PATG: Position-aware Temporal Graph Networks for Surgical Phase Recognition on Laparoscopic Videos

Abdolrahim Kadkhodamohammadi[1*], Imanol Luengo[1] and Danail Stoyanov[1,2]

[1]Innovation Department, Medtronic Digital Surgery, 230 City Road, London, EC1V 2QY, UK.
[2]Wellcome/EPSRC Centre for Interventional and Surgical Sciences, University College London, London, UK.

*Corresponding author(s). E-mail(s): rahim.mohammadi@medtronic.com;
Contributing authors: imanol.luengo@medtronic.com; danail.stoyanov@medtronic.com;

Abstract

Purpose: We tackle the problem of online surgical phase recognition in laparoscopic procedures, which is key in developing context-aware supporting systems. We propose a novel approach to take temporal context in surgical videos into account by precise modeling of temporal neighborhoods.
Methods: We propose a two-stage model to perform phase recognition. A CNN model is used as a feature extractor to project RGB frames into a high-dimensional feature space. We introduce a novel paradigm for surgical phase recognition which utilizes graph neural networks to incorporate temporal information. Unlike recurrent neural networks and temporal convolution networks, our graph-based approach offers a more generic and flexible way for modeling temporal relationships. Each frame is a node in the graph and the edges in the graph are used to define temporal connections among the nodes. The flexible configuration of temporal neighborhood comes at the price of losing temporal order. To mitigate this, our approach takes temporal orders into account by encoding frame positions, which is important to reliably predict surgical phases.

Results: Experiments are carried out on the public Cholec80 dataset that contains 80 annotated videos. The experimental results highlight the superior performance of the proposed approach compared to the state-of-the-art models on this dataset. **Conclusion:** A novel approach for formulating video-based surgical phase recognition is presented. The results indicate that temporal information can be incorporated using graph-based models and positional encoding is important to efficiently utilize temporal information. Graph networks open possibilities to use evidence theory for uncertainty analysis in surgical phase recognition.

Keywords: Positional encoder, graph neural networks, surgical phase recognition, workflow analysis, surgical data science, surgical AI

1 Introduction

Laparoscopic procedures have gained popularity by avoiding large open incisions, decreasing blood loss and pain leading to faster patient recovery time [13]. This however results in new challenges for the surgeon, such as indirect vision, decrease in tactile sensations and the use of laparoscopic instruments. Research in computer-assisted intervention (CAI) has thus focused on developing context-aware supporting systems to alleviate these challenges intraoperatively. Identifying the steps of a procedure, commonly referred to as surgical phases, is a key building block for such supporting systems and allows partitioning procedures into sets of well-defined objectives [11, 13]. Intra-operative surgical phase detection assists surgery monitoring, decision support by delivering context-related information [11, 13, 14] during the procedure, and even developing early warning systems [17]. Such systems can further help by facilitating postoperative reviews and providing a tool for analyzing procedures hence paving the way toward identifying best practices.

The availability of intraoperative video data during laparoscopic interventions has spawned many vision-based approaches for surgical phase recognition. Processing high-dimensional video data is still demanding due to computational requirements, hence two-stage methods are used in the literature. A concise representation is generated for each frame in the first stage and then sequences of frame representations are utilized in the second stage to incorporate temporal information. Convolutional neural networks (CNN) have become the method of choice for generating robust frame features through building task-specific models to map RGB images into a robust feature space, for example AlexNet in EndoNet [15], ResNet in TeCNO [3], Opera [4], DeepPhase [19] and open procedures [11], and I3D in SWNet [18]. In this stage, a frame or a window of frames, e.g. I3D, is processed separately without considering the fact that the data are sequential.

