Published in final edited form as:
Rep U S. 2022 October ; 2022: 8017–8024. doi:10.1109/IROS47612.2022.9981611.

Recognition and Prediction of Surgical Gestures and Trajectories Using Transformer Models in Robot-Assisted Surgery

Chang Shi[1,*], **Yi Zheng**[1,*], **Ann Majewicz Fey**[1,2]

[1]Walker Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX 78712, USA

[2]Department of Surgery, UT Southwestern Medical Center, Dallas, TX 75390, USA

Abstract

Surgical activity recognition and prediction can help provide important context in many Robot-Assisted Surgery (RAS) applications, for example, surgical progress monitoring and estimation, surgical skill evaluation, and shared control strategies during teleoperation. Transformer models were first developed for Natural Language Processing (NLP) to model word sequences and soon the method gained popularity for general sequence modeling tasks. In this paper, we propose the novel use of a Transformer model for three tasks: gesture recognition, gesture prediction, and trajectory prediction during RAS. We modify the original Transformer architecture to be able to generate the current gesture sequence, future gesture sequence, and future trajectory sequence estimations using only the current kinematic data of the surgical robot end-effectors. We evaluate our proposed models on the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) and use Leave-One-User-Out (LOUO) cross validation to ensure generalizability of our results. Our models achieve up to 89.3% gesture recognition accuracy, 84.6% gesture prediction accuracy (1 second ahead) and 2.71mm trajectory prediction error (1 second ahead). Our models are comparable to and able to outperform state-of-the-art methods while using only the kinematic data channel. This approach can enabling near-real time surgical activity recognition and prediction.

I. INTRODUCTION

Yang et al. defined autonomy levels for medical robots on a scale from 0 to 5, ranging from no autonomy to full automation which requires no human input [1]. The highest level of fully automated Robotic-Assisted Surgery (RAS) is still far from reality due to technical challenges, regulatory, legal, and ethical concerns. A more achievable short-term goal may be to envision a higher level of assistance offered by surgical robots - Robot-Enhanced Surgery (RES) - where collaborative and adaptive robot partners can leverage surgeon strengths and help overcome possible motor limitations [2]. Recent research developments are enabling this new class of assistive surgical robots with contributions to predict surgeon intent [3], perceive surgical states perception [4], measure expertise levels [5], estimate

chang.shi@austin.utexas.edu, yi.zheng@austin.utexas.edu.

[*]The first and second authors contributed equally to this work. This work was supported by NIH #1R01EB030125

procedural progress [6], monitor surgeon’s stress levels [7] and provide novel guidance through audio, augmented reality, and haptic channels [8]–[11].

The first step for these RES applications is often to perceive the current activities of surgical task and to predict the future surgical activities based on current activities. Segmentation and recognition can be used to decompose a surgical tasks, for example, suturing, into a sequence of surgical gestures (e.g., reaching needle, position needle), and perform surgical skill evaluation based on the executed sequence of gestures [12]. Padoy and Hager introduced a collaborative control method which could assist the surgeon by recognizing the completion of manual subtasks and automated the remaining ones on a da Vinci surgical robot [13]. Moreover, the gesture recognition can also be used to trigger appropriate information displays on either surgeon console monitor or trainer monitor.

Another important aspect is anticipating or predicting the operator’s intent and robot movements. The prediction of robotic surgical instrument’s trajectory can potentially contribute to preventing collision between instruments or with obstacles; therefore, enabling a method to prevent these dangerous adverse events during RAS. The prediction of surgeon’s movements or the trajectory of surgeon side manipulators can help generate reference trajectories to develop haptic feedback methods to improve surgical training outcomes.

Surgical activity recognition and prediction are both time-series sequence modeling problems. Recurrent Neural Networks (RNN) have been widely used for time-series modeling problems, for example, Gated Recurrent Units (GRU) [14] and Long-Short-Term-Memory (LSTM) networks [15]. These techniques were initially aimed to solve NLP problems [14], [16], but they can be easily used for solving other real-world problems such as stock price prediction and human activity recognition [17], [18].

Bahdanau et al. first introduced attention in machine translation where the output will focus its attention on a certain part of a sequence [19]. Although the attention mechanism has been widely studied, such attention mechanisms are primarily used in conjunction with a RNN [20]. The Transformer model has gained popularity after being published by Vaswani et al. [21]. Unlike attention-based RNN, the key feature of the Transformer model is its novel attention mechanism which avoids recurrence and only relies on attention to draw global dependencies between model input and output. It replaces the recurrent layers commonly used in encoder-decoder architectures with multi-head attentions. According to Vaswani et al., it is believed that Transformer can be trained significantly faster than RNN-based architectures during translation tasks.

Contributions:

In this paper, we propose the use of Transformer model for surgical activity recognition and prediction. Our method relies only on kinematic data from the surgical robot and has comparable if not better performance than state-of-the-art surgical activity recognition and prediction methods.

II. BACKGROUND

A. Prior Work in Surgical Activity Recognition

Surgical activity recognition from robot kinematic data has been studied over the last decade. With developments in machine learning techniques, especially deep learning, the methods for surgical activity recognition have evolved from Hidden Markov Models (HMMs) [22] and Conditional Random Fields (CRFs) [12] to more complex deep learning models such as LSTM neural networks [6], [23]–[25]. LSTM is an appropriate tool for time-series sequence modeling due to its inherent structure to “memorize” and “forget” certain points within a sequence of data. In addition to robot kinematic data, surgical video which directly embeds surgical activity information, was also introduced to surgical activity recognition based on the development of Convolutional Neural Networks (CNN) and computer visions [26]. Qin et al. recently proposed Fusion-KVE which uses Temporal Convolution Networks (TCN) [27] and LSTM to process multiple data sources, such as kinematic data and video for surgical gesture estimation [4].

