Final Report

Improving Technical Proficiency in Robot-mediated Surgery Through Counterfactual Inquiry

EN 601.646 Computer Integrated Surgery II

Hao Ding
Ph.D. student of Computer Science department
hding15@jh.edu

Table of Contents

1 Introduction 1

1.1 Motivation 1

1.2 Goals 2

1.2.1 Data 2

1.2.2 Analysis 2

1.2.3 Algorithm 2

2 Technical Approach 2

2.1 Overall Approach 2

2.2 Data 3

2.2.1 Manual Annotation 4

2.2.2 Dynamic Time Warping 4

2.3 Analysis 5

2.3.1 The performance difference of surgeons at different levels 6

2.3.2 Direct causes of the difference 6

2.3.3 Future directions 6

2.4 Algorithm 6

2.4.1 Deep learning part 6

2.4.2 Causal Inference part 8

3 Results & Deliverables 9

3.1 Data 9

3.2 Analysis 10

3.2.1 The performance difference between Novice and Expert. 10

3.2.2 Analysis for the direct cause of the difference 11

3.3 Algorithm 13

3.3.1 Overfit procedure for one sample 14

3.3.2 Overfit procedure for one series 14

3.3.3 Overfit procedure for one series after scaling 15

3.3.4 Train epochs for 400 segments 15

3.3.5 Train epochs for 400 segments with residue 15

3.3.6 Train epochs for 400 segments with residue with a larger model 16

4 Discussion 16

4.1 Data and analysis 16

4.2 Algorithm 17

5 Progress Evaluation 17

5.1 Dependencies 17

5.2 Activities and Deliverables **18**
5.3 Schedule Adherence **19**
6 Conclusion ***19***
6.1 Significance **19**
6.2 Future work **20**
6.3 Conclusion **20**
References ..... ***I***

1 Introduction

1.1 Motivation

The quality of a robot-mediated surgery is highly related to the skill of the surgeon. The more skillful the surgeon is, the better the patient output will be. Thus, improving technical proficiency is always worth for researchers to look into. Our goal for this project is to develop a robust and interpretable system to empower novice surgeons. One way to achieve this is to translate the novice commands to the surgery robot into the commands that are more likely to be given by more experienced and skillful expert surgeons.

Recent deep learning algorithms dominate large amounts of benchmarks for some areas including action recognition and prediction. Deep learning's success in action prediction provides us a powerful method for surgery assistance. We may treat the assistance as a kinematic prediction task using a deep neural network with the provided context and surgery task as input. Empirically expected, if designed and implemented properly and with enough training data provided, the deep learning methods will produce the state-of-the-art performance to assist a novice surgeon during surgery. However, deep learning methods have their drawbacks. The most significant and related drawbacks are generalization issues and low interpretability. If the environment of the surgery is not from the same distribution where the training data of the networks are generated, the deep learning methods have large probabilities to fail in incomprehensible ways. These drawbacks make the assistance system not reliable enough to put into practice.

Causal inference mechanisms, with causal relation provided, are recently being highly focused on and explored by researchers to deal with the generalization issues and low interpretability of the deep learning algorithms. We follow some recent works in incorporating causal inference mechanisms into deep learning architectures and try to improve the robustness of the assistance system through counterfactual inquiry.

1.2 Goals

The ultimate goal of this project is to build a robust system to assist surgeons especially novice surgeons perform at the expert level by a simple query “Do something (task) somewhere (task context) as an expert”.
This ultimate goal can be divided into three aspects of subgoals: data, analysis, and algorithm.

1.2.1 Data

Both exploring the difference between expert and novice surgeons and building and testing suitable algorithms need the support of data. We need to create a suitable dataset, which is well-aligned and paired, for both the analysis and algorithm.

1.2.2 Analysis

To build a robust system for surgery assistance, we need to have a deep understanding of the difference between novice and expert surgeons. So, in this project, after we build a suitable dataset, it is necessary to do a data analysis to gain more understandings of the performance difference of surgeons at a different level and its potential causes.

1.2.3 Algorithm

The algorithm is the core part of this project. A powerful while the robust and interpretable algorithm is needed. Predicting future kinematics itself is not a trivial task even for the dominatingly high-performance deep learning algorithm (DiPietro & Hager, 2018). So, for the algorithm, we aim at exploring the latest powerful transformer architecture (Vaswani, et al., 2017) to first achieve a feasible result for surgery assistance and then exploring the causal inference mechanisms to increase the robustness and interpretability of the current algorithm.

2 Technical Approach

2.1 Overall Approach

The overall approach of building a robust system to assist surgeons especially novice surgeons perform at expert-level by a simple query “Do something (task) somewhere (task context) as an expert” has 3 aspects – data, human understandings (i.e. data analysis), and algorithms.

As figure 1 shown, data is the prerequisite of both understandings and algorithms. We need to understand the difference of commands given by novice and expert surgeons by some statistical analysis for the data or some intuitional feelings after viewing sufficient videos of surgery operations. At the same time, we need data to train the network. And human understanding of the surgery procedure will help in designing both the deep learning algorithms and causal inference algorithms.