In the second stage, to benefit from the temporal nature of the task, models are proposed to rely on sequential information for making robust and smooth

predictions. Early works have taken temporal context into account using graphical models such as hidden Markov models (HMMs) and Hierarchical HMMs, conditional random fields (CRFs), and Bayesian networks [1]. However, over the last decade, deep learning (DL) models have gained significant success in using temporal context and have replaced probabilistic methods. Recurrent neural networks have feedback connections and gated cells to build an internal state (memory) from a time series. Long short-term memory (LSTM), multi-layer LSTM [10], Bidirectional LSTM [11, 14] have been deployed for surgical phase recognition. Another type of network is temporal convolution networks (TCNs) that rely on dilated convolution applied on the temporal axis to process a time series [6]. Unlike LSTMs, TCNs do not build a memory but combine multi-layer and dilated convolutions to expand the receptive field size for a wider temporal context [3]. TCNs thus offer an implicit control over the receptive field of the temporal model. The temporal receptive field needs to be adapted for different domains. TCNs utilized multi-layer networks, where kernels are dilated further at each layer to achieve long-term temporal context. More recently, *OperA*, a transformer-based model [16], was proposed to encode temporal relationships relying on self-attention mechanism for surgical phase recognition [4]. Self-attention permits exploring long-term relationships in contrast to LSTMs that can forget previous information, or TCNs that require stacking layers to achieve high receptive fields.

In this work, we propose an approach based on graph neural networks (GNNs) to incorporate temporal information. We define a temporal graph over a video, where nodes are video frames and edges connect nodes that are temporally adjacent. Message passing is used to incorporate information from neighboring frames and to update the internal state of nodes. We developed a GNN-based temporal decoder for three main reasons. Firstly, the graph structure allows us to precisely define temporal neighborhoods, hence removing the need for using multiple layers and stages unlike TCNs. Secondly, information from all temporally connected frames are accessible during the update process of each node. This is in contrast to LSTMs that build a memory state and update the memory at each time step. Thirdly, as the temporal aggregation function and the node state update function are shared among all nodes, such a model has a much lower number of parameters compared to transformer-based models. This is especially important in the case of surgical phase recognition due to scarcity of data.

Temporal aggregating can happen in any order. It does not have any notion of a frame's temporal position, which is important to disambiguate different phases. We propose to encode positional information as edge attributes, inspired by the work in transformers [4, 16]. We argue that encoding positions are important to effectively build and use temporal context.

We performed a series of experiments on the Cholec80 dataset and compared with state-of-the-art models. Experiments show that our model successfully incorporated temporal context for building robust models and

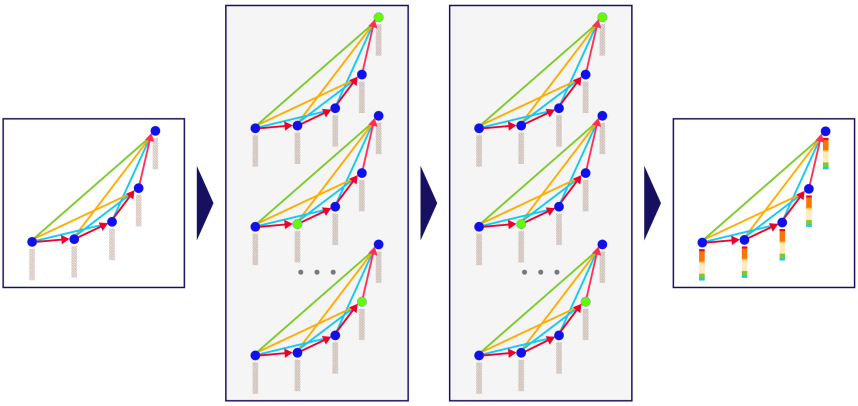

Fig. 1 Position-aware Temporal Graph. Each node in the graph denotes a frame in the video, which is represented by a feature vector (gray boxes). Frame relative position is encoded on the edges, which are denoted by different color here, e.g. blue one time step and green 4 time step. During message passing, each node, highlighted in green, aggregates information from all its neighbors and update its embedding (the gray box). The PATG graph is constructed over a video and passed through several layers, illustrated by gray boxes, to update node embeddings via message passing. The updated embeddings at final layer is used to predict phase labels, illustrated by the color bars.

outperformed state-of-the-art models. To our knowledge, this is the first GNN-based model for surgical phase recognition. This graph representation brings us closer to graphical models and potentially opens possibilities to study uncertainty in phase recognition using evidence theory in addition to Bayesian deep learning.