B. Prior Work in Surgical Activity Prediction

Time-series prediction is another popular topic in deep learning, especially LSTM. For example, LSTM has gained its popularity in stock price prediction [17], [28], weather forecasting [29], etc. The use of LSTM is relatively limited in the surgical activity prediction literature. Qin et al. introduced daVinciNet which can simultaneously predict the instrument paths and surgical states in robotic-assisted surgery [3]. The daVinciNet method uses kinematics, vision and events data sequences as an input and uses an LSTM encoder-decoder model as well as a dual-stage attention mechanism to extract information from input sequence and therefore, making predictions seconds in advance [20].

C. Transformer Applications

Although the Transformer model was first designed for machine translation problems, it has been studied recently in other time-series modeling problems. Wu et al. employed a Transformer-based approach to forecasting time-series data and used influenza-like illness (ILI) forecasting as a case study. They showed that the results produced by the approach were favorably comparable to the state-of-the-art [30]. Giuliari et al. used Transformer Network and the larger Bidirectional Transformer (BERT) to predict the future trajectories of the individual people in the scene [31].

These studies give us the confidence that the Transformer model, currently the state-of-the-art in NLP tasks, could have promising performance on time-series modeling. Therefore, inspired by the recent development and validation of Transformer model, we move a step forward to using Transformer model in RAS applications, i.e., gesture recognition, gesture and trajectory prediction.

III. DATASET

The JIGSAWS dataset contains three types of surgical tasks (Knot tying, Needle passing and Suturing) completed by eight subjects in a benchtop setting using a da Vinci Surgical System

[32], [33]. For each trial, the kinematic data of the two surgeon-side manipulators (MTMs) and two patient-side manipulators (PSMs), as well as synchronized video data, are saved. For each manipulator, the time-series kinematic data includes the end-effector position (3), rotation matrix (9), linear velocity (3), angular velocity (3) and gripper angle (1), resulting in 19 features in total for each end-effector. In addition, a key distinguishing feature of the JIGSAWS dataset is annotated gestures which are synchronized with the kinematic data. The dataset specified a common vocabulary compromised of 15 gestures (Table I [33]). We also labeled the unannotated data as "$0's$", resulting in 16 classes in gestures. We used 39 suturing trials for model evaluation, and used all 38 kinematic features of two MTMs or PSMs.

IV. METHOD

We formulate the surgical activity recognition and prediction tasks as a supervised machine learning tasks. We take advantage of the Transformer encoder-decoder model - a model widely used in natural language processing, to process historical information all at once, aiming at better satisfying the requirement of real-time Robot-Enhanced Surgery.

Though the three tasks of gesture recognition, gesture prediction and trajectory prediction share a similar Transformer model architecture, there are slight differences in the input and output format according to their different objectives. We define an observation window with size T_{obs} and a prediction window with size T_{pred}. The gesture recognition task would take the input of the current kinematic data (K) within the $t + 1$ to $t + T_{obs}$ window to generate the gesture labels (G) for the same window. The gesture prediction task would take the input of the current kinematic data within the $t + 1$ to $t + T_{obs}$, as well as the current gesture labels ($t + 1$ to $t + T_{obs}$ window) estimated by the first task, to predict the future time-series gestures labels within the $t + T_{obs} + 1$ to $t + T_{obs} + T_{pred}$ window. The trajectory prediction task, we will use the current kinematic data within the $t + 1$ to $t + T_{obs}$ window, together with the current gesture labels within the $t + 1$ to $t + T_{obs}$ window (from task 1) and the future gesture labels within the $t + T_{obs} + 1$ to $t + T_{obs} + T_{pred}$ window (from task 2), to predict the future time-series end-effector trajectory (P) within the time window of $t + T_{obs} + 1$ to $t + T_{obs} + T_{pred}$.

A. Transformer Model

Our proposed Transformer model resembles the original Transformer architecture which consists of an Encoder and a Decoder [21]. We made modifications on the original Transformer architecture based on our needs, for example, removing the embedding layers which was designed for machine translation tasks (Fig 1 and Table II). For each trial, we used a sliding window (Window size: $T_{obs} = 1\ \mathit{second}$, stride: $S = 1\ \mathit{sample}$) to organize the kinematic data into frames.

a) Encoder: The Encoder consists of a positional encoding layer, a stack of N identical encoder layers, and an output layer. The encoder dimension d_{enc} is determined by the number of features of encoder input sequence ($d_{\mathrm{enc}} = 38$ for gesture recognition and prediction, $d_{enc} = 54$ for trajectory prediction). In order to make use of the order of the encoder input sequence, positional encoding is used to encode sequential information in the encoder input sequence by element-wise addition of the encoder input sequence with a positional encoding vector. Then, the resulting sequence is fed into encoder layers. Each encoder layer has two sub-layers: a multi-head self-attention mechanism, and a fully connected feed-forward network. The encoder layer is be repeated for N times. Finally, the data is passed through a fully connected output layer to map the data from encoder dimension d_{enc} to decoder dimension d_{dec}.

b) Decoder: Similarly, the Decoder consists of a positional encoding layer, a stack of N identical decoder layers, and an output layer. The decoder dimension d_{dec} is determined by the number of features of decoder input sequence ($d_{dec} = 16$ for gesture recognition and prediction, $d_{dec} = 22$ for trajectory prediction). The decoder input will be discussed in following sections. After positional encoding, the sequence is fed to decoder layers. The decoder layer has an additional multi-head attention layer between the multi-head self-attention mechanism and the fully connected feed-forward network. The added multi-head attention layer performs the attention over the output of the encoder stack. Finally, the sequence passes a fully connected layer with a dimension of d_{out} to generate the results. We also employed the look-ahead masking on the decoder input sequence to ensure that the prediction of a time-series data point will only depend on previous data points.