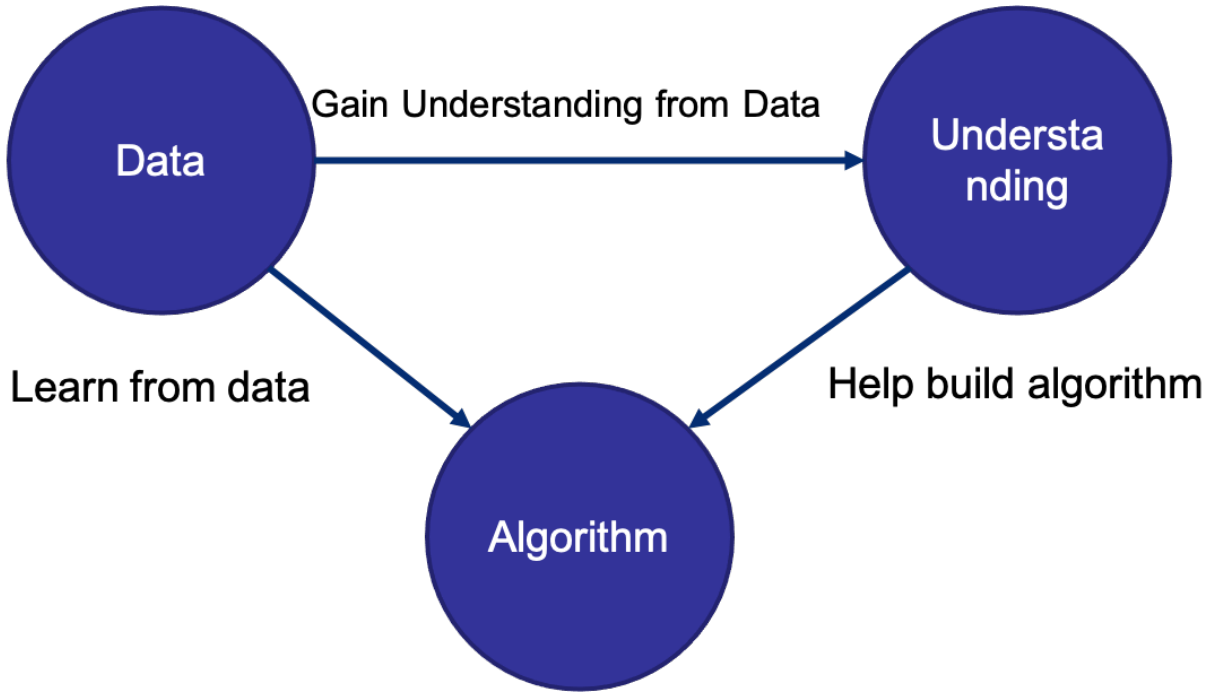

Figure 1. Overall technical approach

2.2 Data

The foundation of our data is the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAW) (Gao, et al., 2014) which was captured using the da Vinci Surgical System from eight surgeons with different levels of skill performing five repetitions of three elementary surgical tasks on a bench-top model. The JIGSAW dataset contains 3 components: kinematic data, video data, and human annotation of skill level and gestures. Figure 2 are examples of video frames in the JIGSAW dataset on different tasks (suturing, knot tying, needle passing)

Figure 2. examples of video frames in JIGSAW

Our target on the dataset is to temporally align the video samples for causal analysis. To achieve this, we firstly manually segment and pair videos and then make paired segments into the same length. We use the dynamic time warping algorithm with the l1 norm between normalized traveled distance as the distance function to align the video segments of different lengths.

2.2.1 Manual Annotation

Although the gesture of each frame is already annotated in the JIGSAW dataset, the annotation is not motion level. One gesture may consist of an arbitrary number of motions with different purposes and states. So, we annotate start and end keyframes as well as state/purpose for motions. The explanation of the annotation for the state/purpose is shown in table 1.

Annotation	Explanation
b	Start of the annotation
e	End of a gesture
s	Successful main motion
f	Failure main motion
m	Make-up motion for the previous failure motion
n	Follow-up motion for the previous successful motion
a	Adjustment motion before the main motion

Table 1. Explanations for the annotation

2.2.2 Dynamic Time Warping

Dynamic Time Warping (DTW) is an algorithm to calculate the optimal matching between two sequences. In our case, we do the dynamic time warping based on kinematics. Let's assume we have two series of kinematics:

$$K_1 = k_1[0], k_1[1], \dots, k_1[i], \dots, k_1[n-1]$$
$$K_2 = k_2[0], k_2[1], \dots, k_2[j], \dots, k_2[m-1]$$

Then, we can form an n-by-m grid, where each point (i, j) is the alignment between $k_1[i]$ and $k_2[j]$. A warping path W maps the elements of K_1 and K_2 to minimize the distance between them. W is a sequence of grid points (i, j). The distance function in our case is defined as the l1 norm between normalized traveled distance:

$$d(i,j) = \left| \frac{dist(k_1[i], k_1[0])}{dist(k_1[n], k_1[0])} - \frac{dist(k_2[j], k_2[0])}{dist(k_2[m], k_2[0])} \right|_1$$

$$dist(k[i], k[j]) = \sum_{x=i}^{j-1} |k[x+1] - k[x]|_2$$

The Optimal path to the grid point (i_x, j_x) can be computed by:

$$D_{min}(i_x, j_x) = \min_{i_{x-1}, j_{x-1}} D_{min}(i_{x-1}, j_{x-1}) + d(i_x, j_x)$$

Where:

$$0 \le i_x \le n-1$$
$$0 \le j_x \le m-1$$
$$i_x - 1 \le i_{x-1} \le i_x$$
$$j_x - 1 \le j_{x-1} \le j_x$$

So, with the optimal path, we have the correspondence of the frames, so that we can make them into the same lengths by repeating some frames which correspond to multiple frames. Or we can do interpolation in replace of the repetition.

2.3 Analysis

We perform the data analysis on two aspects. First, we quantitatively show the difference in the performance of surgeons at different levels. Then, we Qualitatively analyze the direct causes of the difference. After analysis, we proposed some future directions for the exploration of surgery training or automated surgery research. The whole analysis is based on the Knot Tying subset of the JIGSAW dataset to analyze the direct cause of the different performances between novice and expert surgeons. In this dataset we have 36 videos and corresponding kinematics series of knot tying procedure from 8 surgeons. 2 surgeons are at the expert level (>100hrs), 4 surgeons are at the novice level

(<10hrs), 2 surgeons are at the intermediate level (10-100hrs). In each video, the surgeon performs knot typing twice at different pre-defined positions in the same order.

2.3.1 The performance difference of surgeons at different levels

We calculate the failure rate for each gesture by treating any failure or incomplete operation as a failure case for that gesture. We calculate the average time consumed for the successful cases.