2 Method

Our proposed approach consists of an encoder and decoder. The encoder is a *SEResNet50* convolutional network to extract high-level concise representations [9]. The encoder is trained for frame-wise phase recognition only. Our model only relies on phase annotations for training, unlike other methods that rely on other sources of information, like instrument in case of [3, 4]. Although Cholec80 videos are annotated with both phase and instrument labels [15], frame-level instrument labels are not always available. Generating such labels is more expensive and time-consuming compared with phase labels. The last fully connected layer of the encoder is used to extract 2048-dimension feature vectors. The frame representations are then fed into the temporal model described next.

2.1 Graph Neural Networks

A graph neural network (GNN) is a class of neural networks designed to represent data in a structured manner using graphs. Graphs can be of arbitrary

topology, and thus, makes this a very flexible representation that can encode a variety of spatiotemporal relationships. GNNs offer to model problems using spectral graph theory and generalize convolutions to non-Euclidean data for different tasks, such as classification and regression [5, 12]. This is achieved by a differentiable implementation of message passing that enables exchanging vector messages between nodes in a graph through a form of belief propagation and utilizing neural networks for updating messages and node embeddings. Formally, a graph $\mathcal{G}$ can be defined as a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges indicating node adjacency. Each node $v_i \in \mathcal{V}$ is represented by a feature vector $x_i \in R^D$, where D is 2048 in our case. The edges define neighborhoods over the graph, which are used to aggregate information. Message passing or neighborhood aggregation is defined as:

$$x_i^{k+1} = f^k(x_i^k, Agg_{j \in N(i)} g^k(x_i^k, x_j^k)) \tag{1}$$

where f and g are differentiable functions, i.e., neural networks, and Agg is a permutation invariant and differentiable function. During each iteration of the message passing, the embedding x_i is updated according to the aggregated information from v_i's neighborhood denoted by $N(i)$. As the iterations progress each node will have access to information from further away nodes. For example, after the second iteration, i.e., second layer, each node contains information from nodes that are reachable by a path of maximum length of 2 in the graph.

2.2 Position-aware Temporal Graph

The message passing provides a powerful way to incorporate information from neighboring nodes. We therefore propose to define a temporal graph over a video and rely on message passing to aggregate and incorporate temporal context. Temporal graph enables us to precisely define the temporal neighborhood by constructing the topology of the selected graph allowing us to encode known procedural information. But as the aggregation function is permutation invariant and also does not preserve frame position in the video, all direct neighbors will equally contribute to the node. This is not a desired behavior for a temporal graph since variation in the temporal distance between frames should affect how a frame contributes to the context of another frame. One solution would be to build the graph in way that only nodes with the same temporal distance are connected. Each frame will only be connected to the next frame in case of online phase recognition. Increasing temporal context would imply adding more layers for more massage passing iterations hence expanding the neighborhood by the number of layers [2]. For example, 60 layers would be required to construct a neighborhood of 60 seconds over a 1 FPS video. This results in adding more parameters and limited ability in defining the graph.

To mitigate these issues, we proposed position-aware temporal graph (PATG). PATG allows encoding frames positions and utilizes them during

message passing to more accurately use neighbors for updating node embeddings. Diagram of a sample PATG is shown in Figure 1. As we are interested in online phase recognition, a directed graph is used to connect past frames to the current frame in PATG. Temporal edges are grouped based on their corresponding path length. We use a positional encoding function to inject frame positions during message passing iterations by:

$$x_i^{k+1} = f^k(x_i^k, Agg_{j \in N(i)} g^k(x_i^k, x_j^k, P_{i,j})) \tag{2}$$

where $P_{i,j}$ is function to encode frame positions. Similarly to Transformer [16], we define the positional encoder as:

$$P_{i,j}^{2l} = sin(\frac{i-j}{10000^{2l/d}}), P_{i,j}^{2l+1} = cos(\frac{i-j}{10000^{2l/d}}) \tag{3}$$

where $l \in [0, d/2]$, d is the positional encoding dimension and i, j denoting the frame index in the video. The message from v_j to v_i is computed based on their embeddings and positions in the video. The function g determines this relationship, which is learned through backpropagation. We can therefore define graphs that can connect frames from different parts of a video and rely on the neural network g to take the temporal context into account and compute the update message for each node.