B. Training and Testing

Transformer model has three important hyperparameters that can significantly affect the model performance: Number of Encoder/Decoder Layers (N in Figure 1), Number of Encoder Heads (h_{enc}), Number of Decoder Heads (h_{dec}). The hyperparameters were tuned using grid search. We shuffled the data frames of all 39 suturing trials in JIGSAWS and splitted the training/testing set by 70/30 split for grid search purposes.

a) Gesture Recognition: Gesture recognition can be treated as “translating” from current kinematic data to current gestures. During training, the encoder input sequence consisted of all 38 current kinematic features of either MTMs or PSMs ($K_i, i = t + 1, t + 2, \ldots, t + T_{obs}$). The decoder ground-truth (output) sequence consisted of all 16 gesture classes (current gestures) of the input time steps ($G_i,\ i = t + 1, t + 2, \ldots, t + T_{obs}$). Following the teacher forcing procedure, the decoder input sequence was the shifted-right ground-truth sequence ($G_i, i = t, t + 1, \ldots, t + T_{obs} - 1$), as shown in Fig 2a. During testing and inference, shown in Fig 2b, the first instance in decoder input was a randomized vector R. Then, for each inference step, the predicted value $\hat{G}_i$ from decoder output was added to the decoder input sequence recurrently for the next inference step.

b) Gesture Prediction: Similar to gesture recognition, the encoder input sequence consisted of all 38 current kinematic features of either MTMs or PSMs ($K_i, i = \ t + 1, t + 2, \ldots, t + T_{obs}$). The decoder ground-truth (output) sequence consisted of all 16 gesture classes of the future time steps ($G_i, i = t + T_{obs} + 1, t + T_{obs} + 2, \ldots, t + T_{obs} + T_{pred}$). The decoder input sequence consisted of the current gesture classes ($G_i, i = t + 1, t + 2, \ldots, t + T_{obs}$).

During testing and inference, since the inputs of Encoder and Decoder were known: current kinematic data K from $t + 1$ to $t + T_{obs}$ and current gesture data G from $t + 1$ to $t + T_{obs}$, respectively, the inference did not have any recurrence. Both inputs can be fed into the model at the same time. In both gesture recognition and gesture prediction, cumulative categorical cross-entropy loss is used for the discrepancies between the model output and the ground-truth gesture label.

c) Trajectory Prediction: The encoder input sequence consisted of the concatenation of all 38 current kinematic features and the one-hot vector of the 16 gesture class, of PSMs $([\boldsymbol{K}_i, \boldsymbol{G}_i],\ \boldsymbol{i} = \boldsymbol{t} + 1,\ \boldsymbol{t} + 2, \ldots, \boldsymbol{t} + \boldsymbol{T}_{\boldsymbol{obs}})$. The decoder ground-truth (output) sequence consisted of 6 position dimensions $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ of both left and right end-effectors during the future time steps $(\boldsymbol{P}_i, \boldsymbol{i} = \boldsymbol{t} + \boldsymbol{T}_{obs} + 1, \boldsymbol{t} + \boldsymbol{T}_{obs} + 2, \ldots, \boldsymbol{t} + \boldsymbol{T}_{\boldsymbol{obs}} + \boldsymbol{T}_{pred}$. The decoder input sequence consisted of the current position information $(\boldsymbol{P}_i, \boldsymbol{i} = \boldsymbol{t} + 1, \boldsymbol{t} + 2, \ldots, \boldsymbol{t} + \boldsymbol{T}_{\boldsymbol{obs}})$ and the future gesture information $(\boldsymbol{G}_i, \boldsymbol{i} = \boldsymbol{t} + \boldsymbol{T}_{\boldsymbol{obs}} + 1, \boldsymbol{t} + \boldsymbol{T}_{\boldsymbol{obs}} + 2, \ldots, \boldsymbol{t} + \boldsymbol{T}_{\boldsymbol{obs}} + \boldsymbol{T}_{pred})$. We use the cumulative $\boldsymbol{L}_2$ loss between the predicted end-effector trajectory and the ground-truth trajectory, summed over the $\boldsymbol{T}_{pred}$, as the trajectory loss function. To take the advantage of transformer for processing time series all at once, no recurrent inference as gesture recognition is used. Both inputs are fed into the model all at the same time.

Similar to the original Transformer implementation, We used the Adam optimizer [34] with $\beta_1 = 0.9, \beta_2 = 0.98$ and $\epsilon = 10^{-9}$ and varied the learning rate (lr) over training steps [21]. We used $warmup\_steps = 2000$:

$$lr = d_{\text{dec}}^{-0.5} * \left(steps^{-0.5}, steps * warmup\_steps^{-1.5}\right) \tag{1}$$

V. EXPERIMENTAL EVALUATIONS

We evaluated our gesture recognition, gesture prediction and trajectory prediction models on the JIGSAWS dataset (See Table I).

To evaluate gesture recognition and gesture prediction accuracy, for each data frame, we calculated the percentage of accurately recognized or predicted time steps in the frame. Then, the accuracy was averaged across all the frames in testing dataset.

To evaluate the performance of end-effectors trajectory prediction, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were used:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(y^i - \hat{y}^i\right)^2}{N}} \tag{2}$$

$$MAE = \frac{\sum_{i=1}^{N} \left|y^i - \hat{y}^i\right|}{N} \tag{3}$$

We calculated both metrics for each dimension of the Cartesian end-effector path in the endoscopic reference frame $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ and also the end-effector distance $d = \sqrt{x^2 + y^2 + z^2}$ from the origin (camera tip).

We adopted the Leave-One-User-Out cross validation (LOUO) to train and test the generalizability of our model. In LOUO, for each iteration, the data of i^{th} subject was left out as testing set, and the rest of data for training. Then we averaged the evaluation metrics across all the iterations which led to averaging over all subjects as testing set and reported the mean. The $batch\_size = 64$ and the model was trained for $epoch = 15$ during gesture recognition, $epoch = 40$ during gesture prediction, and $epoch = 50$ during trajectory prediction. For the final evaluation of the trajectory prediction performance, the evaluation metrics is calculated only on the time step of T_{pred} without accounting for the previous prediction steps in the entire prediction window.