2.3.2 Direct causes of the difference

we qualitatively show some of the direct causes we find while annotating the video which is divided into two aspects – different accuracy of perception and different surgery choices.

2.3.3 Future directions

We think more informative datasets, more detailed affordance detection, and better 3-d space perception ability are among the most important works at this period

2.4 Algorithm

The algorithms have two parts, the deep learning part, and the causal inference part.

2.4.1 Deep learning part

For the deep learning part, our design is based on a basic transformer network. The inputs to the network are (1) Task information: task id and its context (2) Expected skill level: novice- or expert- level (3) Kinematics of previous frames. All of them are the inputs into the encoder part and the third one is the input into the decoder. The output of the network is the kinematics of the following frames.

We implement the architecture and the corresponding training procedure on the ubuntu 18.04 operating system with python and the PyTorch deep learning package, the detailed environment information is listed in the repository which is published on Github. In the following subsections, we give a more detailed introduction of the data generation, architecture, and training procedure of the proposed transformer for kinematics prediction

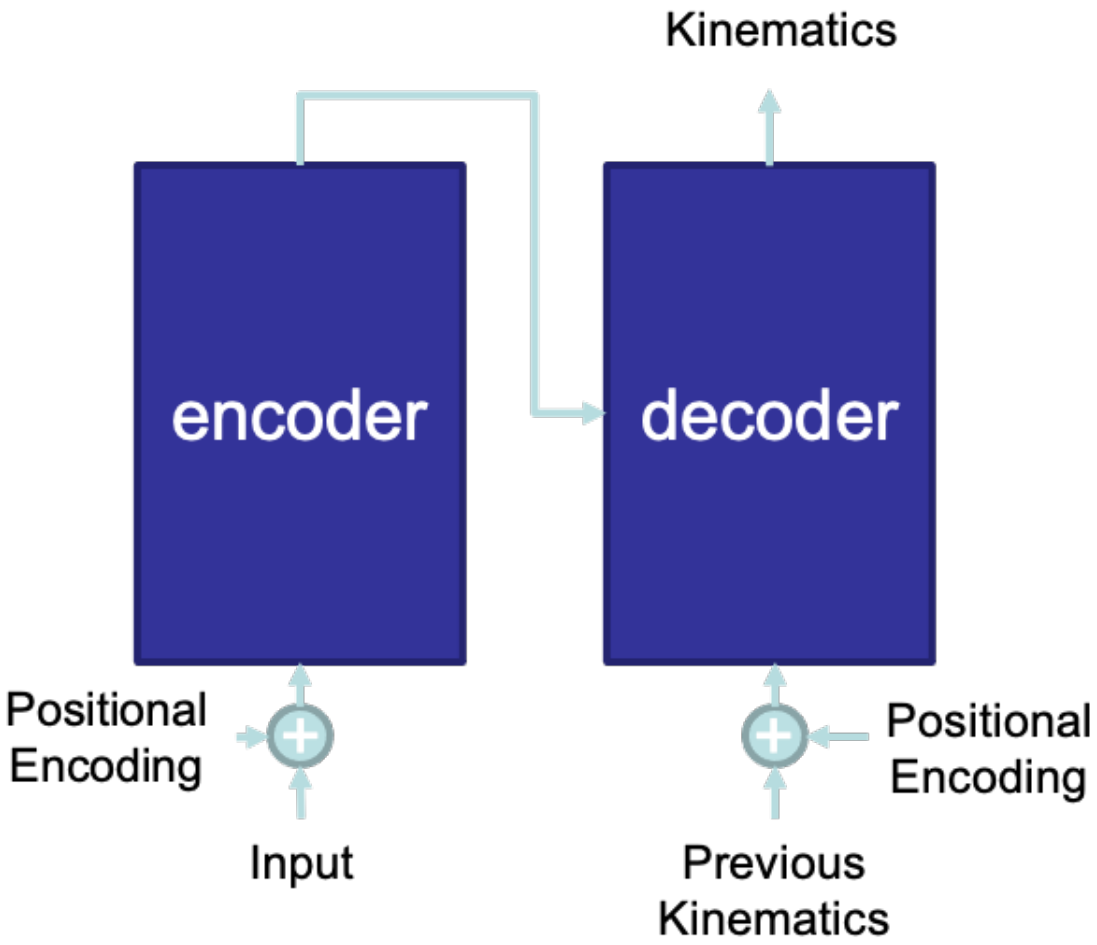

Figure 3. Illustration of the network

2.4.1.1 Data generation

We generate batched data for each segment with a fixed length as both source and target for the transformer architecture. Since the length for each segment is arbitrary, we do sampling corresponding to the desired batch size, input length, and output length. We sample the same number as the batch size of start points and get the first input size of data as source series and the following output size of data as target series.

2.4.1.2 Architecture

This model is made up of an encoder, a decoder, and a generator which are defined by the class Encoder, Decoder, and generator respectively.

The encoder consists of N encoder layers. The input of the encoder is the embedding of the input info (kinematics, task, state, video frames) plus a positional

encoding. The output of the decoder is a feature representation of the input info. The encoder layer consists of a self-attention layer, a feed-forward network, and sublayer connections.

Decoder consists of N decoder layers. The input of the encoder is the feature representation given by the encoder and the embedding of the target info (kinematics) plus a positional encoding. the output of the decoder is a feature representation. The decoder consists of a self-attention layer, cross-attention layer, feed-forward network, and sublayer connections.

The generator is an MLP for the kinematics prediction. The input of the generator is the feature representation given by the decoder. the output of the generator is the desired kinematics.

2.4.1.3 Training procedure

We implemented two training procedures. The first is the overfitting procedure where the model is only trained on samples generated from one segment to test the ability of the model and the correctness of the implementation. The second is the regular training procedure where the model is trained on the training split of the dataset for epochs until convergence.