Our temporal decoder model architecture consists of three blocks: The first block is a convolutional layer followed by a nonlinearity ReLU function for reducing encoder representation to the dimension of the node embeddings $\mathcal{F}$. The second block has n layers of graph convolutions. We use principal neighborhood aggregation (PNA) layers as graph convolution [2]. PNA has demonstrated significant improvement over most of graph convolutions on different benchmark from real-world domains by combining multiple aggregators and degree-based scalars [2]. The last block is classification head that first reduces the node dimension to half, which is followed a ReLU and finally a fully connected layer of the size of the ouput classes.

3 Experimental setup

Dataset. For our experiments, we used the public *Cholec80* dataset [15]. This dataset was generated from 80 laparoscopic videos recorded intra-operatively during laparoscopic cholecystectomy procedures performed by 13 surgeons. The videos were captured at 25 FPS and annotated with surgical phases to provide annotation for seven surgical phases. In average, the phases are from 3 to 10 minutes long. We downsampled the videos to 1 fps. Model performance was measured using accuracy, precision, recall and F1 scores for online recognition of the phases, which means future frames were not used for the prediction of the current frame. We reported the performance of our model on two splits: the original splits suggested by [15], where the first 40 videos for training and the rest for test, and five-fold cross validation similarly to [4].

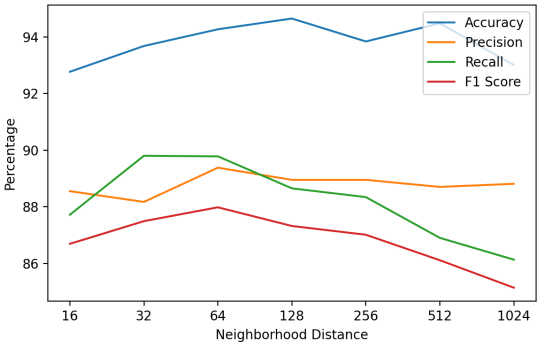

Fig. 2 Neighborhood distance. The effect of neighborhood distance on model performance.

Training and Model Parameters. We set $\mathcal{F}$, the node embedding dimension, to 256. The PNA graph convolution layers uses $min, max, mean$ and STD aggregation functions, and $identity, attenuation$ and $amplification$ as degree-based scalars. As in [2, 8], we use four towers to improve model generalization and computational complexity. This means that the node embedding representations are divided into four, forwarded through the graph convolution layer and then concatenated. We noticed that adding graph convolution layers up to four layers improves performance, but adding more layers did not show any improvement. Hence, the number of graph convolutional layers, n, was set to four. We set the positional encoding dimension, d, to 32. PyTorch Geometric v1.7.2 [7] was used to implement our model. We used the Adam optimizer with learning rate of 0.0001 and weight decay of $1e^{-5}$. The dropout for the graph convolution layers is set to 0.2.