VI. RESULTS AND DISCUSSIONS

In order to compare with previous studies in the literature, during gesture recognition we kept the data as its original frequency 30Hz. During gesture and trajectory prediction, we downsampled the data to 10Hz. For each task, the model was trained and evaluated independently.

A. Gesture Recognition

We kept the JIGSAWS data as its original frequency 30Hz. We did not run hyperparameter tuning for gesture recognition as it would taken significant computational effort with the data of 30Hz. Instead, we decided to keep the model for gesture recognition in its simplest form: $N = 1$, $h_{enc} = 1$, and $h_{dec} = 1$. The encoder input sequence was 38 dimensional kinematic features with a length of current observation $T_{obs} = 1s(30\ samples)$. The decoder output was a 16-class gestures with a length of current observation $T_{obs} = 1s(30\ samples)$ of the corresponding encoder input sequence.

We used the data of MTMs and PSMs individually to test the model performance. After LOUO cross validation, the reported accuracy was promising and outperformed the state-of-the-art algorithms (89.3% with MTMs kinematic data as encoder input; 89.2% with PSMs kinematic data as encoder input, in Table III). An example of gesture recognition using the kinematic data of PSMs of a random trial as testing dataset is illustrated in Fig 3.

Fusion-KVE is a method which incorporates kinematic data, video and events data. However, our proposed Transformer model only uses kinematic data and outperforms Fusion-KVE in accuracy. Using less types of data could potentially shorten the computational time for gesture recognition and therefore, enables a near-real-time recognition manner.

B. Gesture Prediction

In order to compare our proposed model with the state-of-the-art, we also downsampled JIGSAWS data to 10Hz. The downsampled data could shorten the computational time during training, therefore, we applied hyperparameter tuning using grid search to optimize the prediction performance. After grid search, the resulting hyperparameters were: $N = 4$, $h_{enc} = 1$ and $h_{dec} = 4$.

During training, the encoder input sequence was 38 dimensional kinematic features of current observation $(K_i,\ \ i = t - T_{obs} + 1, t - T_{obs} + 2, \ldots, t, T_{obs} = 1s)$. The decoder output sequence was predictions of 16-class future gestures $(G_i, i = t + 1, t + 2, \ldots, t + T_{pred})$. It is worth noting that the decoder input sequence is the current gesture sequence $(G_i, i = t + 1, t + 2, \ldots, t + T_{obs})$.

Although our proposed gesture recognition model could output a good estimation of the current gesture sequence, and it's more reasonable to use the current gesture estimation to mimic the real-world application, we decided to train and evaluate the gesture prediction model on "real" current gestures, assuming a perfect current gesture estimation during gesture recognition. It allowed us to independently evaluate the performance of gesture prediction model.

We summarized the gesture prediction model performance in Table IV. Both observation T_{obs} and prediction T_{pred} time lengths were 1 second. Although the reported accuracy (84.6% with MTMs kinematic data as encoder input; 84.0% with PSMs kinematic data as encoder input) did not significantly outperform the state of the art, we still believe our proposed gesture prediction model is promising since less data source (only kinematic data) was used in our study. One example of gesture prediction using the kinematic data of MTMs is shown in Fig 4.

C. Trajectory Prediction

For the trajectory prediction task, we downsampled JIGSAWS data to 10Hz. We used the hyperparameters: $N = 1,\ \ h_{\text{enc}} = 6$ and $h_{dec} = 11$. The Encoder took both the kinematic features and the 16-class gesture class as input. The Decoder took the current x, y, z positions of two endeffectors and the future gestures as input. Similar to Gesture Prediction, during training and testing, we used the ground truth future gestures in decoder input, assuming a perfect gesture prediction, to evaluate the model independently.

Table V summarizes the Transformer performance on the JIGSAWS suturing dataset with the prediction time-step of 1 second $(T_{pred} = 10)$. Using only the PSM data, the Transformer has better performance than daVinciNet on the right arm trajectory prediction, while slightly worse performance on the left arm. Although results are mixed, the Transformer still only uses the kinematics data to obtain as competitive results as using those using both complex video and kinematic data. This would help largely reduce the computation complexity in the applications with a guarantee of accurate and real-time motion and gesture monitoring.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we used the Transformer model, a novel deep learning model initially designed for NLP tasks, to recognize and predict the surgical activities. We modified the Transformer model architecture from the original paper according to the need of our tasks: gesture recognition, and gesture and trajectory prediction during RAS. In gesture recognition, the model took current kinematic data as its input sequence and estimated the corresponding surgical gestures (accuracy: 89.3% using MTMs and 89.2% using PSMs); In gesture prediction, the model took current kinematic data and current surgical gestures as its input sequences and predicted the future (1 second) surgical gestures (accuracy: 84.6% using MTMs and 84.0% using PSMs); In trajectory prediction, we jointly utilize the current kinematic data as well as the future gestures to predict the future (1 second) end-effector trajectory, and reached distance error as low as 2.71 mm.

Considering that our models are purely based on the kinematic data of the end-effectors (MTMs and PSMs) of the daVinci Surgical System without the aid of visual features, the results are very much competitive. Although some studies have shown that combining kinematic data and video could improve the recognition and prediction performance, our work shows the potential to achieve similar performance with only kinematic data, which is preferred when running surgical activity recognition and prediction in a real-time manner [12], [35], since vision data processing is inherently time consuming.