2.4.2 Causal Inference part

For the causal inference part, the causal model is shown in figure 4, where we have 4 nodes, task id, context, skill level, and kinematic sequence. The context node is one of the direct causes of the choice of the task. Then the context and task choice and the skill level are all direct causes of the kinematic sequence.

To better describe the context and the task, besides feed in the video frame we might also extract the context affordance of the context like for suturing task we extract the in state and the out state from the video frame of the phantom.

As for how to incorporate the causal inference mechanism of this causal model into the deep learning architecture we refer to the following works to find out an effective way.

Deep Structural Causal Models for Tractable Counterfactual Inference https://arxiv.org/abs/2006.06485

CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training https://arxiv.org/pdf/1709.02023.pdf
Learning Functional Causal Models with Generative Neural Networks https://arxiv.org/abs/1709.05321
Counterfactual Generative Networks https://arxiv.org/pdf/2101.06046v1.pdf

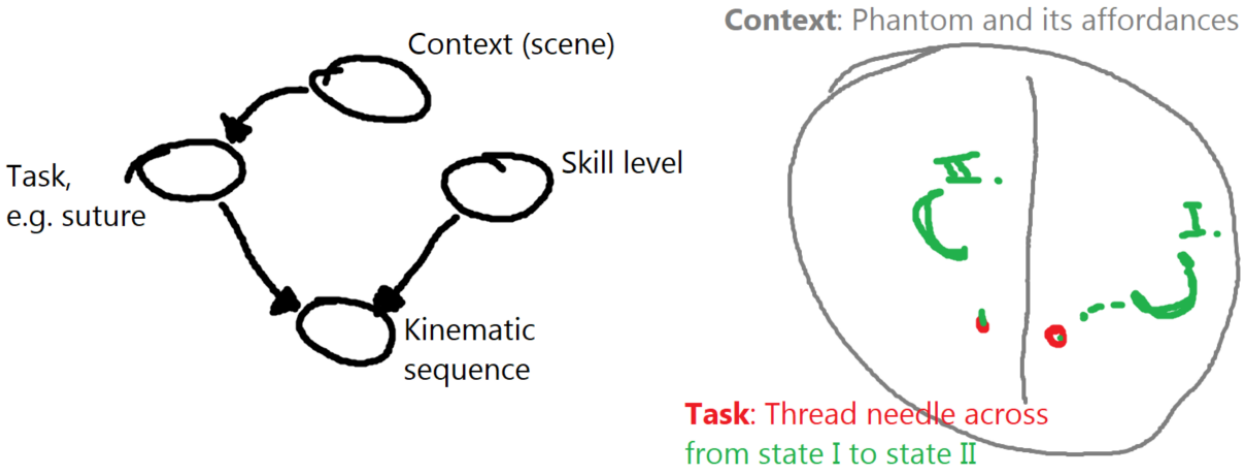

Figure 4. Causal Model and Illustration of Context

3 Results & Deliverables

3.1 Data

We implemented a convenient annotator for annotating videos. The code with the documentation of the annotator and the code for the dynamic time warping algorithm are share by this link. To use this annotator, input the python run command and input the id of the video as arguments. After inputting the command, a frozen video will be shown (the first frame of the video). You can press different buttons for operation. The function of each button is shown in table 2.

We annotated the Knot-tying subset of the JIGSAW dataset the annotated dataset is share by this link, the rectified motion-wise annotations are stored in the folder named rectified_transcriptions and the segmented videos and kinematics are stored in the folder named videoSegments.

Button	Function
p	to move to the next frame
l	to move to the previous frame
a	mark a keyframe and annotate the segment of this and the last keyframe as an adjustment motion before the main motion
b	mark a keyframe as the start of the annotation
n	mark a keyframe and annotate the segment of this and the last keyframe as a follow-up case for the previous successful case
s	mark a keyframe and annotate the segment of this and the last keyframe as a successful case
f	mark a keyframe and annotate the segment of this and the last keyframe as a failure case
m	mark a keyframe and annotate the segment of this and the last keyframe as a make-up case for the previous failure case
e	mark as a keyframe as the end of a gesture, after press an 'e', you need to: (1) give an annotation for the previous segments ('s', 'f', 'm', 'n'), (2) give two 0-9 digits to give annotation for the gesture (between 00-15)

Table 2. explanation of annotation command

3.2 Analysis

3.2.1 The performance difference between Novice and Expert.

We calculate the failure rate for each gesture by treating any failure or incomplete operation as a failure case for that gesture. We calculate the average time consumed for the successful cases. Results are shown in table 3 and table 4.

From the table, we can see that the experts have better performance than the novice on the overall failure rate and time consumed as expected. For the failure rate comparison, from gesture-wise results, we can see that the advantage of the experts mainly comes from gesture 15 (~0.28 to the novice) and gesture 14 (~0.25 to the intermediate). For other gestures, the differences are not significant (< 0.1) giving limited sampled numbers. For the time consumed, the main advantage comes from gestures 13,14, and 15 (>20 frames) which are the 3 core steps of the knot tying operation.

	G1	G12	G13	G14	G15	G11	Overall
Novice	0.077	0.187	0.117	0.205	0.333	0.067	0.189
Intermediate	0	0.053	0.1	0.4	0.25	0	0.158
Expert	0.05	0.143	0.2	0.15	0.05	0	0.108

Table 3. The failure rate of different level of skills

	G1	G12	G13	G14	G15	G11	Overall
Novice	87.5	88.0	103.8	81.8	134.4	86.9	85.73
Intermediate	66.9	79.3	52.8	55.9	164.9	83.5	84.25
Expert	87.3	70.3	67.8	62.1	164.2	97.6	105.04

Table 4. Time consumed for the successful cases of different level of skills (evaluated as number of video frames)

3.2.2 Analysis for the direct cause of the difference

3.2.2.1 Accuracy of perception

While annotating the data, we find that the most frequent failure cause is the wrong estimation of the relative 3d position between the end effector (the grabber) and the wire. Some of the failure cases move the grabber and make the wire beside the two fingers of the grabber instead of between them. Some of the failure cases don't move the grabber deep enough to reach the wire. The examples are shown in the following figures. In those cases, the overall trajectories of the movements are correct and simple to learn. The key to the success of the operation is to precisely locate the robot at the correct position and having an accurate perception of that. In figure 5, the left image is an example of a grabber beside wire and the right image is an example of the grabber not deep enough.