4 Results and Discussion

The SEResNet50 encoder was first trained to classify frames into different phases. The fully connected layer before the classification layer was used to encode RGB frames into compact 2048 feature representations. These representations were fed to temporal models. We first conducted experiments to determine and assess maximum neighborhood distance on a set of 20 randomly selected test videos for the first fold, which are provided later in this section. In Figure 2, the horizontal axis indicates the maximum temporal distance to connect nodes in the graph and the vertical axis indicates performance in four metrics. Our model consists of four PNA layers, which means the effective neighborhood at the last layer is up to four times of the max distance used to generate the temporal neighborhood. Small temporal neighborhood limits the receptive field size hence leads to low performance. The best precision, recall and F1 score were obtained when the temporal graph was defined over a neighborhood of 64 frames, which means each frame can use the temporal context up to four minutes. We however noticed that enlarging the neighborhoods

leads to drop in recall and hence F1 score. The maximum temporal distance of 256 implies that a history of more than half an hour is accessible for each frame. The small size of the training dataset and relatively short duration of the videos made it difficult to leverage such a long-term temporal context. This has resulted in missing short phases that had been reflected by the low recall. For the rest of the experiments, temporal edges were defined between each frame and its past 64 frames.

	Accuracy	**Precision**	**Recall**	**F1 Score**
EndoNet	81.7	73.7	79.6	—
MTRCNet-CL	89.2	86.9	88.0	87.4
SENet50+LSTM	89.36	82.96	83.01	80.74
SENet50+TCN	88.31	83.05	82.07	80.1
SENet50+PATG	91.36	86.88	84	84.19

Table 1 Performance results on Cholec80's original 40/40 split. The performance results of our PATG model and LSTM and TCN models trained on the same encoder are presented and compared with EndoNet and MTRCNet-CL multitask models.

We followed the original split suggested by [15] and used the last 40 videos as a test set and the first 40 videos to train our models. These results are presented in Table 1. To establish strong baselines, TCN- and LSTM-based models are trained on the same encoder feature as our PATG model. In *SENet50+LSTM*, a single-layer LSTM with 512 cells were used and adding more layer does not lead to any improvement. *SENet+TCN* had five stages and ten layers in each stage similar to [3]. Our GNN-based approach achieves better results compared to the two baseline models. Evan though our PATG model outperformed the LSTM-based model, it achieved a lower recall compared to MTRCNet-CL that used a similar convolutional network encoder and LSTM. One should however note that EndoNet and MTRCNet-CL models are not directly comparable to our models as both models require extra instrument labels.

	Accuracy	**Precision**	**Recall**	**F1 Score**
OperA	91.26±0.64	—	—	84.49±0.60
TeCNO	88.56±0.27	81.64±0.41	85.24±1.06	—
PATG	**93.77±0.44**	**89.79±0.79**	**89.11±0.65**	**88.22±0.18**

Table 2 Performance results on Cholec80. The performance results of state-of-the-art models are compared with our proposed model. The results are computed over a five-fold cross validation. Our PATG model is a single-task model while both OperA and TeCNO are multi-task models that require instrument presence labels in addition to phase labels.

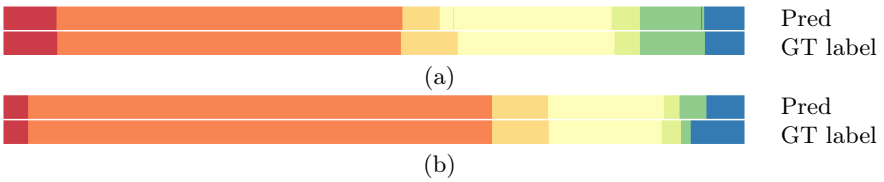

Fig. 3 Qualitative results. The prediction of our PATG model are plotted against ground truth for two test videos. Each pair of bar plot belongs to one video, where our prediction plotted in the first row and the ground truth label in the second row.

Similarly to the experiment setup in [4], we also computed the performance of our model on a five-fold cross validation setup [1], where for each fold 20 videos were randomly selected as the test set and the other 60 videos were used to train the model. The performance results of our approach are presented in Table 2 and compared with state-of-the-art models on the Cholec80 dataset. The average and standard deviation of our model performance are presented in Table 2. The average confusion matrix over all folds is shown in Figure 4. One can notice that confusion occurs frequently between consecutive phases, which are indications of early and late transitions. The Gallbladder Packaging phase is an exception that is more frequently confused with other phases. We believe this is due to the fact the packaged gallbladder will remain in the field of the view, hence confusing the model. As the gallbladder is dissected in this phase, it is more often confused with the Gallbladder Dissection phase.