Though the proposed models have outperformed the state-of-the-art methods from literature, there are still some limitations remain unsolved. Future work would include jointly evaluating the performance of gesture and trajectory prediction models. Our gesture prediction model took the current gesture sequence as its decoder input. And our trajectory prediction model took the future gesture sequence as part of its decoder input. However, in our current implementation, we used the ground truth values of the current gestures in gesture prediction and the ground truth values of the future gestures in trajectory prediction, in order to train and evaluate the models independently. To test the robustness and the feasibility of near-real-time manner of the models in real-world gesture and trajectory prediction tasks, our next step would be to use the estimated values of gesture recognition as gesture prediction decoder input, and then, use the estimated values of gesture prediction as trajectory prediction decoder input. We also plan to do more ablation study on measuring the respective difficulty of trajectory prediction of the 16 gesture classes. This would help develop a global sense of when the RAS system should offer more motion guidance and deviation warning.

We believe our proposed methods can contribute to implementation of robot enhanced surgical applications, therefore, augment the role of robots in assisting surgeons through modern control strategies.

References

[1]. Yang G-Z, Cambias J, Cleary K, Daimler E, Drake J, Dupont PE, Hata N, Kazanzides P, Martel S, Patel RV, et al., “Medical robotics—regulatory, ethical, and legal considerations for increasing levels of autonomy,” p. eaam8638, 2017.

[2]. Battaglia E, Boehm J, Zheng Y, Jamieson AR, Gahan J, and Fey AM, “Rethinking autonomous surgery: focusing on enhancement over autonomy,” European urology focus, vol. 7, no. 4, pp. 696–705, 2021. [PubMed: 34246619]

[3]. Qin Y, Feyzabadi S, Allan M, Burdick JW, and Azizian M, “DaVinciNet: Joint prediction of motion and surgical state in robot-assisted surgery,” IEEE International Conference on Intelligent Robots and Systems, pp. 2921–2928, oct 2020.

[4]. Qin Y, Pedram SA, Feyzabadi S, Allan M, McLeod AJ, Burdick JW, and Azizian M, "Temporal segmentation of surgical subtasks through deep learning with multiple data sources," in 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 371–377.

[5]. Wang Z and Majewicz Fey A, "Deep learning with convolutional neural network for objective skill evaluation in robot-assisted surgery," International journal of computer assisted radiology and surgery, vol. 13, no. 12, pp. 1959–1970, 2018. [PubMed: 30255463]

[6]. Van Amsterdam B, Clarkson MJ, and Stoyanov D, "Multi-Task Recurrent Neural Network for Surgical Gesture Recognition and Progress Prediction," Proceedings - IEEE International Conference on Robotics and Automation, pp. 1380–1386, may 2020.

[7]. Zheng Y, Leonard G, Zeh H, and Fey AM, "Frame-wise detection of surgeon stress levels during laparoscopic training using kinematic data," International Journal of Computer Assisted Radiology and Surgery, pp. 1–10, 2022. [PubMed: 35922706]

[8]. Kitagawa M, Dokko D, Okamura AM, and Yuh DD, "Effect of sensory substitution on suture-manipulation forces for robotic surgical systems," The Journal of thoracic and cardiovascular surgery, vol. 129, no. 1, pp. 151–158, 2005. [PubMed: 15632837]

[9]. Culjat M, King C-H, Franco M, Bisley J, Grundfest W, and Dutson E, "Pneumatic balloon actuators for tactile feedback in robotic surgery," Industrial Robot: An International Journal, 2008.

[10]. Ershad M, Rege R, and Fey AM, "Adaptive surgical robotic training using real-time stylistic behavior feedback through haptic cues," IEEE Transactions on Medical Robotics and Bionics, vol. 3, no. 4, pp. 959–969, 2021.

[11]. Zheng Y, Ershad M, and Majewicz-Fey A, "Toward correcting anxious movements using haptic cues on the da vinci surgical robot," in submission to International Conference on Biomedical Robotics and Biomechatronics (BioRob). IEEE RAS/EMBS, 2022.

[12]. Tao L, Zappella L, Hager GD, and Vidal R, "Surgical gesture segmentation and recognition," in International conference on medical image computing and computer-assisted intervention. Springer, 2013, pp. 339–346.

[13]. Padoy N and Hager GD, "Human-machine collaborative surgery using learned models," in 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011, pp. 5285–5292.

[14]. Cho K, Van Merriënboer B, Bahdanau D, and Bengio Y, "On the properties of neural machine translation: Encoder-decoder approaches," arXiv preprint arXiv:1409.1259, 2014.

[15]. Hochreiter S and Schmidhuber J, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997. [PubMed: 9377276]

[16]. Sundermeyer M, Schlüter R, and Ney H, "Lstm neural networks for language modeling," in Thirteenth annual conference of the international speech communication association, 2012.

[17]. Selvin S, Vinayakumar R, Gopalakrishnan E, Menon VK, and Soman K, "Stock price prediction using lstm, rnn and cnn-sliding window model," in 2017 international conference on advances in computing, communications and informatics (icacci). IEEE, 2017, pp. 1643–1647.

[18]. Singh D, Merdivan E, Psychoula I, Kropf J, Hanke S, Geist M, and Holzinger A, "Human activity recognition using recurrent neural networks," in International cross-domain conference for machine learning and knowledge extraction. Springer, 2017, pp. 267–274.

[19]. Bahdanau D, Cho KH, and Bengio Y, "Neural machine translation by jointly learning to align and translate," in 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. International Conference on Learning Representations, ICLR, sep 2015.

[20]. Qin Y, Song D, Chen H, Cheng W, Jiang G, and Cottrell G, "A dual-stage attention-based recurrent neural network for time series prediction," arXiv preprint arXiv:1704.02971, 2017

[21]. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, and Polosukhin I, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[22]. Tao L, Elhamifar E, Khudanpur S, Hager GD, and Vidal R, "Sparse hidden markov models for surgical gesture classification and skill evaluation," in International conference on information processing in computer-assisted interventions. Springer, 2012, pp. 167–177.

