSTM	PR-GCN + TCN
5A	2	3	1	4
5B	2	4	1	3
5C	1	4	2	3
10A	2	3	1	4
10B	2	4	1	3
10C	1	4	2	3
20A	2	4	1	3
20B	2	4	1	3
20C	2	4	1	3
30A	1	3	2	4
30B	1	4	2	3
30C	2	4	3	1
40A	2	4	1	3
40B	2	4	1	3
40C	2	4	1	3
Average rank	1.73	3.80	1.40	3.07

confidence that any model consistently outperforms the others in terms of clip-wise segmentation accuracy

Performing a statistical test comparing Levenshtein edit distance scores gives the average ranks listed in table E.2. The correspsonding Friedman statistic $\mathcal{X}_F^2$ is 34.3 corresponding to a p-value «0.001 indicating that there is a model that is statistically different from the others.

To determine which model(s) differ significantly, a Nemenyi post-hoc test was performed using the *scikit_posthocs* python library. The resulting pairwise p-values are shown in Table E.3.

The post-hoc analysis reveals several statistically significant differences between TCN-based and LSTM-based models, but no statistically significant difference between MS-G3D-based and PR-GCN based models. This suggests that LSTM-based models provide more accurate temporal sequences than TCN-based models in the context of skeleton-based surgical phase segmentation.

It is important to note that the Friedman test assumes that each “experimental condition” is independent of the others. However, because of the nested dataset structure (5A $\subset$ 10A $\subset$ 20A, etc.), subsets are not independent. This violates Friedman’s assumption and artificially inflates the sample size, potentially resulting in overly optimistic p-values and an increased risk of false positives. However, given the

Table E.3: Nemenyi post-hoc test results (p-values) for pairwise model comparisons based on edit distance.

	MS-G3D + LSTM	**MS-G3D + TCN**	**PR-GCN + LSTM**	**PR-GCN + TCN**
MS-G3D + LSTM	1.000	«0.001	0.894	0.024
MS-G3D + TCN	«0.001	1.000	«0.001	0.404
PR-GCN + LSTM	0.894	«0.001	1.000	0.002
PR-GCN + TCN	0.024	0.404	0.002	1.000

size of the achieved p-values and observing the achieved results rationally, it is expected that this does not influence the conclusion. Nevertheless, given the magnitude of the observed differences in p-values and the observed ranking patterns, it is unlikely that this changes the overall interpretation of the results.