As stated in Section 2, we only used phase annotations to train our model, while OperA and TeCNO relied on both phase and instrument presence signals to train the models. Both our model and OperA were evaluated on a similar setup. Despite using instrument presence signals in OperA, our model outperformed OperA. This indicates the benefit of modeling temporal context using our graph-based approach over the transformer-based architecture used in OperA for the task of surgical phase recognition. Even though OperA can model long-term temporal context, using such a powerful model would potentially require more data. On the other hand, while our model can take into account a large temporal receptive field, it allows us to precisely define the neighborhood based on the problem and dataset size. The comparison with OperA and the results presented in Figure 2 highlight the superiority of our model in cases with a limited number of samples.

Positional encoding. PATG incorporates frame positions to more effectively exploit temporal context. No frame position leads to significant drop in performance. The performance for neighborhoods smaller than 32 decreased by around 5%. While the performance drop was much higher for larger neighborhoods, for example, for a neighborhood of 512, the F1 score drops by 10%. This indicates the importance of frame position in order to effectively build and

[1]The test video are selected from the original Cholec80 test set. The id of the videos in the random test splits are: 1: [45, 52, 63, 55, 62, 65, 60, 78, 72, 53, 66, 44, 73, 49, 59, 50, 77, 42, 46, 80], 2: [43, 59, 78, 53, 66, 42, 41, 65, 67, 55, 46, 54, 45, 73, 49, 56, 50, 74, 64, 69], 3: [56, 46, 48, 69, 79, 72, 41, 68, 76, 67, 51, 57, 53, 47, 52, 78, 54, 65, 43, 80], 4: [62, 49, 60, 75, 55, 44, 67, 64, 52, 53, 70, 74, 69, 80, 45, 78, 66, 63, 50, 43], 5: [56, 48, 72, 71, 57, 43, 78, 59, 50, 41, 80, 75, 55, 69, 66, 65, 64, 77, 42, 60]

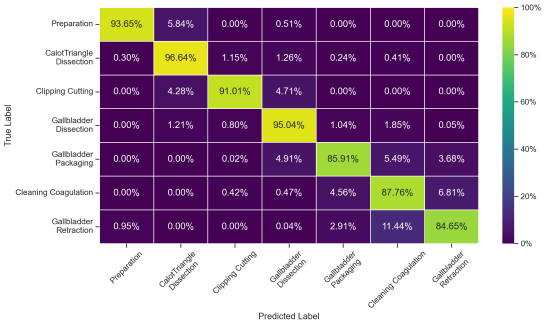

Fig. 4 Average confusion matrix. Average confusion matrix is computed over all folds. The majority of misclassifications are happened between consecutive phases.

utilize long-term temporal context. We also noticed that using absolute frames location in Equation 3 prevents the model from converging. This is potentially due to patient anatomy and surgeon preference that result in variable video length.

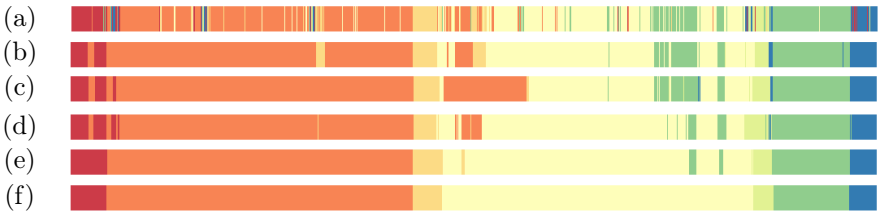

Fig. 5 Phases of Video 72 in Cholec80: (a) Encoder output (b) LSTM-based model (c) TCN-based model (d) Prediction using our PATG-based model trained on 40 videos (e) the same model trained on 60 videos (f) GT label.