[23]. DiPietro R, Lea C, Malpani A, Ahmidi N, Vedula SS, Lee GI, Lee MR, and Hager GD, "Recognizing surgical activities with recurrent neural networks," in International conference on medical image computing and computer-assisted intervention. Springer, 2016, pp. 551–558.

[24]. Yasar MS and Alemzadeh H, "Real-Time Context-Aware Detection of Unsafe Events in Robot-Assisted Surgery," Proceedings - 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2020, pp. 385–397, jun 2020.

[25]. DiPietro R, Ahmidi N, Malpani A, Waldram M, Lee GI, Lee MR, Vedula SS, and Hager GD, "Segmenting and classifying activities in robot-assisted surgery with recurrent neural networks," International journal of computer assisted radiology and surgery, vol. 14, no. 11, pp. 2005–2020, 2019. [PubMed: 31037493]

[26]. Lea C, Reiter A, Vidal R, and Hager GD, "Segmental spatiotemporal cnns for fine-grained action segmentation," in European Conference on Computer Vision. Springer, 2016, pp. 36–52.

[27]. Lea C, Vidal R, Reiter A, and Hager GD, "Temporal convolutional networks: A unified approach to action segmentation," in European Conference on Computer Vision. Springer, 2016, pp. 47–54.

[28]. Mehtab S, Sen J, and Dutta A, "Stock price prediction using machine learning and lstm-based deep learning models," in Symposium on Machine Learning and Metaheuristics Algorithms, and Applications. Springer, 2020, pp. 88–106.

[29]. Salman AG, Heryadi Y, Abdurahman E, and Suparta W, "Single layer & multi-layer long short-term memory (lstm) model with intermediate variables for weather forecasting," Procedia Computer Science, vol. 135, pp. 89–98, 2018.

[30]. Wu N, Green B, Ben X, and O'Banion S, "Deep transformer models for time series forecasting: The influenza prevalence case," arXiv preprint arXiv:2001.08317, 2020.

[31]. Giuliari F, Hasan I, Cristani M, and Galasso F, "Transformer networks for trajectory forecasting," in 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021, pp. 10335–10342.

[32]. Ahmidi N, Tao L, Sefati S, Gao Y, Lea C, Haro BB, Zappella L, Khudanpur S, Vidal R, and Hager GD, "A Dataset and Benchmarks for Segmentation and Recognition of Gestures in Robotic Surgery," IEEE Transactions on Biomedical Engineering, vol. 64, no. 9, pp 2025–2041, sep 2017. [PubMed: 28060703]

[33]. Gao Y, Vedula SS, Reiley CE, Ahmidi N, Varadarajan B, Lin HC, Tao L, Zappella L, Béjar B, Yuh DD, et al. , "Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling," in MICCAI workshop: M2cai, vol. 3, 2014, p. 3.

[34]. Kingma DP and Ba J, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[35]. Zappella L, Béjar B, Hager G, and Vidal R, "Surgical gesture classification from video and kinematic data," Medical Image Analysis, vol. 17, no. 7, pp. 732–745, oct 2013 [PubMed: 23706754]