Figure 5. Failure grabbing case

3.2.2.2 Motion choices

During annotating, we find that some operation choices have an important effect on the success and efficiency of the following motions. One example is the grab point of gesture 14 which might be the main cause of the difference in the performance for gestures 14 and 15 between novice and expert. If that point is near the endpoint instead of the knot point, the motion of gesture 15 would be smoother because the end of the wire can be pulled out of the knot easier. The examples and explanations are shown in the following figures.

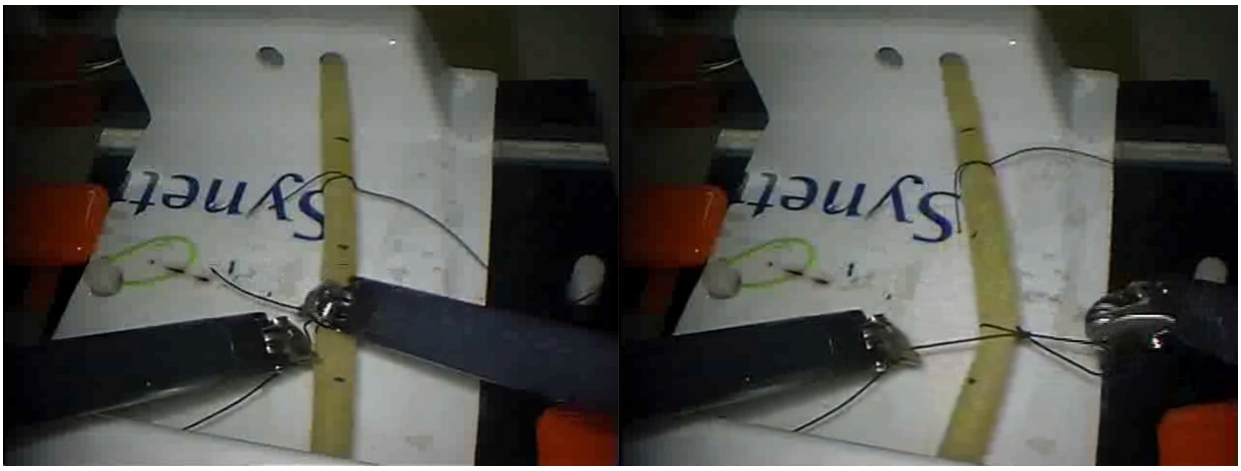

Figure 6. Example of grabbing near the knot

In figure 6, we can see that if the grab point is too near the knot, the end of the wire might be hard to pull out and might cause a failure case or longer time consumed or even damage to the tissue.

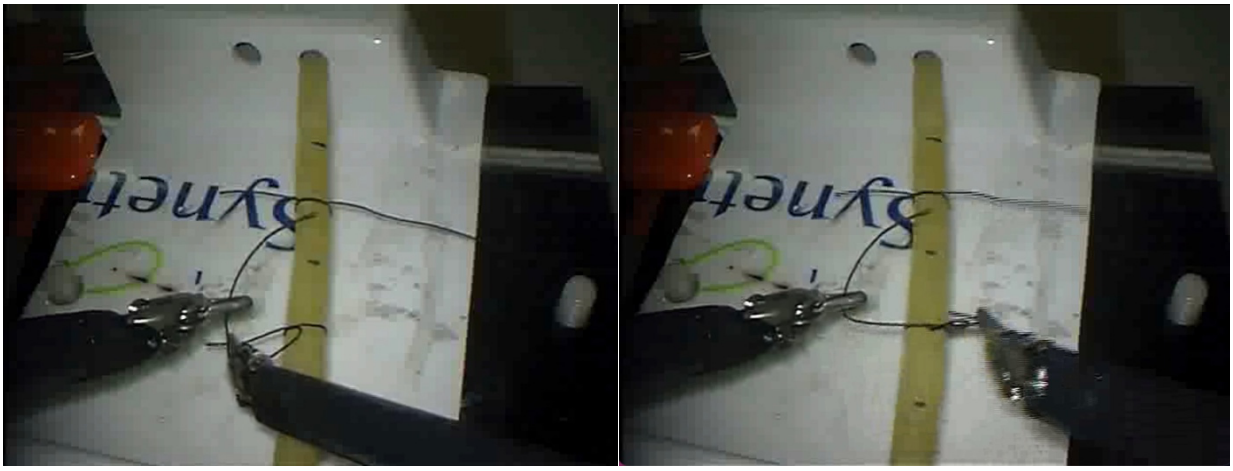

Figure 7. Example of grabbing near the end

In figure 7, we can see that if the grab point is near the endpoint of the wire, the end of the wire can be pulled out more easily which makes the pulling motion smoother, safer, and more efficient.

3.2.2.3 Potential future works

Informative datasets: The JIGSAW dataset is a well-designed and well-annotated dataset that contains rich data (kinematics and video) for surgery state recognition. However, according to the analysis for the causes of the different level performance, the precise environment information is vital for the success of a surgical operation. So, to further explore the potential of automated surgery or robot surgery training, a more informative dataset which contains not only video, kinematics, and gesture annotations but also precise environment information which may be present by depth information, semantic segmentation, affordance annotation, etc.

Detailed affordance detections: To learn the best surgery choice, more detailed affordance information should be provided. For example, where the best point or area to grab the wire is or where the wire can and should be bent to make the following grabbing easier.

Precise real-time 3d space perception: The 3d space information is also vital for the success of the surgical operation and it is also variable during the surgery. Precise real-time 3d space perception is the prerequisite of the success of an automated surgery procedure or robot assistance. Only with a high-quality real-time 3d perception algorithm, e.g. depth estimation, the robot can make precise and safe operations or offer the right and precise assistance.