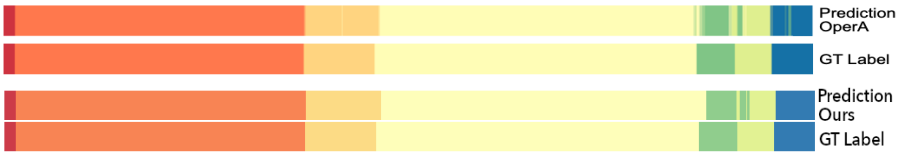

Fig. 6 Phase prediction for video 66. The top two barplots are OperA prediction and the GT label for video 66, courtesy of [4]. The two bottom barplots show our model's prediction for the same video along with GT label.

Qualitative results. Figure 5 shows qualitative results for phase prediction on video 72 in Cholec80. Both LSTM and TCN models (Figure 5 (b) and (c)) utilized the temporal context to reduce alternation between phases. The PATG models smoothed out the predictions and removed the majority of incorrect predictions compared to other models. It is worth mentioning that

training our model on more data resulted in a more robust temporal model, hence better in coping with noisy frame representations.

Figure 3 depicts the performance of our model on two test videos. For each video, two barplots are shown: the model prediction (top) and ground truth label (bottom). The PATG model has successfully used temporal information to correctly recognize phases of frames with noisy representations. The majority of incorrect predictions occurred at phase transitions. In our future work, we will work on phase transition prediction to improve this aspect of the model. Qualitative comparison of our model prediction with OperA on a test video has been shown in Figure 6. OperA is a transformer-based model that can theoretically utilize all the history. In this video, we have noticed that OperA have struggled toward the end of the video. This can potentially be due to either very noisy encoder features or difficulty of using the long-term temporal context. One the other hand, our model, which uses the same SENet50 encoder, has achieved a smoother and more consistent prediction relying on the fixed defined temporal graph.

5 Conclusion

Surgical workflow recognition is a fundamental block in developing context-aware supporting systems to assist clinical teams. In this paper, we present a GNN-based approach for surgical phase recognition on laparoscopic videos. We propose a position-aware temporal graph (PATG) to precisely define a temporal neighborhood and incorporate frame locations. Encoded frame positions are used during the message passing process enabling the use of large temporal neighborhoods. Our model therefore allows us to effectively build and utilize long-term temporal context for robust surgical phase recognition. Our experiment shows that our PATG model has achieved state-of-the-are results on the public Cholec80 dataset. To our knowledge this is the first GNN-based model for surgical phase recognition by constructing temporal graphs over surgical videos. In future work, we would like to use dynamic graphs to explore temporal connections between different frames and evolve the graph accordingly.

Statements and Declarations

Drs. Kadkhodamohammadi and Luengo; and Prof. Stoyanov are employees of Digital Surgery, Medtronic, which is developing products related to the research described in this paper. Prof. Stoyanov is also a co-founder and shareholder in Odin Vision, Ltd.

Conflict of Interests: The authors declare that they have no conflict of interest.

This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Charrière, K., Quellec, G., Lamard, M., Martiano, D., Cazuguel, G., Coatrieux, G. and Cochener, B. [2017], 'Real-time analysis of cataract surgery videos using statistical models', *Multimedia Tools and Applications* **76**(21), 22473–22491.

[2] Corso, G., Cavalleri, L., Beaini, D., Liò, P. and Veličković, P. [2020], 'Principal neighbourhood aggregation for graph nets', *Advances in Neural Information Processing Systems* **33**.

[3] Czempiel, T., Paschali, M., Keicher, M., Simson, W., Feussner, H., Kim, S. T. and Navab, N. [2020], Tecno: Surgical phase recognition with multi-stage temporal convolutional networks, *in* 'Medical Image Computing and Computer Assisted Intervention – MICCAI 2020', Springer International Publishing, pp. 343–352.

[4] Czempiel, T., Paschali, M., Ostler, D., Kim, S. T., Busam, B. and Navab, N. [2021], Opera: Attention-regularized transformers for surgical phase recognition, *in* 'Medical Image Computing and Computer Assisted Intervention – MICCAI 2021', Springer, pp. 604–614.