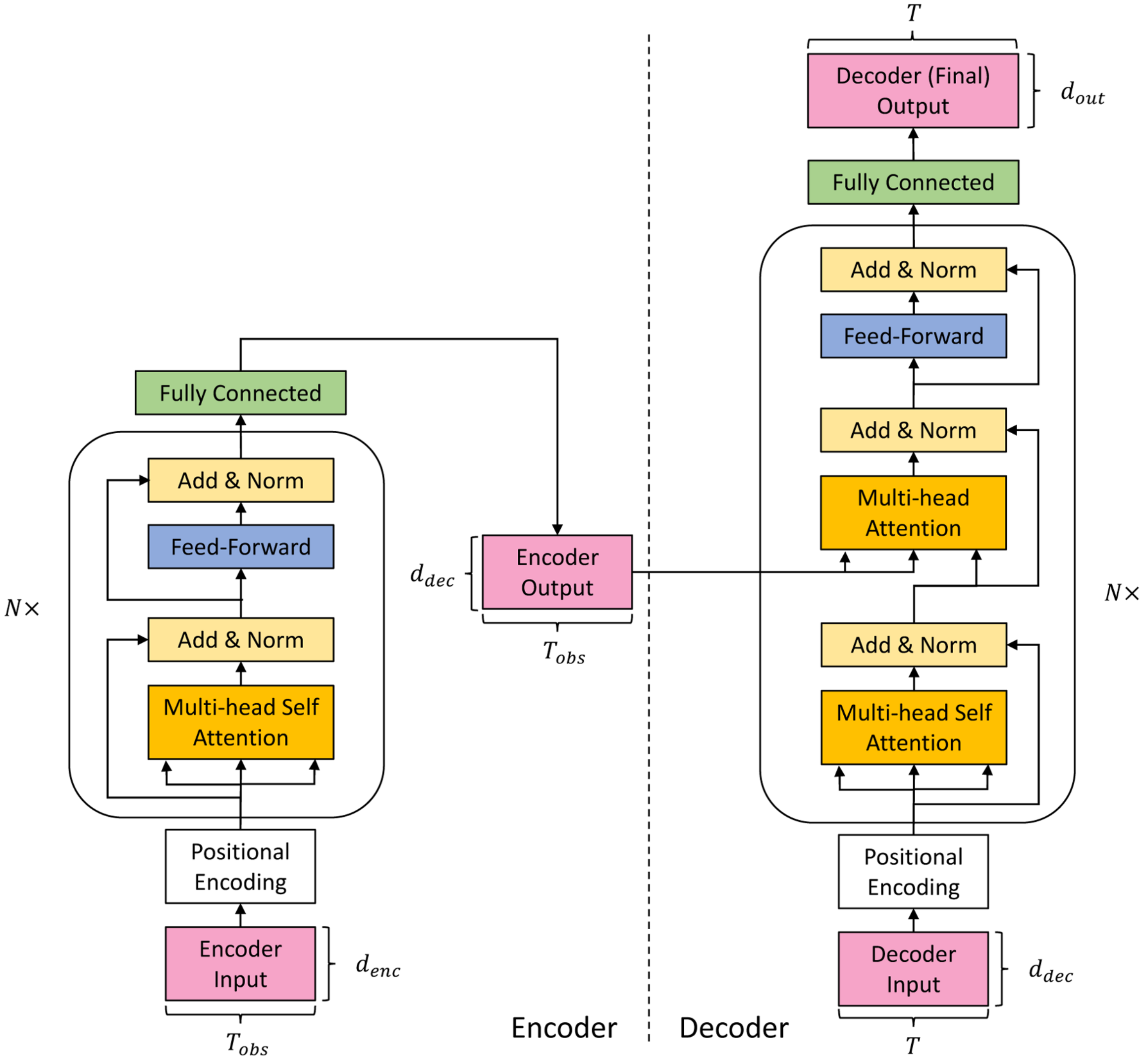
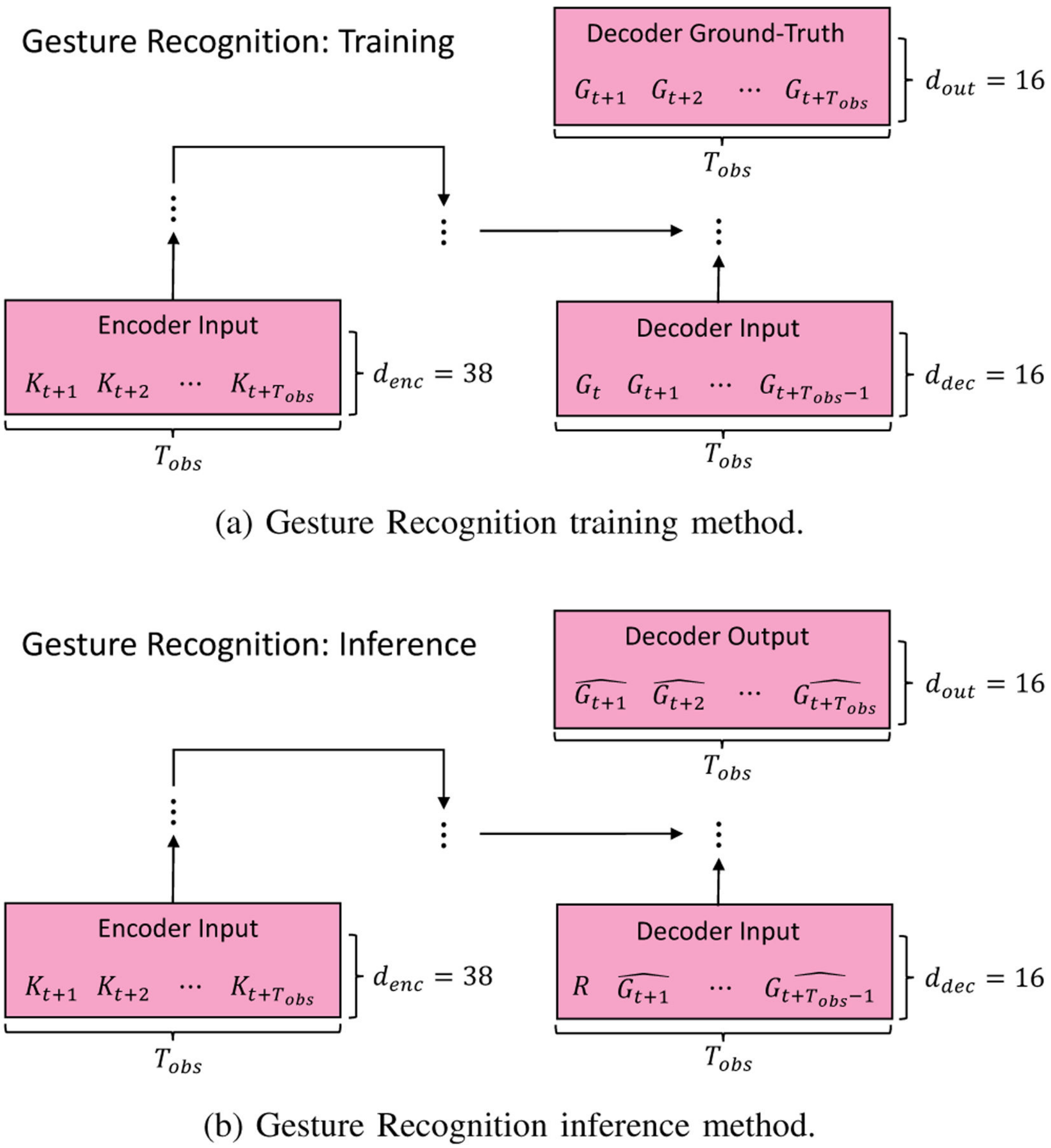


Fig. 1:

Architecture of the proposed Transformer model. In Encoder, $d_{enc} = 38$ during gesture recognition and prediction, $d_{enc} = 54$ during trajectory prediction. In Decoder, $T = T_{obs}$ during gesture recognition and $T = T_{pred}$ during gesture and trajectory prediction; $d_{dec} = 16$ during gesture recognition and prediction, $d_{dec} = 22$ during trajectory prediction; $d_{out} = 16$ during gesture recognition and prediction, $d_{out} = 6$ during trajectory prediction.