3.3 Algorithm

We successfully implemented the kinematics predictor, which is an encoder-decoder with 10 corresponding layers each, with a feature dimension of 512. We perform some experiments on the model to check the correctness and the ability of the model with an input length of 30, and an output length of 10, and a batch size of 10. In the following sections, we introduce the detail of each experiment and the conclusion we can make from the experiments. We only take kinematics as input and output at this period. We use l1-norm as the loss function for training and Adam as the optimizer.

3.3.1 Overfit procedure for one sample

In this experiment, we sample once from a kinematics segment and train our model on this sample batch until convergence. In the end, we test our model on the same sample and the results of the x, y, z positions of the left tool and the right tool are shown in figure 8.

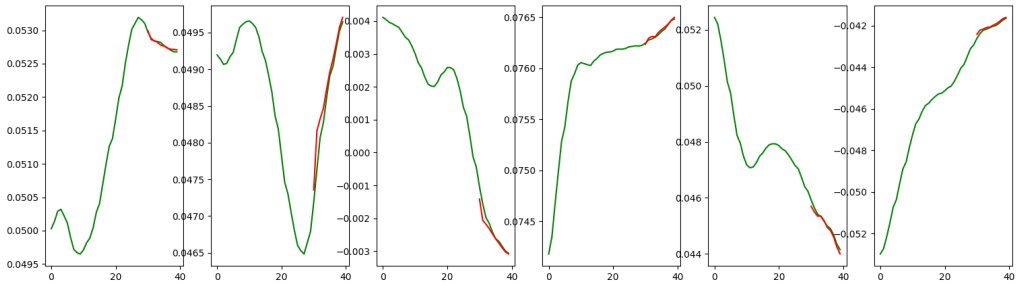

Figure 8. Overfit results on one sample.

As we can see, the model fit fairly well on this one sample, which suggests that our implementation of the model and the training procedure is correct to some extent.

3.3.2 Overfit procedure for one series

In this experiment, we train randomly sampled samples from one kinematics segment and train our model on randomly sampled data each iteration. In the end, we test our model on a random sampled sample from the same kinematics segments and the results of x, y, z positions of the left tool and right tool are shown in figure 9.

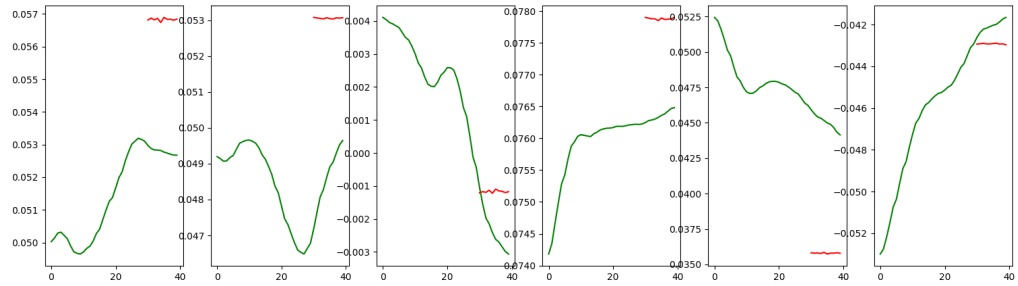

Figure 9. Overfit results on random samples from one segment.

The figure shows bad results, which we think is the issue of data scale, the data has too small magnitudes which make the model hard to describe, is some small perturbation happens.

3.3.3 Overfit procedure for one series after scaling

In this experiment, we scale up our kinematics data by a ratio of 100 and train on the same training setting of section 3.3.2, and we get a better result shows in figure 10. This shows that our guess is correct and our model is capable of handling some randomness during training.

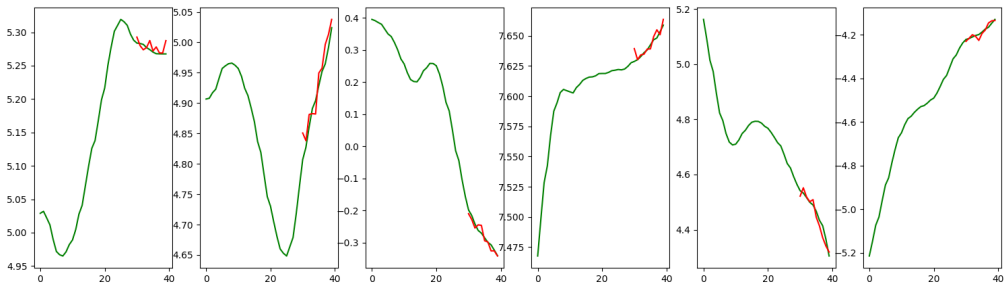

Figure 10. Overfit results on random samples from one segment.

3.3.4 Train epochs for 400 segments

In this experiment, we train our model on the training split which is the first 400 segments out of all 555 segments. In the end, we test our model on the same segment from all previous experiments and the results of x, y, z positions of the left tool and right tool are shown in figure 11.

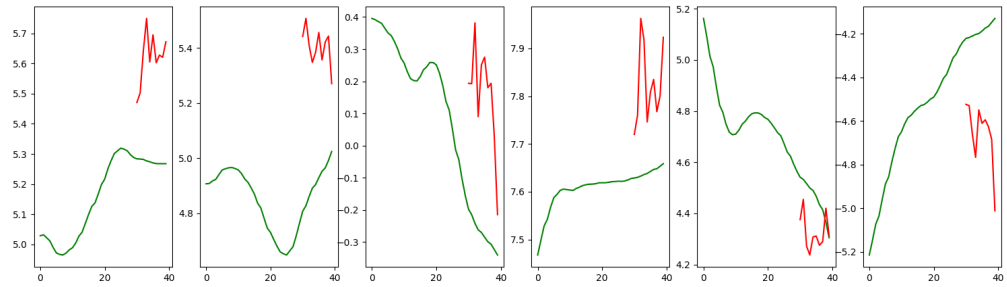

Figure 11. results on samples from one segment in training split

As we can see in the figure, the model does not learn well and perturbation a lot under this setting.