[5] Defferrard, M., Bresson, X. and Vandergheynst, P. [2016], 'Convolutional neural networks on graphs with fast localized spectral filtering', *Advances in neural information processing systems* **29**, 3844–3852.

[6] Farha, Y. A. and Gall, J. [2019], Ms-tcn: Multi-stage temporal convolutional network for action segmentation, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition', pp. 3575–3584.

[7] Fey, M. and Lenssen, J. E. [2019], 'Fast graph representation learning with pytorch geometric'.
URL: *http://arxiv.org/abs/1903.02428*

[8] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. and Dahl, G. E. [2017], Neural message passing for quantum chemistry, *in* 'International conference on machine learning', PMLR, pp. 1263–1272.

[9] Hu, J., Shen, L. and Sun, G. [2018], Squeeze-and-excitation networks, *in* '2018 IEEE Conference on Computer Vision and Pattern Recognition', pp. 7132–7141.

[10] Jin, Y., Li, H., Dou, Q., Chen, H., Qin, J., Fu, C.-W. and Heng, P.-A. [2020], 'Multi-task recurrent convolutional network with correlation loss for surgical video analysis', *Medical image analysis* **59**, 101572.

[11] Kadkhodamohammadi, A., Uthraraj, N. S., Giataganas, P., Gras, G., Kerr, K., Luengo, I., Oussedik, S. and Stoyanov, D. [2021], 'Towards video-based surgical workflow understanding in open orthopaedic surgery', *Comput. methods Biomech. Biomed. Eng. Imaging Vis.* **9**(3), 286–293.

[12] Kipf, T. N. and Welling, M. [2017], Semi-supervised classification with graph convolutional networks, *in* '5th International Conference on Learning Representations, ICLR 2017'.

[13] Maier-Hein, L., Eisenmann, M., Sarikaya, D., März, K., Collins, T., Malpani, A., Fallert, J., Feussner, H., Giannarou, S., Mascagni, P., Nakawala, H., Park, A., Pugh, C. M., Stoyanov, D., Vedula, S. S., Müller-Stich, B. P., Cleary, K., Fichtinger, G., Forestier, G., Gibaud, B., Grantcharov, T. P., Hashizume, M., Kenngott, H., Kikinis, R., Mündermann, L., Navab, N., Onogur, S., Sznitman, R., Taylor, R. H., Tizabi, M. D., Wagner, M., Hager, G. D., Neumuth, T., Padoy, N., Jannin, P. and Speidel, S. [2020], 'Surgical data science - from concepts to clinical translation', *CoRR* **abs/2011.02284**.

[14] Padoy, N. [2019], 'Machine and deep learning for workflow recognition during surgery', *Minimally Invasive Therapy & Allied Technologies* **28**(2), 82–90.

[15] Twinanda, A. P., Shehata, S., Mutter, D., Marescaux, J., de Mathelin, M. and Padoy, N. [2017], 'EndoNet: A deep architecture for recognition tasks on laparoscopic videos', *IEEE Transactions on Medical Imaging* **36**(1), 86–97.

[16] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. [2017], Attention is all you need, *in* 'Advances in neural information processing systems', pp. 5998–6008.

[17] Vercauteren, T., Unberath, M., Padoy, N. and Navab, N. [2019], 'Cai4cai: the rise of contextual artificial intelligence in computer-assisted interventions', *Proceedings of the IEEE* **108**(1), 198–214.

[18] Zhang, B., Ghanem, A., Simes, A., Choi, H., Yoo, A. and Min, A. [2021], Swnet: Surgical workflow recognition with deep convolutional network, *in* 'Medical Imaging with Deep Learning'.

[19] Zisimopoulos, O., Flouty, E., Luengo, I., Giataganas, P., Nehme, J., Chow, A. and Stoyanov, D. [2018], Deepphase: surgical phase recognition in cataracts videos, *in* 'International Conference on Medical Image Computing and Computer-Assisted Intervention', Springer, pp. 265–272.