Gesture Recognition: Training
Decoder Ground-Truth
G_{t+1} G_{t+2} $\cdots$ $G_{t+T_{obs}}$
$d_{out} = 16$
T_{obs}
Encoder Input
K_{t+1} K_{t+2} $\cdots$ $K_{t+T_{obs}}$
$d_{enc} = 38$
T_{obs}
Decoder Input
G_t G_{t+1} $\cdots$ $G_{t+T_{obs}-1}$
$d_{dec} = 16$
T_{obs}
Gesture Recognition: Inference
Decoder Output
$\widehat{G_{t+1}}$ $\widehat{G_{t+2}}$ $\cdots$ $\widehat{G_{t+T_{obs}}}$
$d_{out} = 16$
T_{obs}
Encoder Input
K_{t+1} K_{t+2} $\cdots$ $K_{t+T_{obs}}$
$d_{enc} = 38$
T_{obs}
Decoder Input
R $\widehat{G_{t+1}}$ $\cdots$ $\widehat{G_{t+T_{obs}-1}}$
$d_{dec} = 16$
T_{obs}

(a) Gesture Recognition training method.

(b) Gesture Recognition inference method.

Fig. 2:

Training and inference (testing) methods during Gesture Recognition. R in (b) is a random vector for initializing inference. $\hat{G}_i$ in (b) is the estimated gesture value by the model.

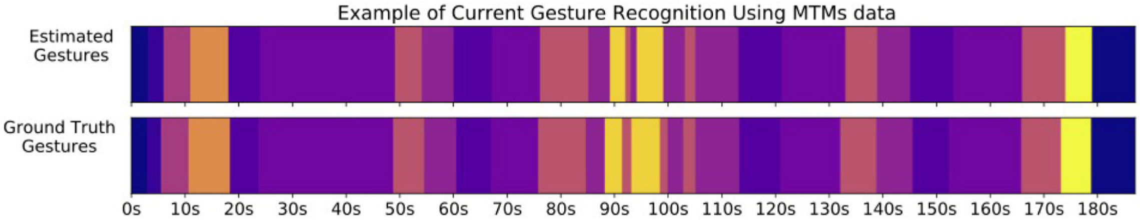


Fig. 3:

An example of Gesture Recognition using a random suturing trial. The top row is the estimated gestures by our proposed Gesture recognition model. The bottom row is the ground-truth labels.

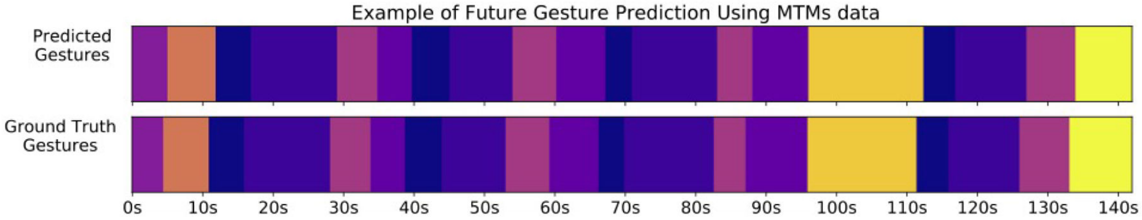


Fig. 4:

An example of Gesture Prediction using a random suturing trial. The top row is the predicted gestures by our proposed gesture prediction model. The bottom row is the ground-truth labels. The Decoder uses real current gestures as input.

Table I:

Gesture Descriptions in JIGSAWS

Gesture ID	Description
G1	Reaching for needle with right hand
G2	Positioning needle
G3	Pushing needle through tissue
G4	Transferring needle from left to right
G5	Moving to center with needle in grip
G6	Pulling suture with left hand
G7	Pulling suture with right hand
G8	Orienting needle
G9	Using right hand to help tighten suture
G10	Loosening more suture
Gil	Dropping suture at end and moving to end points
G12	Reaching for needle with left hand
G13	Making C loop around right hand
G14	Reaching for suture with right hand
G15	Pulling suture with both hands

Table II:

List of modifications based on original Transformer model [21].

Modification	Description
Removing the Embedding layer	It is designed for NLP tasks in which words are embedded to vectors.
Removing the Padding mask	It is designed for NLP tasks in which sentence lengths are different.
	Padding mask ensures that the loss can be calculated efficiently.
Adding Encoder output layer	It is a fully connected layer to map the encoder output’s dimension to decoder dimension, in order to calculate the attention between encoder sequence and decoder sequence.

Table III:

Comparison to prior works of Gesture Recognition under LOUO cross validation. The listed models used the data of 30Hz.

	Data Sources	Accuracy
Fusion-KVE [4]	PSMs+Video	86.3%
Forward LSTM [23]	PSMs	80.5%
Bidir. LSTM [23]	PSMs	83.3%
Transformer	**MTMs**	**89.3%**
Transformer	**PSMs**	**89.2%**

Table IV:

Comparison to prior works of Gesture Prediction under LOUO cross validation. The listed models used the data of 10Hz in JIGSAWS and a prediction length of 1 second.

	Data Sources	**Accuracy**
daVinciNet [3]	PSMs+Video	84.3%
Transformer	**MTMs**	**84.6%**
Transformer	PSMs	84.0%

Table V:

End-effector trajectory prediction performance measures with prediction window of one second ahead(($T_{pred} = 10$)). The prediction performances are reported for the Cartesian end-effector path in the endoscopic reference frame (x, y, z) and $d = \sqrt{x^2 + y^2 + z^2}$.

	Data Sources	**Metric**	x_1	y_1	z_1	d_1	x_2	y_2	z_2	d_2
daVinciNet	PSMs	RMSE	**2.81**	**2.42**	**3.28**	4.16	**3.8**	4.26	4.75	5.92
Transformer	PSMs		3.15	3.03	3.30	**2.93**	3.93	**4.21**	**4.22**	**4.92**
daVinciNet	PSMs	MAE	**2.19**	**1.95**	**2.86**	3.7	**3.42**	3.91	4.31	5.34
Transformer	PSMs		2.86	2.85	3.00	**2.71**	3.60	**3.88**	**3.84**	**4.44**