3.3.5 Train epochs for 400 segments with residue

In this experiment, to make the training for our model easier to learn from the data, we add the last frame of the source kinematics to the output to make the model learning a residue instead of the absolute kinematics of the tool. The model is trained as the same setting of section 3.3.4 and we get results in figure 12.

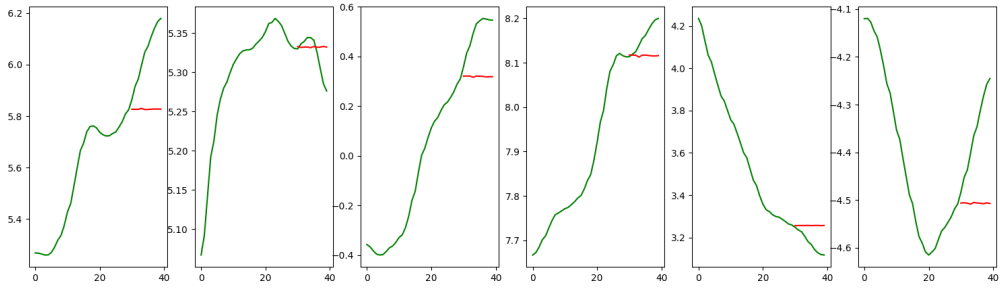

Figure 12. results of the residue learning on samples from one segment

As we can see, the difference truly decreases but the model converged to a local minimum which is a trivial solution where the residue is always 0. There might be 2 reasons for this. First, we might don't have enough information to let the model learn a more detailed trend of motion. Second, our model might lack the capacity to learn the trend of motion.

3.3.6 Train epochs for 400 segments with residue with a larger model

In this experiment, we increase the depth of the model from 10 layers to 20 layers for both the encoder and the decoder. The model is trained as the same setting of section 3.3.5 and we get results in figure 13

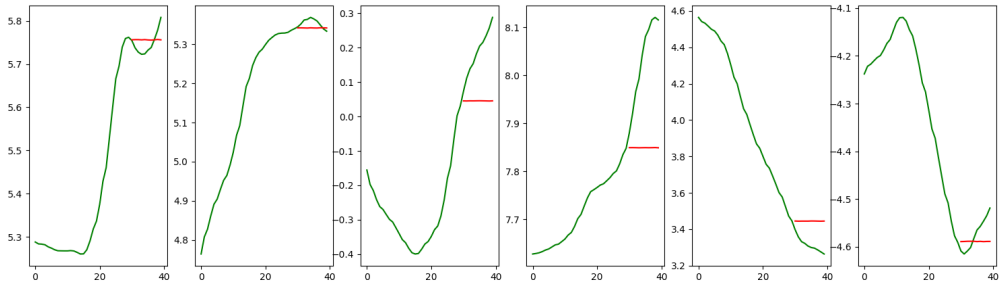

Figure 12. results of the residue learning for a larger model

As we can see, the results are very similar which makes us confident that the capacity of the model is not the reason.

4 Discussion

4.1 Data and analysis

JIGSAW dataset is a good dataset for multi-modal surgery state estimation. However, using this dataset to learn the robot kinematics prediction is not perfectly suitable. We as a human, give commands to the robot during surgery not only

based on what has been happened to the robot but also the current task and the context. Especially some key information. For example, for knot tying, if we are performing gesture 12 which is reach to the suture with the left tool, we need to know exactly where the future is. Although the machine we might be able to learn from provided video frames without supervision, it can still be imprecise. Even expert surgeons make mistakes in the perception of visual information which results in the failure of some motions. This means if we want to really achieve automated surgery, a more detailedly annotated dataset is needed.

4.2 Algorithm

From the experiment results, we can draw a rough conclusion that only use kinematics to predict kinematics is not feasible, we need more informative features such as video features, task information, etc. These will be further explored in the future.

5 Progress Evaluation

5.1 Dependencies

Since our project is performed purely on public datasets and is more related to analysis and algorithms, our dependencies are not difficult to get. The JIGSAW dataset is downloaded from its official website and doesn't have any contingency to be considered. LCSR has provided the thin6 server with 3 GPUs as the computational resources for the project. All dependencies were acquired at the beginning of the semester.

Dependency	**Need**	**Status**	**Contingency Plan**
JIGSAW dataset	Fundamental dataset	Acquired	N/A
Computational Resources	For Deep Learning experiments	Acquired	If crashed, ask Dr. Unberath for other computers, or acquire some cloud resources (e.g. AWS).

Table 5 Dependencies

5.2 Activities and Deliverables

The minimum target for this semester is to finish the preparation of the datasets. The dataset is the prerequisite both for the analysis and the algorithm design. This task will be performed by two activities – firstly manually segment and pair the videos, afterward implement the dynamic time warping algorithm to temporally align those pairs of videos. This is achieved and the deliverables - the task-aligned knot-tying subset of the JIGSAW dataset and the documented codes of annotator and alignment are provided. So, the minimal target is completed.

The expected target is to make statistical analysis for the properties of the moving trajectory of the operations in the prepared dataset. A written report for the analysis of the understandings into the novice- and expert-level robot command is provided. So, the expected target is completed.

The Stretch target is to implement the algorithms for the kinematic predictions. The deep learning algorithm is successfully implemented. The documented code repository is provided. However, since the model does not work as expected we need to explore more to make the model work better. Then incorporate causal inference into it to increase the robustness and interpretability. So, the stretch target is partially completed.

Level	Activities	Deliverables
Minimal	Manually segment and pair videos.	Task-aligned JIGSAW dataset for causal analysis + Codes for Annotator and alignments
	Implement DTW to align videos segments temporally	
Expected	Statistical analysis for some properties of the trajectory.	Written analysis of understandings into novice- and expert-level robot command
Stretch	Implement the Kinematic Prediction Network	Development and evaluation of counterfactual model
	Incorporate Causal inference mechanism into the DL methods	

Table 6. Activities and Deliverables

5.3 Schedule Adherence

We adjusted our schedule twice. When we find our initial annotation strategy is not well designed and re-designed the final motion-wise annotation. We gave two more weeks to re-annotate the data and two more weeks for the analysis. During the implementation of the network, we finally decide to focus on the deep learning algorithm and put the incorporation of causal inference into the future plan. Other than these two adjustments, we adhered to our schedule and made significant progress on the three subgoals of our project.

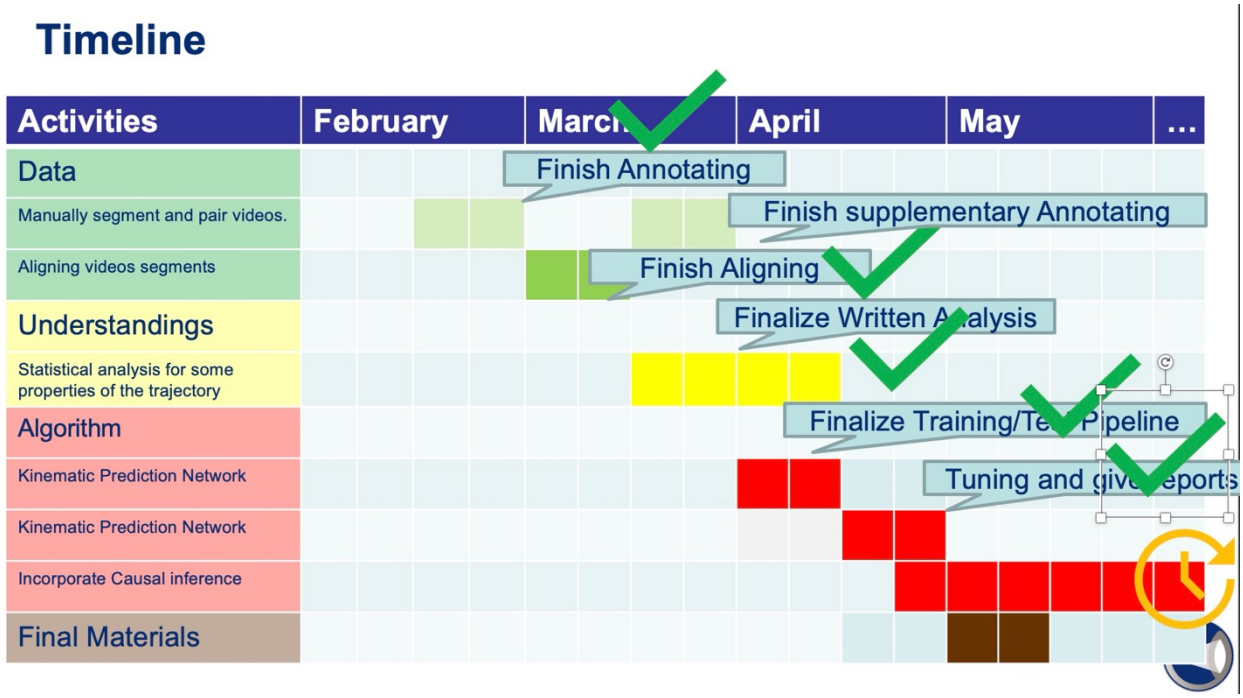

Figure 13. Gantt Chart

6 Conclusion

6.1 Significance

The results of this project are significant for the ultimate goal: building an assistance system for surgeons.

With the motion-wise annotated dataset, we are able to compare the difference between novice and expert surgeons at the motion level and also train the model for predicting kinematics of a well-defined motion which is a more feasible start point.

In our analysis, we show the difference and some direct causes of the difference which deepens our understanding of the surgical skills. Also, we proposed some deficiencies of the current dataset and proposed some future directions for the automated surgery study.

Our implementation and overfit experiments for the deep learning model proved the learning ability of the transformer architecture. Meanwhile, results of regular training correspond well to our analysis – a more informative dataset is needed. Only with more related and apparent information, the deep learning algorithm is able to learn to a meaningful solution instead of trivial local minimums.

6.2 Future work

From the discussion and the current state, we will continue on modifying the deep learning algorithm and perform experiments to get a more reasonable solution. When this is done, we will choose the next step from creating a more informative dataset and incorporating causal inference mechanisms. The choice depends on the final performance that the deep learning algorithm can achieve. If it finally gives a fairly good result, we will try to incorporate causal inference to improve the robustness of the deep learning algorithm. Otherwise, we will focus on creating a more suitable dataset to train the deep learning models.

6.3 Conclusion

Ultimately, this project is a good start point of the investigation of the surgeon assistance system and automated surgery. With the annotation and the analysis of the existing dataset, we gain more understanding on the surgical skills and the insufficiency of the existing dataset for autonomy study. By implementing the deep learning architecture and performing experiments on that. We proved our understanding and see the desire for more informative data to train a feasible kinematics predictor. We have already made a great progress on the three subgoals. However, we find that to achieve the ultimate goal, there is still a long journey.

References

DiPietro, R. S., & Hager, G. D. (2018). Unsupervised Learning for Surgical Motion by Learning to Predict the Future. *Medical Image Computing and Computer Assisted Intervention - MICCAI 2018 - 21st International Conference, Proceedings, Part IV. 11073*, pp. 281--288. Granada, Spain: Springer.

Gao, Y., Vedula, S. S., Reiley, C. E., Ahmidi, N., Varadarajan, B., Lin, H. C., . . . Hager, G. D. (2014). The JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS): A Surgical Activity Dataset for Human Motion Modeling. *In Modeling and Monitoring of Computer Assisted Interventions (M2CAI) – MICCAI Workshop.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems 30* (pp. 5998--6008). Long Beach, CA, USA: Neural Information Processing Systems Foundation, Inc.